Labelled Structures and Provability in Resource Logics - extended abstract -

E. Dumoulin and D. Galmiche

LORIA - Université Henri Poincaré
54506 Vandœuvre-lès-Nancy Cedex, France
dumoulin{galmiche}@loria.fr

Abstract We aim to emphasize the interest of labelled structures for analyzing provability in some resource logics. Labels and constraints allow to capture the semantic consequence relation in some resource logics, like BI logic that combines intuitionistic and linear connectives. They provide new methods in proof theory which are based on specific structures, namely dependency graphs or labelled proof nets. Such semantic structures are central for the analysis of provability and the generation of proofs or countermodels. Knowing that BI is conservative w.r.t. Multiplicative Intuitionistic Linear Logic (MILL), we consider MILL from the BI perspective and show how labelled proof structures can provide a new based-on connection characterization of MILL provability. We also provide an algorithm that builds MILL proof nets and its related connection method based on labelled structures and constraints. The generation of proofs and countermodels is analyzed in this context.

1 Introduction

Since last years there is an increasing amount of interest for logical systems that are resource sensitive. Among so-called resource logics, we can mention Linear Logic (LL)[11] with its resource consumption interpretation, Bunched Implications logic (BI) [17] with its resource sharing interpretation but also order-aware non-commutative logic (NL) [1]. As specification logics, they can represent features as interaction, resource distribution and mobility, non-determinism, sequentiality or coordination of entities. For instance, BI has been recently used as an assertion language for mutable data structures [12]. In this context, it is important to verify pre- or post-conditions expressed in this logic but also to discover non-theorems and, if possible, to provide explanation about non-validity by generating readable and usable countermodels.
In this paper we aim to discuss about models and labelled structures that capture the interactions and the semantics of resources. We present structures with labels and constraints, mainly labelled proof structures or nets, that can be seen as a kind of abstraction of formulae derivations in which we mainly manipulate labels and associated constraints instead of formulae. These structures give a geometrical representation of possible interactions in a formula [3]. Our perspective is to define methods in proof theory with a focus on semantic structures that provide a bridge between semantics and syntax and then could capture the essence of provability.
It is not trivial to find adequate methods in proof theory for the above mentioned resource logics. It is due to the specific management of resources or sets of resources (like bunches in BI or NL) and also to the difficulty to capture the specific interactions between connectives (additive and multiplicative connectives in BI, commutative and non-commutative connectives in NL), namely the particular semantics of these logics. Based-on semantics methods like tableaux or connection-based methods cannot be obtained for BI from standard methods with prefixes as defined for classical, intuitionistic or linear logics [14]. The notion of prefix is not strong enough to capture the semantics of BI and thus it is necessary to introduce another structure to deal with semantic information, namely a dependency graph (or resource graph) from which provability can be studied.
The correspondence between (resource) semantics and syntactic labels and constraints, used to defined new labelled calculi for BI, can be seen in both directions. For instance, in the case of BI without $\perp$, the labels and constraints directly reflect the elementary Kripke semantics of the logic [6] and then the relationships between semantics and dependency (or resource graphs) are clearly identified. In the case of BI (with $\perp$), the specific labels and constraints of the new proof theory do not reflect the initial Grothendieck topological semantics for which BI has been proved complete.

But, as the dependency graph associated to a BI formula contains the necessary information for analyzing provability, we can also deduce a new simple resource semantics that is complete for BI [8]. Even if our approach can be illustrated for a given logic, namely BI, we have to discuss and study if it can be developed for other logics in which we consider the formulae as resources and if it leads to new proof-theoretical foundations that are appropriate for verification tools, for instance in separation logics [12,20]. Moreover, it is important to notice that the labelled calculi with constraints can be used with different proof search methods: a tableaux method that deals with dependency graphs [9] but also a connection method that deals with particular labelled trees (including constraints) [5]. It is known that connection methods drastically reduce the search space compared to calculi analyzing the outer structure of formulae such as sequent or tableau calculi [2,22]. By introducing labels and constraints in a connection-based characterization of provability, we propose a simple and natural way to capture the essence of the provability and thus we generalize the based-on prefixes methods that are defined for IL or MLL [14]. The connection-based characterization of provability in BI with constraints has been defined in [5] and refined in [15] and provides an alternative method based on formula trees with constraints.
Starting from our previous results, we aim to emphasize the use and the interest of labelled structures with constraints by defining a connection-based characterization of provability for MILL. In fact, it is derived from the one proposed for BI by taking into account that BI is conservative w.r.t. MILL. This point was only mentioned in [5] and then we develop it in this paper in order to illustrate the underlying concepts. There exists a connection-based characterization of provability of Multiplicative Linear Logic (MLL) [14] based on prefixes but not for MILL because prefixes are not adapted to capture the initial semantics of MILL. Our new method is based on constraints that allow to capture the Urquhart's semantics of MILL at a syntactic level and generalize prefixes. Another reason to consider MILL is the possibility to relate our results with the notion of proof net that is a particular geometric representation of proofs defined in Linear Logic. As it was previously done for MLL [4], we can show that such a connection-based characterization and its related method for MILL can lead to an algorithm for the construction of MILL proof nets. Conversely, this algorithm can be seen as a new connection method, that also builds in parallel sequent proofs. This method starts with the formula (or decomposition) tree and builds, step by step and automatically, axiom-links (or connections) and partial proof nets, being guided by strategies and resolution of constraints. Taking into account labels and constraints attached to different positions in the formula tree, some steps of proof nets construction are only possible if some constraints are satisfied. Another interesting point to study is the generation of proofs and mainly of countermodels in case of non-provability in MILL.
In section 2, we summarize what is BI logic, mainly in a semantic perspective and we remind our results about provability based on dependency (or resource) graphs. BI being conservative w.r.t. MILL, we derive new results for MILL from the ones obtained for BI. In section 3 we develop the main concepts of a new characterization of provability in MILL. In section 4 we propose an algorithm for MILL proof nets construction, based on labels and constraints, that can be considered as a connection method that implements our new characterization. In section 5 we focus on the generation of proofs and countermodels from this connection method. This paper is focused on a particular approach of the quest of the essence of provability and of proofs and we expect to relate it to other studies based on other kinds of structures and semantic objects and also with works on games semantics for proof-search.

2 From BI to MILL

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 [18,19]. Its propositional fragment freely combines multiplicative (or linear) $*$ and $-\!\!*$ connectives and additive (or intuitionistic) $\wedge$, $\rightarrow$ and $\vee$ connectives [17] 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) [17] 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" [21]. 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, $m \models A \mathrel{-\!\!*} B$ 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 $A \mathrel{-\!\!*} B$ 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) and does not admit the usual number-of-uses reading [17]. BI logic has a sequent calculus with bunches with good properties but not well adapted to proof-search following backward reasoning (from the goal to the axioms).
In order to capture the specific interactions between connectives, for instance additive and multiplicative connectives in BI, and finally the semantics of connectives, we have defined labelled calculi including specific labels, constraints that allow to build semantic structures, called dependency (or resource graphs) [7]. Such structures contain the necessary semantical information from which provability can be studied. We can generate proofs or countermodels [6] from such labelled structures that can be also defined for other mixed logics like Non-commutative logic (NL) for which one has no simple resource semantics but only a bunched calculus not adapted to proof-search. In fact, a resource graph corresponds to a particular set of constraints and in order to propose a based-on connection method for BI we have integrated similar constraints in this formalism and thus proposed a new characterization of provability [5].
An important result is that BI logic is conservative w.r.t. Multiplicative Intuitionistic Linear Logic (MILL) and thus we can try to adapt the previous approach based on labelled semantic structures to MILL and analyze its impact on provability. It corresponds to generate semantic structures from MILL's Urquhart's semantics [21] and to develop a characterization of provability with labels and constraints that capture this semantics. It can be done with labelled tableaux calculi [6] but here we prefer to focus on connection-based calculi with constraints. A connection-based characterization of provability with constraints for BI has been defined in [5] and refined in [15]. Here we present its adaptation for MILL that was mentioned in previous works on BI but not explicitly developed. Our focus on MILL is also motivated by the possible relationships between our works and methods for proof nets construction or verification [4,16].

3 A Characterization of Provability in MILL

In this section, we focus on the notions of *labels* and *constraints* that are adequate for capturing the semantics of the connectives and their interactions. As said before, a connection-based characterization of provability with constraints for BI has been defined in [5] and refined in [15]. Thus, we present here its specialization (or refinement) for MILL and thus provide a new characterization based on labels and constraints for this logic. There exist methods based on prefixes defined for instance for IL or MLL [13,14,22] cannot be applied to MILL, the notion of prefix being not enough strong for capturing MILL semantics constraints. In fact, this development corresponds to start from MILL's Urquhart's semantics and to build specific semantic structures.

3.1 Labels and constraints

Given an alphabet C (for instance $a, b, c \cdots$), C^0, the set of *atomic labels on* C is defined as the set C extended with the unit symbol e. Then we define $C^\star$, the set of *labels on* C, as the smallest set including C^0 closed by composition ($x, y \in C^\star$ implies $xy \in C^\star$).
Let us note that *aabcc*, *cbaca* and *cbcaa*1 are equivalent by definition (associativity and identity 1. A *constraint* is an expression $x \leq y$ in which x and y are labels. A constraint $x \leq x$ is an *axiom* and we write $x = y$ to denote $x \leq y$ and $y \leq x$. The inference rules used for reasoning on constraints are:

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

The rule *trans* formalizes the transitivity of $\leq$ and the rule $func$ corresponds to the compatibility of the label composition for $\leq$. In this system, given a constraint k and a set of constraints H, we denote $H \mid\!\approx k$ the deduction of k from H. The notation $H \mid\!\approx K$, where K is a non-empty set of constraints, means that for all $k \in K$, $H \mid\!\approx k$.

3.2 Indexed formula trees with labels

Here we recall the standard notions coming from previous characterizations of provability by matrix [14,22]. A *decomposition tree* of a formula A is its representation as syntactic tree with nodes called *positions*. A position u exactly identifies a subformula of A denoted $f(u)$. An *atomic position* is a position for an atomic formula. The decomposition tree induces a partial order $\ll$ on the positions such that the root is the least element and if $u \ll v$ then u dominates v in the tree. In fact, we do not distinguish a formula A from its decomposition tree. For each position, we assign a polarity $pol(u)$, a principal type $ptyp(u)$ and a secondary type $styp(u)$. Therefore, we have different principal types depending on the connective and the associated polarity. For instance in BI we have four principal types named $\alpha, \beta, \pi\alpha, \pi\beta$.
Depending of the principal type, we associate a label $slab(u)$ and sometimes a constraint $kon(u)$ to a position u. Such a label is either a position or a position with a tilde in order to identify the formula that introduces resources. We define constraints in order to capture the composition and distribution of formulae that are considered as resources. The *labelled signed formula* $lsf(u)$ of a position u is a triple $(slab(u), f(u), pol(u))$ and is denoted $slab(u) : f(u)^{pol(u)}$.
The construction of the indexed formula tree is inductively defined in Figure 1.

$lsf(u)$	$ptyp(u)$	$kon(u)$	$lsf(u_1)$	$lsf(u_2)$
$x : (A \twoheadrightarrow B)^0$	$\pi\alpha$	$xu = \tilde{u}$	$u : A^1$	$\tilde{u} : B^0$
$x : (A * B)^1$	$\pi\alpha$	$u\tilde{u} \leq x$	$u : A^1$	$\tilde{u} : B^1$
$x : (A \twoheadrightarrow B)^1$	$\pi\beta$	$xu = \tilde{u}$	$u : A^0$	$\tilde{u} : B^1$
$x : (A * B)^0$	$\pi\beta$	$u\tilde{u} \leq x$	$u : A^0$	$\tilde{u} : B^0$

Figure 1. Signed formulae for MILL

For a given formula A the root position a_0 has a polarity $pol(a_0) = 0$, a label $slab(a_0) = 1$ and the signed formula $1 : (A)^0$ where 1 is the identity of the label composition. u_1 and u_2 are respectively the first and second subpositions. The subpositions inherite the formula and the polarity of the position. The principal type of a position u depends on its principal type and its polarity and the associated constraint is built from its principal connector and its label.
Like in BI, the constraints associated to $\pi\alpha$ formulae are called *assertions* and those associated to $\pi\beta$ formulae are called *obligations* (or *requirements*) and they must be satisfied from the set of assertions.
In fact, the rules $*_R$, $\twoheadrightarrow_L$ and $\rightarrow_L$ of BI's sequent calculus, that deal with $\pi\beta$ formulae, divide contexts and distribute resources. Because of weakening and contraction, we can have several occurrences of formulae and than we introduce a notion of *multiplicity* μ attached to $\pi\beta$ formulae. In this presentation, we keep this notion for MILL but we expect to show that a multiplicity of 1 is enough in MILL. Therefore, like in [22], we consider a formula A associated to a multiplicity μ and call this couple an *indexed formula* A^μ. Then the indexed formula tree for A^μ ($indt(A^\mu)$) is inductively defined as follows

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

The order relation $\ll^\mu$ between two indexed positions u^κ and v^τ is defined in the following way: $u^\kappa \ll^\mu v^\tau$ iff $u \ll v$ and κ is an initial sequence of τ. We denote $\mathcal{O}cc(u^\kappa)$ the set of indexed positions $u^{\kappa\tau}$ that are in A^μ.

3.3 Paths, connections and covers

In this paragraph, we consider the adaptations for MILL of the notions of paths, connections and covers as defined in [15] for BI.

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

An *atomic path* is a path that only contains atomic positions. A *configuration* of A^μ is a finite set of paths of A^μ.

Definition 3 (Reduction). *A* reduction *of an indexed formula A^μ is a finite sequence $(\mathcal{S}_i)_{1\le i\le n}$ of configurations in A^μ such that $\mathcal{S}_{i+1}$ is obtained from $\mathcal{S}_i$ by reduction of a position u in a path s of $\mathcal{S}_i$ following Definition 2. We say that $\mathcal{S}_{i+1}$ is obtained by reduction of $\mathcal{S}_i$ of u in s.*

Definition 4 (Connection). *A* connection *is a couple $\langle u, v \rangle$ of atomic positions such that $f(u) = f(v)$, $pol(u) = 1$ and $pol(v) = 0$. We denote $\mathcal{C}on$ the set of connections of A^μ.*[1]*.*

Definition 5 (Cover). *Let A^μ be an indexed formula. A connection $\langle u, v \rangle$ in A^μ* covers *a path s in A^μ if $u, v \in s$. Let S be a set of paths in A^μ, a* cover *of S is the set C defined as $C = \{ (s, \langle u, v \rangle)/s \in S$ and $\langle u, v \rangle \in \mathcal{C}on$ and $\langle u, v \rangle$ cover $s \}$ such that*

$$(s, \langle u, v \rangle) \in C \text{ and } (s, \langle u', v' \rangle) \in C \text{ imply that } u = u' \text{ et } v = v'.$$

A cover of A^μ is a cover of the set of atomic (consistent) paths in A^μ.

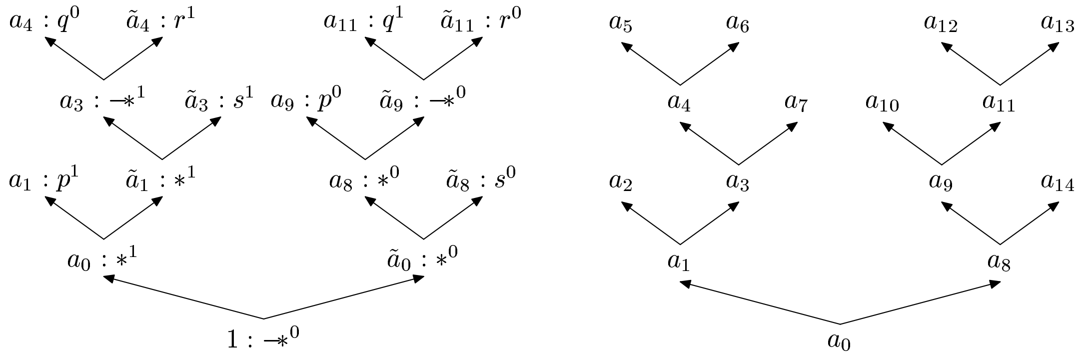
Given a formula A de BI, we use the following notations. The set of positions of A is denoted $\mathcal{P}os$. Given a set of positions $p \subseteq \mathcal{P}os$ we introduce:

- the set of positions of type $\pi\alpha$: $P_\alpha(p) = \{ u \mid u \in p \text{ et } ptyp(u) = \pi\alpha \}$
- the set of positions of type $\pi\beta$: $P_\beta(p) = \{ u \mid u \in p \text{ et } ptyp(u) = \pi\beta \}$
- the set of positions of secondary type $\pi\alpha_i$: $S_{\alpha i}(p) = \{ u \mid u \in p \text{ et } styp(u) = \pi\alpha_i \} (i \in \{ 1, 2 \})$
- the set of positions of secondary type $\pi\beta_i$: $S_{\beta_i}(p) = \{ u \mid u \in p \text{ et } styp(u) = \pi\beta_i \} (i \in \{ 1, 2 \})$
- the set of positions of secondary type $\pi\alpha$: $S_\alpha(p) = S_{\alpha 1}(p) \cup S_{\alpha 2}(p)$
- the set of positions of secondary type $\pi\beta$: $S_\beta(p) = S_{\beta_1}(p) \cup S_{\beta_2}(p)$
- the set of constants: $\Sigma_\alpha(p) = \{ slab(u) \mid (\exists v \in P_\alpha(p))(u \text{ in the set of subpositions of } v) \}$
- the set of variables: $\Sigma_\alpha(p) = \{ slab(u) \mid (\exists v \in P_\alpha(p))(u \text{ in the set of subpositions of } v) \}$
- the set of assertions: $\mathcal{K}_\alpha(p) = \{ kon(u) \mid u \in P_\alpha(p) \}$
- the set of obligations: $\mathcal{K}_\beta(p) = \{ kon(u) \mid u \in P_\beta(p) \}$

When $p = \mathcal{P}os$, we simply write $P_\alpha, Pib, S_{\alpha i}, S_{\beta_i}, S_\alpha, S_\beta, \Sigma_\alpha$ et Σ_β.

An example. Let us consider the formula $A^\mu \equiv (p * ((q -\!\!* r) * s)) -\!\!* ((p * (q -\!\!* r)) * s)$ of MILL. Figure 2 presents its indexed formula tree.
The two sets of positions $\pi\alpha$ and $\pi\beta$ are $P_\alpha = \{ a_0, a_1, a_3, a_{11} \}$ and $P_\beta = \{ a_4, a_8, a_9 \}$. Moreover, the set of assertions is $\mathcal{K}_\alpha = \{ a_0 = \tilde{a}_0, a_1\tilde{a}_1 = a_0, a_3\tilde{a}_3 \le \tilde{a}_1, \tilde{a}_9 a_{11} \le \tilde{a}_{11} \}$ and the set of obligations is $\mathcal{K}_\beta = \{ a_3 a_4 = \tilde{a}_4, a_8\tilde{a}_8 \le \tilde{a}_0, a_9\tilde{a}_9 \le a_8 \}$.
The reduction path process from $\{ a_0 \}$ provides the following atomic paths:
$s_1 = \{ a_2, a_5, a_7, a_{10} \}$, $s_2 = \{ a_2, a_5, a_7, a_{12}, a_{13} \}$, $s_3 = \{ a_2, a_5, a_7, a_{14} \}$, $s_4 = \{ a_2, a_6, a_7, a_{10} \}$, $s_5 = \{ a_2, a_6, a_7, a_{12}, a_{13} \}$ et $s_6 = \{ a_2, a_6, a_7, a_{14} \}$.
Their respective connections are:
$\langle a_2, a_{10} \rangle$, $\langle a_{12}, a_5 \rangle$, $\langle a_7, a_{14} \rangle$, $\langle a_2, a_{10} \rangle$, $\langle a_6, a_{13} \rangle$ and $\langle a_7, a_{14} \rangle$ and we have a cover C of A^μ : $C = \{ (s_1, \langle a_2, a_{10} \rangle), (s_2, \langle a_{12}, a_5 \rangle), (s_3, \langle a_7, a_{14} \rangle), (s_4, \langle a_2, a_{10} \rangle), (s_5, \langle a_6, a_{13} \rangle), (s_6, \langle a_7, a_{14} \rangle) \}$.
The set of constraints from connections generated by C is then
$\mathcal{K}_C = \{ (s_1, a_1 \le a_9), (s_2, a_{11} \le a_4), (s_3, \tilde{a}_3 \le \tilde{a}_8), (s_4, \tilde{a}_1 \le \tilde{a}_9), (s_5, \tilde{a}_4 \le \tilde{a}_{11})(s_6, \tilde{a}_3 \le \tilde{a}_8) \}$.

[1] Remark: the first position in a connection is the one of polarity 1.



u	$pol(u)$	$f(u)$	$ptyp(u)$	$styp(u)$	$slab(u)$	$kon(u)$
a_0	0	$(p * ((q -\!\!* r) * s)) -\!\!* ((p * (q -\!\!* r)) * s)$	$\pi\alpha$	$-$	1	$a_0 = \tilde{a}_0$
a_1	1	$p * ((q -\!\!* r) * s)$	$\pi\alpha$	$\pi\alpha_1$	a_0	$a_1\tilde{a}_1 \leq a_0$
a_2	1	p	$-$	$\pi\alpha_1$	a_1	$-$
a_3	1	$(q -\!\!* r) * s$	$\pi\alpha$	$\pi\alpha_2$	$\tilde{a}_1$	$a_3\tilde{a}_3 \leq a_1$
a_4	1	$q -\!\!* r$	$\pi\beta$	$\pi\alpha_1$	a_3	$a_3 a_4 = \tilde{a}_4$
a_5	0	q	$-$	$\pi\beta_1$	a_4	$-$
a_6	1	r	$-$	$\pi\beta_2$	$\tilde{a}_4$	$-$
a_7	1	s	$-$	$\pi\alpha_2$	$\tilde{a}_3$	$-$
a_8	0	$(p * (q -\!\!* r)) * s$	$\pi\beta$	$\pi\alpha_2$	$\tilde{a}_0$	$a_8\tilde{a}_8 \leq \tilde{a}_0$
a_9	0	$p * (q -\!\!* r)$	$\pi\beta$	$\pi\beta_1$	a_8	$a_9\tilde{a}_9 \leq a_8$
a_{10}	0	p	$-$	$\pi\beta_1$	a_9	$-$
a_{11}	0	$q -\!\!* r$	$\pi\alpha$	$\pi\beta_2$	$\tilde{a}_9$	$\tilde{a}_9 a_{11} = \tilde{a}_{11}$
a_{12}	1	q	$-$	$\pi\alpha_1$	a_{11}	$-$
a_{13}	0	r	$-$	$\pi\alpha_2$	$\tilde{a}_{11}$	$-$
a_{14}	0	s	$-$	$\pi\beta_2$	$\tilde{a}_8$	$-$

Figure 2. Indexed formula tree of $(p * ((q -\!\!* r) * s)) -\!\!* ((p * (q -\!\!* r)) * s)$

3.4 A new characterization of MILL provability

We now derive a new based-on characterization for MILL based on constraints that generalize the standard notion of prefix, already adapted to the MLL fragment [14]. It is derived from previous works in BI [15].

Definition 6 (MILL-substitution). *Let A^μ an indexed formula. A* MILL-substitution *is an application $\sigma : \Sigma_\beta \to {\Sigma_\alpha}^\star$, that can be extended to labels and constraints as follows:*

- *$\sigma(x) = x$ if x is a constant or if $x = 1$,*
- *$\sigma(x \bullet y) = \sigma(x) \bullet \sigma(y)$,*
- *$\sigma(x \leq y) = \sigma(x) \leq \sigma(y)$.*

Definition 7 (MILL-certification). *Let A^μ an indexed formula. A* MILL-certification *for A^μ is an application $\gamma : P_\beta \to \wp(P_\alpha)$ that associates, to any indexed position of principal type $\pi\beta$, a subset of the set of positions with $\pi\alpha$ as principal type.*

Definition 8 (Complementarity). *Let A^μ be an indexed formula and σ a MILL-substitution, a path s of A^μ is* complementary under σ*, or* σ-complementary*, if it is covered by a connection $\langle u, v \rangle$ such that $\sigma(\mathcal{K}_\alpha) \models\!\!\!\approx \sigma(slab(v)) \leq \sigma(slab(u))$. A cover C is complementary under σ if all paths s are complementary under σ.*

Definition 9 (CMILL-Provability). *A formula A of MILL is* CMILL-provable *if there exist a multiplicity μ, a cover C of the set of atomic paths of A^μ, a MILL-substitution σ and a MILL-certification γ for A^μ such that:*

(CBI1) the reduction relation $\lhd$ is irreflexive,

(CBI2) $\forall(s,\langle u,v\rangle)\in C, \forall w\in P_\beta(\{u,v\}), \gamma(w)\subseteq P_\alpha(s)$,
(CBI3) $\forall(s,\langle u,v\rangle)\in C, \forall w\in P_\beta(\{u,v\}), \sigma(k(\gamma(w)))\models\!\!\approx \sigma(k(w))$,
(CBI4) $\forall(s,\langle u,v\rangle)\in C, \forall x\in \Sigma_\beta(\{u,v\}), \sigma(x)\in\Sigma_\alpha(s)$,
(CBI5) $\forall(s,\langle u,v\rangle)\in C, \sigma(\mathcal{K}_\alpha(s))\models\!\!\approx (\sigma(slab(v))\leq\sigma(slab(u))$.

Let us come back to our example. In order to find a MILL-substitution σ from $\mathcal{K}_C$, we consider: $\sigma(a_9)=a_1, \sigma(a_4)=a_{11}, \sigma(\tilde{a}_8)=\tilde{a}_3, \sigma(\tilde{a}_4)=\tilde{a}_{11}, \sigma(a_8)=X, \sigma(\tilde{a}_9)=Y$.
Then we have to compute $\sigma(\mathcal{K}_\alpha)\models\!\!\approx\sigma(\mathcal{K}_\beta)$ and then

1. $a_0=\tilde{a}_0, a_1\tilde{a}_1\leq a_0, a_3\tilde{a}_3\leq\tilde{a}_1, Ya_{11}=\tilde{a}_{11}\models\!\!\approx a_3a_{11}=\tilde{a}_{11}$
2. $a_0=\tilde{a}_0, a_1\tilde{a}_1\leq a_0, a_3\tilde{a}_3\leq\tilde{a}_1, Ya_{11}=\tilde{a}_{11}\models\!\!\approx X\tilde{a}_3\leq\tilde{a}_0$
3. $a_0=\tilde{a}_0, a_1\tilde{a}_1\leq a_0, a_3\tilde{a}_3\leq\tilde{a}_1, Ya_{11}=\tilde{a}_{11}\models\!\!\approx a_1Y\leq X$

From 1. we directly deduce $Y=a_3$ and also $\gamma(a_4)=\{a_{11}\}$. The obligation of 1. is the one of the position a_4 and in order to verify it we use the assertion $a_3a_{11}=\tilde{a}_{11}$ of position a_{11}. From 3. we deduce a trivial solution for X that is $X=a_1a_3$ and also that $\gamma(a_9)=\emptyset$. The condition 2. is verified because we have:

$$\dfrac{\dfrac{\dfrac{a_1\leq a_1 \quad a_3\tilde{a}_3\leq\tilde{a}_1}{a_1a_3\tilde{a}_3\leq a_1\tilde{a}_1}\,func \qquad a_1\tilde{a}_1\leq a_0}{a_1a_3\tilde{a}_3\leq a_0}\,trans \qquad a_0=\tilde{a}_0}{a_1a_3\tilde{a}_3\leq\tilde{a}_0}\,trans$$

and then we deduce $\gamma(a_8)=\{a_0,a_1,a_3\}$ since $a_0=\tilde{a}_0$, $a_1\tilde{a}_1\leq a_0$, $a_3\tilde{a}_3\leq\tilde{a}_1$ are the respective assertions of a_0,a_1,a_3.

In order to verify the conditions (CBI2) to (CBI5), let us consider the following table

$(s,\langle u,v\rangle)$	$P_\beta(\{u,v\})$	$P_\alpha(s)$	$\Sigma_\beta(\{u,v\})$	$\Sigma_\alpha(s)$
$(s_1,\langle a_2,a_{10}\rangle)$	a_8,a_9	a_0,a_1,a_3	$a_8,\tilde{a}_8,a_9,\tilde{a}_9$	$a_0,\tilde{a}_0,a_1,\tilde{a}_1,a_3,\tilde{a}_3$
$(s_2,\langle a_{12},a_5\rangle)$	a_4,a_8,a_9	a_0,a_1,a_3,a_{11}	$a_4,\tilde{a}_4,a_8,\tilde{a}_8,a_9,\tilde{a}_9$	$\tilde{a}_0,a_1,\tilde{a}_1,a_3,\tilde{a}_3,a_{11},\tilde{a}_{11}$
$(s_3,\langle a_7,a_{14}\rangle)$	a_8,a_9	a_0,a_1,a_3	$a_8,\tilde{a}_8$	$a_0,\tilde{a}_0,a_1,\tilde{a}_1,a_3,\tilde{a}_3$
$(s_4,\langle a_2,a_{10}\rangle)$	a_8,a_9	a_0,a_1,a_3	$a_8,\tilde{a}_8,a_9,\tilde{a}_9$	$a_0,\tilde{a}_0,a_1,\tilde{a}_1,a_3,\tilde{a}_3$
$(s_5,\langle a_6,a_{13}\rangle)$	a_4,a_8,a_9	a_0,a_1,a_3,a_{11}	$a_4,\tilde{a}_4,a_8,\tilde{a}_8,a_9,\tilde{a}_9$	$\tilde{a}_0,a_1,\tilde{a}_1,a_3,\tilde{a}_3,a_{11},\tilde{a}_{11}$
$(s_6,\langle a_7,a_{14}\rangle)$	a_8,a_9	a_0,a_1,a_3	$a_8,\tilde{a}_8$	$a_0,\tilde{a}_0,a_1,\tilde{a}_1,a_3,\tilde{a}_3$

Moreover, $\gamma(a_9)=\emptyset\subseteq P_\alpha(s)$, for any path $s\in\{s_1,s_2,s_4,s_5,s_6\}$ and $\gamma(a_8)=\{a_0,a_1,a_3\}\subseteq P_\alpha(s)$ for any path $s\in\{s_1\ldots s_6\}$. Moreover, for any path, $s\in\{s_2,s_5\}$, $\gamma(a_4)=\{a_{11}\}\subseteq P_\alpha(s)$. Then the condition (CBI2) is verified. In addition, for any path $s\in\{s_1,s_2,s_4,s_5\}$ we have $\sigma(a_9)=a_1\in\Sigma_\alpha(s)^\star$ and $\sigma(\tilde{a}_9)=a_3\in\Sigma_\alpha(s)^\star$ for any path $s\in\{s_1,s_2,s_3,s_4,s_5,s_6\}$ we have $\sigma(a_8)=a_1a_3\in\Sigma_\alpha(s)^\star$ and $\sigma(\tilde{a}_8)=\tilde{a}_3\in\Sigma_\alpha(s)^\star$, and for any path $s\in\{s_2,s_5\}$ we have $\sigma(a_4)=a_{11}\in\Sigma_\alpha(s)^\star$ and $\sigma(\tilde{a}_4)=\tilde{a}_{11}\in\Sigma_\alpha(s)^\star$.
It remains to compute the reduction relation $\lhd$ that is obtained by the transitive closure of the domination relation $\ll$, the instantiation relation $\sqsubset$ and the deduction relation $\sqsubset'$. The instantiation relation induced by σ is

$$a_1\sqsubset a_9, a_3\sqsubset a_9, a_1\sqsubset a_8, a_{11}\sqsubset a_4$$

and the deduction relation induced by γ is

$$a_0\sqsubset' a_8, a_1\sqsubset' a_8, a_3\sqsubset' a_8, a_{11}\sqsubset' a_4.$$

The reduction relation $\lhd$ is represented in Figure 3. As the graph is acyclic, A^μ is valid in MILL.

As illustrated by the example, the constraints have composed labels on the lefthand side. Moreover, the constraints for the implication deal with equality.

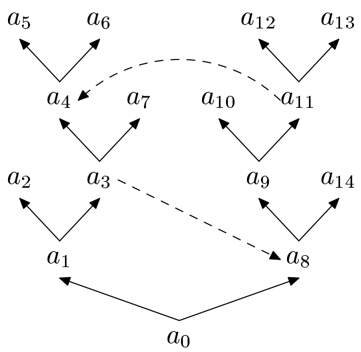


Figure 3. Reduction order for $(p * ((q -\!\!* r) * s)) -\!\!* ((p * (q -\!\!* r)) * s)$

3.5 Soundness and Completeness

These properties are proved like the corresponding ones for BI [15] with adequate restrictions to MILL.

Definition 10 (Complete reduction). *Let A^μ be an indexed formula and C be a cover of A^μ, a reduction $(\mathcal{S}_i)_{1\leq i\leq n}$ in A^μ is* complete *for C if C is a cover of $\mathcal{S}_n$.*

Definition 11 (Proper reduction). *Let A^μ be an indexed formula and σ be a MILL-substitution pour A^μ, a reduction $(\mathcal{S}_i)_{1\leq i\leq n}$ is* σ-proper *if*
(i) $\forall S \in (\mathcal{S}_i)_{1\leq i\leq n}, \forall s \in S, \sigma(\mathcal{K}_\alpha(s)) \models\!\!\!\approx \sigma(\mathcal{K}_\beta(s))$
(ii) $\forall S \in (\mathcal{S}_i)_{0\leq i\leq n}, \forall s \in S, \forall x \in \Sigma_\beta(s), \sigma(x) \in \Sigma_\alpha(s)^\star$.

Definition 12 (Realization). *Let A^μ be an indexed formula and s be a path of A^μ. A* CMILL-interpretation *of s in a resource model $\mathcal{K} = \langle\, (M, \sqsubseteq, \bullet, e), \models, [\![-]\!]\,\rangle$ is a function $\parallel - \parallel: \Sigma_\alpha(s) \to M$ that can be extended to labels $\Sigma_\alpha(s)^\star$ with $\parallel 1 \parallel = e$ and $\parallel xy \parallel = \parallel x \parallel \bullet \parallel y \parallel$.*
Given a MILL-substitution σ, a realization *of s is a couple $(\parallel - \parallel, \sigma)$ such that:*

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

A path is said realizable *if there exists a realization of s in a model $\mathcal{K}$. A configuration is realizable if there is at least one of its paths that is realizable.*

Lemma 1. *Let A^μ be an indexed formula, σ be a MILL-substitution for A^μ and $(\mathcal{S}_i)_{1\leq i\leq n}$ be a σ-proper reduction for A^μ, if $\mathcal{S}_i$ is σ-realizable then $\mathcal{S}_{i+1}$ is σ-realizable.*

Lemma 2. *Let A^μ be an indexed formula and σ be a MILL-substitution for A^μ. If a path s is complementary under σ then it is not realizable under σ.*

Proof. Let us suppose s σ-complementary and realizable for an interpretation ι in a resource model M. s is σ-complementary because it contains a connection that is σ-complementary. In fact, s contains a connection $\langle\, u, v\,\rangle$ that is complementary and such that
$f(u) = f(v)$, $pol(u) = 1$, $pol(v) = 0$ and $\sigma(\mathcal{K}_\alpha(s)) \models\!\!\!\approx \sigma(slab(v)) \leq \sigma(slab(u))$.
As s is realizable, we have $\parallel \sigma(slab(u)) \parallel \models f(u)$, $\parallel \sigma(slab(v)) \parallel \not\models f(u)$ et $slab(v) \sqsubseteq slab(u)$ that is contradictory because, by monotonicity, $slab(v) \sqsubseteq slab(u)$ and $\parallel \sigma(slab(u)) \parallel \models f(u)$ imply $\parallel \sigma(slab(v)) \parallel \models f(u)$.

In order to study the properties of this characterization, we consider the Urquhart's resource semantics for MILL [21].

Theorem 1 (Soundness of the characterization). *If a formula A is CMILL-provable then it is valid in the resource semantics of MILL.*

Proof. The conditions (CBI1) to (CBI5) of Definition 9 are verified because A^μ is provable. Let us assume that A is not valid in the semantics, then there exists a resource model M such that $e \not\models A$. Then, the initial set of paths $S_1 = \{ \{ a_0 \} \}$ is realizable under σ for the interpretation $\| - \|$ with an empty domain. The above mentioned conditions imply that there exists a reduction $(\mathcal{S}_i)_{1 \leq i \leq n}$ from S_1, that is complete for C, σ-proper and such that each path of S_n contains at least a connection of C. As S_1 is realizable under σ, by Lemma 1, S_n is also realizable under σ. But S_n cannot be complementary from Lemma 2. That is contradictory and then A is valid.

Theorem 2 (Completeness of the characterization). *If a formula A of MILL is valid in the resource semantics, then it is CMILL-provable.*

Proof. It is sufficient to show that if A is provable then it is CMILL-provable. The proof is by induction on the derivation in the MILL calculus deduced from LBI. Let us remind that a sequent $\Gamma \vdash A$ is provable in LBI iff $\Phi_\Gamma \multimap\!\!* A$ is provable in LBI. We only consider the case $-\!\!*_R$, the others being similar.
By induction hypothesis, we assume that $\Gamma, A \vdash B$ is CMILL-provable and show that $\Gamma \vdash A -\!\!* B$ is also CMILL-provable. As $\Gamma, A \vdash B$ is CMILL-provable, there exist a multiplicity μ, an atomic reduction $R_1 = (\mathcal{S}_i)_{1 \leq i \leq n}$ of $((\Phi_\Gamma * A) -\!\!* B)^\mu$, a cover C of S_n, a MILL-substitution σ and a MILL-certification γ for A^μ that satisfies the conditions of Definition 9.
From the atomic reduction R_1 for $((\Phi_\Gamma * A) -\!\!* B)^\mu$, we can build an atomic reduction R_2 for $(\Phi_\Gamma -\!\!* (A -\!\!* B))^\mu$. In the following figure, we give the first steps of R_1 on the lefthand side and those of R_2 and the righthand side.

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

After the two first reduction steps, we observe that the two paths of R_1 and R_2 contain the same signed formulae modulo a label renaming a_1 in a_i and $\tilde{a}_1$ in $\tilde{a}_i$. Consequently, the next steps of R_2 can be the same as for R_1 and both reductions introduce same signed formulae and label constraints modulo renaming. The assertions of the two first steps of R_1, $\{ a_0 = \tilde{a}_0, a_1\tilde{a}_1 \leq a_0 \}$, are weaker than those of R_2, $\{ a_0 = \tilde{a}_0, \tilde{a}_0 a_i = \tilde{a}_i \}$. Since R_1 provides a set of atomic paths satisfying CMILL-provability (see Definition 9), by induction hypothesis, the set of atomic paths for R_2 also satisfies these conditions.

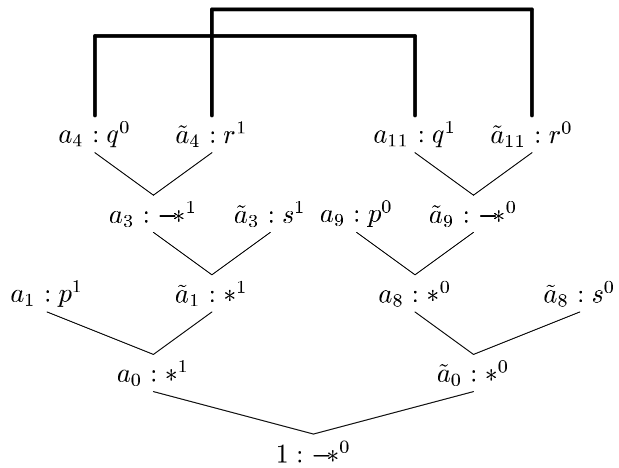
4 Connections and Proof Nets construction

The notion of proof nets has been introduced by Girard [11] in order to deal with the intrinsic parallelism of the sequent calculus. It has been defined for various fragments of linear logic and studied from both construction and verification perspectives [4,16]. It is known that there are strong relationships between connection methods and proof nets construction for MLL [4] and our aim is to analyze if the previous results can be related to MILL proof nets construction.

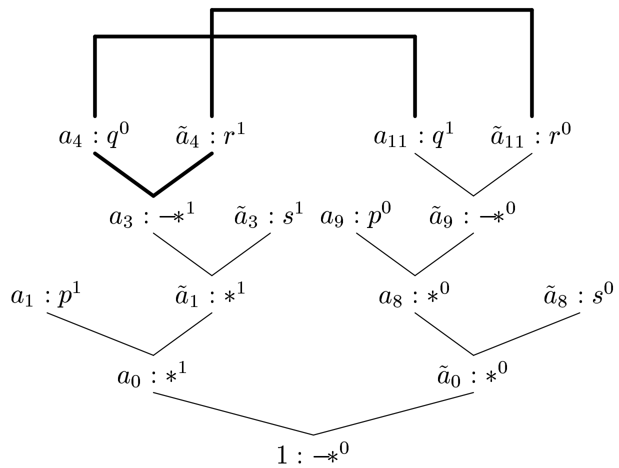
4.1 Proof nets with constraints

Let us first present the main ideas that are derived from the principles used in MLL and adapted to MILL with an emphasis on constraints. This approach is based on the construction of the decomposition tree with semantical information included in the tree: formulae, polarities, labels, constraints associated to subformulae.
Then, from basic results about permutabilities and proof-search strategies in Linear Logic [10], we know that we have to treat the connectives $-\!\!*^1$, $*^0$,$*^1$ and $-\!\!*^0$ following this order. It means that the $\pi\beta$-formulae have to be dealt before the $\pi\alpha$ formulae.

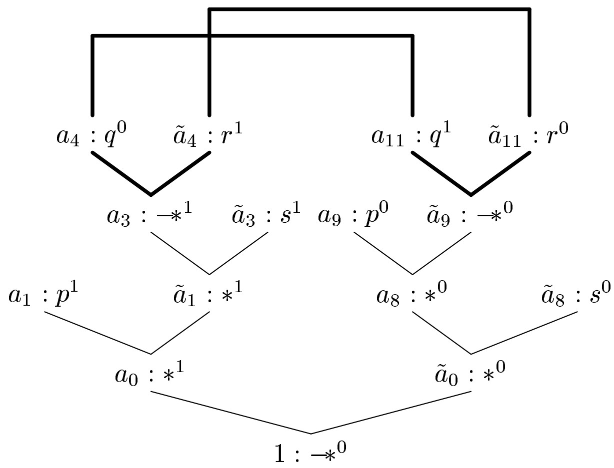
Therefore, we start from leaves (belonging to Σ_β) of a subformula of type $\pi\beta$ that is the highest in the labelled tree and try to connect them to leaves of the set Σ_α in order to build axiom-links (or connections). Each time the subformulae of a $\pi\beta$-formula belong to a net under construction, we generate a MILL-substitution σ and then apply it to the obligation associated to the formula. For instance, for a connection $\langle u, v \rangle$ in which $slab(u)$ is a variable and $slab(v)$ is a constant, we have $\sigma(slab(u)) = slab(v)$. When two premisses of a subformula of type $\pi\beta$ are conclusions of two partial proof nets, we merge them and extend the resulting net with a $\pi\beta$-link and provide a new proof net. We also add the corresponding constraint to the set of obligations after the application of σ. When two premisses of a subformula of type $\pi\alpha$ belongs to the same net, we extend it with a $\pi\alpha$-link and add the associated constraint to the set of assertions.
During this construction, several choices of connections are possible and then if, after a first choice, the resolution of constraints leads to a failure we must backtrack and going on with another choice. It is necessary to test all possibilities until the net covers all the initial decomposition tree. Then we can deduce if it is provable or not.
Let us consider the example of Section 3. First, we build the initial labelled tree that is the one of Figure 2. The highest $\pi\beta$ formula in this tree is a_4 and thus we start by trying to connect its subformulae (or subpositions) a_5 and a_6. Let us start with a_5 that is connected to a_{12} because $f(a_5) = f(a_{12}) = q, pol(a_5) = 0$ and $pol(a_{12}) = 1$. We obtain an elementary net $\mathcal{R}_1$ and $\sigma(a_4) = a_{11}$. Then, we connect a_6 to the only possible position a_{13} because $f(a_4) = f(a_{13}) = r$, $pol(a_6) = 0$ et $pol(a_{13}) = 1$. We create a new elementary net $\mathcal{R}_2$ and generate $\sigma(\tilde{a}_4) = \tilde{a}_{11}$.



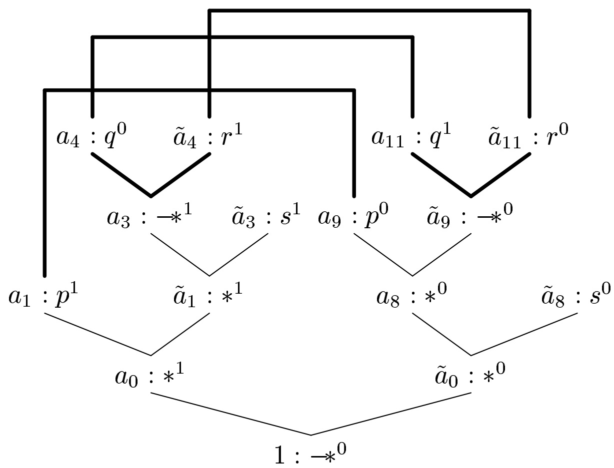
Thus, we merge $\mathcal{R}_1$ and $\mathcal{R}_2$ into a new net $\mathcal{R}_1$ with a $\pi\beta$-link and we apply σ to the obligation $a_3a_4 = \tilde{a}_4$. Then, we deduce $a_3a_{11} = \tilde{a}_{11}(Req1)$.



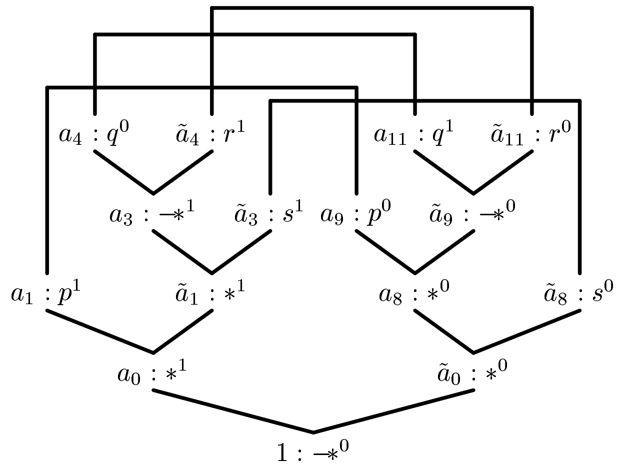
The leaves a_{12} et a_{13} are now conclusions of the net $\mathcal{R}_1$ and also the premisses of the position a_{11} that is a $\pi\alpha$-position. Then, we extend $\mathcal{R}_1$ with a $\pi\alpha$-link and obtain $\sigma(\tilde{a}_9) = a_3$ because the positions a_4 and a_{11} are linked and the assertion becomes $a_3a_4 = \tilde{a}_4$. Thus, it satisfies the obligation of a_4. Let us remark that, in order to satisfy $Req1$, we have used the position a_{11}.



Position a_9 corresponds to a $\pi\beta$ formula and its subposition a_{11} has been already treated. Then we need to consider the position a_{10}. We connect it with a_2 (that is the only possibility) and generate a new net $\mathcal{R}_2$ and deduce $\sigma(a_9) = a_1$. As the two subpositions of a_9 are the conclusions of $\mathcal{R}_1$ and $\mathcal{R}_2$, we merge them with extension by a $\pi\beta$-link in order to provide a new net $\mathcal{R}_1$. The application of σ to the obligation provides $a_1a_3 \leq \sigma(a_8)$. Then we have a solution for a_8 that is $\sigma(a_8) = a_1a_3$. In order to satisfy this obligation, we have not to use assertions.



It remains to consider the $\pi\beta$-position a_8. Its subposition a_{14} does not belong to a net and then we connect it to a_2 and thus we have $\sigma(\tilde{a}_8) = \tilde{a}_3$. It provides a new elementary net $\mathcal{R}_2$. The positions a_9 and a_{14} respectively belong to $\mathcal{R}_1$ and $\mathcal{R}_2$ and are the premisses of a_8. We can then merge $\mathcal{R}_1$ and $\mathcal{R}_2$ into a new $\mathcal{R}_1$ with a $\pi\beta$-link and apply σ to the obligation associated to a_8: $a_1a_3\tilde{a}_3 \leq \tilde{a}_0$ $(Req2)$. At position a_3, the two premisses are in fact conclusions of $\mathcal{R}_1$ and then we extend it with a $\pi\alpha$-link and we add the assertion $a_3\tilde{a}_3 \leq \tilde{a}_1$ to our set of constraints. By compatibility, we have $a_1a_3\tilde{a}_3 \leq a_1\tilde{a}_1$ (Ass). As before, we extend the net at the position a_1 with a $\pi\alpha$-link and a new assertion $a_1\tilde{a}_1 \leq a_0$ and then (Ass) becomes, by transitivity, $a_1a_3\tilde{a}_3 \leq a_0$. Finally, at position a_0, we extend the net $\mathcal{R}_1$ with a $\pi\alpha$-link and by transitivity (Ass) becomes $a_1a_3\tilde{a}_3 \leq \tilde{a}_0$ and consequently $(Req2)$ is verified. As for the obligation $(Req1)$, we are in position to claim that the assertions necessary to satisfy $(Req2)$ come from positions a_0, a_1 and a_3. We then conclude that all connections are σ-complementary and then A^μ is provable and then valid in MILL.



This method for proof nets construction corresponds to a particular algorithm that implements the based-on connection characterization of provability and then can be viewed as a new connection method for MILL. We observe in our example that the connections we find are the same than the ones obtained in the example developed in Section 3. We have generated connections that could be σ-complementary and then verified, step by step, the admissibility of the MILL-substitution. Moreover, for the resolution of constraints, we can say which assertions is used for the satisfaction of an obligation without using the process of path reduction. In addition, the computation of obligations from the assertions and of the certification are automatically made step by step.

4.2 An algorithm for proof nets construction

Let us describe more formally this algorithm that is based on the algorithm dedicated to the construction of MLL proof nets [4] in which, in addition to the search of connections, one must verify that the obligations are satisfied from the assertions. Here, we only describe the main steps of the algorithm.

input: Formula A
output: Proof net of A or failure.
step 0: Construction of the labelled decomposition tree.
step 1: Choice of a $\pi\beta$-position pm, not being already treated[2] and highest in the labelled tree.
step 2: If the left subposition pbg of pm is not already treated, connect pbg as follows:
If pbg is a leaf then
choose a position pa_1 such that $slab(pa_1)$ constant, $pol(pa_1) \neq pol(pbg)$ and $f(pa_1) = f(pbg)$;
build an elementary net R_1;
generate $\sigma(slab(pbfg)) = slab(pa1)$
Else return failure.
step 3: If the right subposition pbd of pm is not already treated, connect pbd as follows:
If $pbfd$ is a leaf then
choose a position pa_2 such that $slab(pa_2)$ constant, $pol(pa_2) \neq pol(pbd)$ and $f(pa_2) = f(pbd)$;
build an elementary net R_2;
generate $\sigma(slab(pbd)) = slab(pa2)$
Else return failure.
step 4: Merge R_1 and R_2 into a net R at position pbm and apply σ to its obligation (Obl) that becomes the current obligation;
Verify if (Obl) is an axiom: if yes return at step 1;
step 5: For all $\pi\alpha$-positions,
if the two premisses of a $\pi\alpha$-position, pam, are conclusions of R then
extend the net at this position; apply σ to the assertion of pam; add it to the resolution system;
verify (Obl) from the set of assertions.

[2] A position is treated if it belongs to a net

step 6: If the initial labelled tree is not completely covered by the net R then
if there is no $\pi\beta$-position to treat and at least a $\pi\beta$-position u has one subposition for which connections are possible then
if there are nets then
break the net from the current position to the position u and return to step 1
else if it remains $\pi\beta$ positions to treat then return to step 1.
else return failure
else return the net.

To verify an obligation means here that each time an assertion is added to the resolution set, one does the transitive and compatible closure of the assertion added from the last generated constraint and compare the new constraint with the last obligation.

This algorithm can be proved correct and complete from similar proofs of the algorithm for MLL proof nets [4] with addition of a specific treatment of the constraints.

Theorem 3 (Correctness). *If the algorithm returns a proof net for A then the formula A is provable in MILL.*

Theorem 4 (Completeness). *If a formula A is provable in MILL then the algorithm returns a proof net for A.*

Moreover this algorithm provides a connection method for MILL because it builds, step by step, a set of connections, a cover, a substitution and a certification such that A is CMILL-provable.
If we aim to relate the connection method associated to our new characterization and the construction of a proof net, we again consider our example with the formula $(p{*}((q{-\!*}r){*}s)){-\!*}((p{*}(q{-\!*}r)){*}s)$. We can observe that they both generate the same set of connections, namely $\langle a_2, a_{10} \rangle$, $\langle a_{12}, a_5 \rangle$, $\langle a_7, a_{14} \rangle$, $\langle a_2, a_{10} \rangle$, $\langle a_6, a_{13} \rangle$ and $\langle a_7, a_{14} \rangle$. Moreover, the MILL-substitutions and the MILL-certifications are the same in both cases. We aim, in future works, to study such an algorithm also in the context of verification [16] by focusing on the constraint resolution.

5 Generation of proofs and countermodels

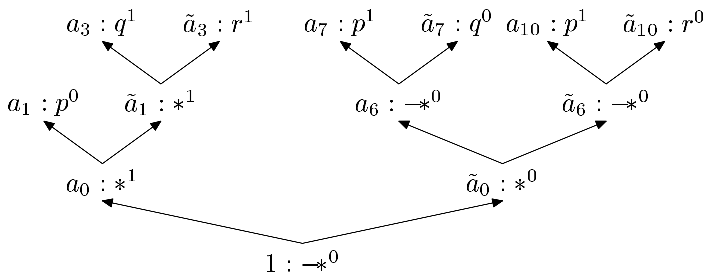
The previous algorithm builds a set of connections and a proof net in case of provability in MILL. It corresponds to a proof-search procedure with forward reasoning (from axioms to the goal formula) that can, step by step, build a proof in the MILL sequent calculus. In parallel with the proof nets construction, the algorithm builds a proof in a top-down way, from axioms (corresponding to axiom-links) by application of inference rules each time the corresponding partial nets are extended by a new link.
Let us come back to our example and show how the algorithm builds a sequent proof. The first connection $\langle a_5, a_{12} \rangle$ corresponds to the sequent $q \vdash q$ and the second one to $r \vdash r$. Then we build the sequent $q, q -\!\!* r \vdash r$ by the $\pi\beta$-link of the position a_4, and then, at position a_{11}, the $\pi\alpha$-link provides the sequent $q -\!\!* r \vdash q -\!\!* r$.
With the connection $\langle a_4, a_{11} \rangle$, we build the sequent $p \vdash p$ and the $\pi\beta$-link at position a_9 generates the sequent $p, q -\!\!* r \vdash p * (q -\!\!* r)$. Then, the connection $\langle a_7, a_{14} \rangle$ provides the sequent $s \vdash s$ and the $\pi\beta$-link applied to position a_8 generates $p, q -\!\!* r, s \vdash (p * (q -\!\!* r)) * s$. Finally, successive applications of $\pi\alpha$-links to the positions a_3, a_1, a_0 (following this order) provide, step by step, the sequent $p, (q -\!\!* r) * s \vdash (p * (q -\!\!* r)) * s$, the sequent $p * ((q -\!\!* r) * s) \vdash (p * (q -\!\!* r)) * s$ and finally the sequent $\vdash (p * ((q -\!\!* r) * s)) -\!\!* ((p * (q -\!\!* r)) * s)$.

The final sequent proof built in parallel of the proof net construction is the following:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{p \vdash p \quad \dfrac{\dfrac{q \vdash q \quad r \vdash r}{q, q \mathrel{-\!\!*} r \vdash r}\,{\mathrel{-\!\!*}}_L}{q \mathrel{-\!\!*} r \vdash q \mathrel{-\!\!*} r}\,{\mathrel{-\!\!*}}_R}{p, q \mathrel{-\!\!*} r \vdash p * (q \mathrel{-\!\!*} r)}\,*_R \qquad s \vdash s}{p, q \mathrel{-\!\!*} r, s \vdash (p * (q \mathrel{-\!\!*} r)) * s}\,*_R}{p, (q \mathrel{-\!\!*} r) * s \vdash (p * (q \mathrel{-\!\!*} r)) * s}\,*_L}{p * ((q \mathrel{-\!\!*} r) * s) \vdash (p * (q \mathrel{-\!\!*} r)) * s}\,*_L}{\vdash (p * ((q \mathrel{-\!\!*} r) * s)) \mathrel{-\!\!*} ((p * (q \mathrel{-\!\!*} r)) * s)}\,{\mathrel{-\!\!*}}_R$$

The most interesting point is the case of non-provability of a formula and then the possible generation of countermodels in a given semantics. Recent results on proof-search in BI, and consequently in MILL and IL [6], are based on a specific semantic structures, called resource graphs, that graphical representation of the set of assertions generated through the proof-search process. In case of non-provability, we can extract countermodels from such structures that can be also considered directly as geometric representations of countermodels. With our approach based on connections and on proof nets construction the questions of generation and representation of countermodels also arise. It appears that we can extract a countermodel from the labelled formula tree, an incomplete set of connections and the partial proof structure built before the failure in the construction process.

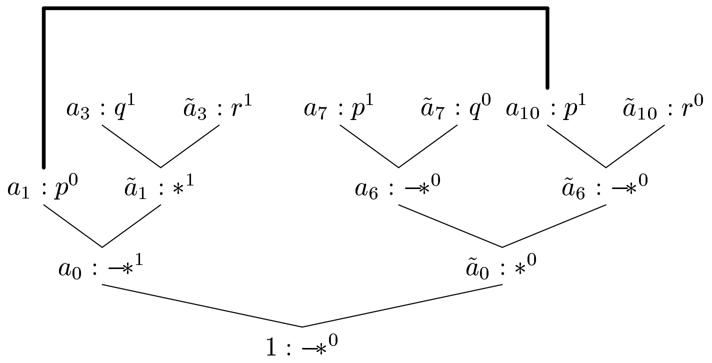
Let us illustrate this point with an example of a non-provable formula, namely the formula $(p \mathrel{-\!\!*} (q * r)) \mathrel{-\!\!*} ((p \mathrel{-\!\!*} q) * (p \mathrel{-\!\!*} r))$. Its indexed formula tree is given in Figure 4.



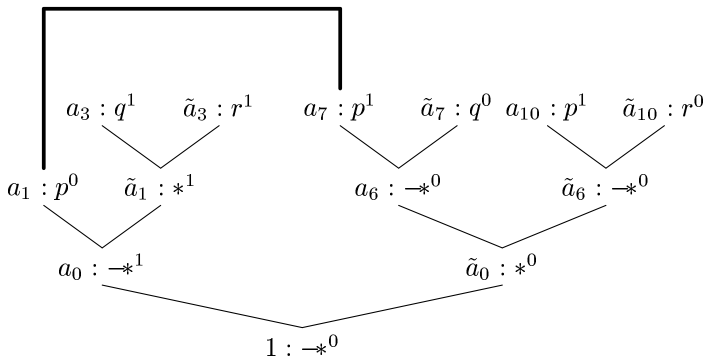
u	$pol(u)$	$f(u)$	$ptyp(u)$	$styp(u)$	$slab(u)$	$kon(u)$
a_0	0	$(p \mathrel{-\!\!*} (q * r)) \mathrel{-\!\!*} ((p \mathrel{-\!\!*} q) * (p \mathrel{-\!\!*} r))$	$\pi\alpha$	$-$	1	$a_0 = \tilde{a}_0$
a_1	1	$p \mathrel{-\!\!*} (q * r)$	$\pi\beta$	$\pi\alpha_1$	a_0	$a_0 a_1 = \tilde{a}_1$
a_2	0	p	$-$	$\pi\beta_1$	a_1	$-$
a_3	1	$q * r$	$\pi\alpha$	$\pi\beta_2$	$\tilde{a}_1$	$a_3 \tilde{a}_3 \leq \tilde{a}_1$
a_4	1	q	$-$	$\pi\alpha_1$	a_3	$-$
a_5	1	r	$-$	$\pi\alpha_2$	$\tilde{a}_3$	$-$
a_6	0	$(p \mathrel{-\!\!*} q) * (p \mathrel{-\!\!*} r)$	$\pi\beta$	$\pi\alpha_2$	$\tilde{a}_0$	$a_6 \tilde{a}_6 \leq \tilde{a}_0$
a_7	0	$p \mathrel{-\!\!*} q$	$\pi\alpha$	$\pi\beta_1$	a_6	$a_6 a_7 = \tilde{a}_7$
a_8	1	p	$-$	$\pi\alpha_1$	a_7	$-$
a_9	0	q	$-$	$\pi\alpha_2$	$\tilde{a}_7$	$-$
a_{10}	0	$p \mathrel{-\!\!*} r$	$\pi\alpha$	$\pi\beta_2$	$\tilde{a}_6$	$\tilde{a}_6 a_{10} = \tilde{a}_{10}$
a_{11}	1	p	$-$	$\pi\alpha_1$	a_{10}	$-$
a_{12}	0	r	$-$	$\pi\alpha_2$	$\tilde{a}_{10}$	$-$

Figure 4. Indexed formula tree for $(p \mathrel{-\!\!*} (q * r)) \mathrel{-\!\!*} ((p \mathrel{-\!\!*} q) * (p \mathrel{-\!\!*} r))$

Let us build a proof net following our algorithm. The only variable on a leaf is a_1 the position of which is a_2. There are two possibilities: either we connect it with the position a_{11} or with the position a_8. Let us try to connect with the position a_{11} (with label a_{10}). We obtain the MILL-substitution $\sigma(a_1) = a_{10}$. Then, we consider the position a_4 where $lsf(a_4) = a_3 : q^1$ and try to connect it. The only possibility is a_9 with the label $\tilde{a}_7$ that does not belong to the set of variables Σ_β but to the set of constants Σ_α. As we have $\sigma : \Sigma_\beta \rightarrow {\Sigma_\alpha}^\star$, we obtain a failure.



Let us erase the previous connections and try now to connect a_2 with a_8, having the label a_7. We deduce $\sigma(a_1) = a_7$.



As in the previous case, no connection is possible because the set of non-treated leaves is included in Σ_α and then it is not possible to build connections. Therefore, we conclude that the formula is not valid in MILL.
From the indexed labelled tree and the connection in the second case (and consequently from the MILL-substitution), we can build a countermodel. The forcing relation is defined as follows: for p : $a_7 \models p$ and $a_{10} \models p$, for $q : a_3 \models q$, for $r : \tilde{a}_3 \models r$.
In order to check, we can show that $a_0 \models p -\!\!* (q * r)$ and $\tilde{a}_0 \not\models (p -\!\!* q) * (p -\!\!* r)$. We have $a_3 \models q$ and $\tilde{a}_3 \models r$ and then $a_3\tilde{a}_3 \models q * r$. As $a_3\tilde{a}_3 = \tilde{a}_1$ we deduce $\tilde{a}_1 \models q * r$. Moreover, we have $a_7 \models p$ and, from $a_0a_1 = \tilde{a}_1$ and $\sigma(a_1) = a_7$, we deduce that $a_0a_7 = \tilde{a}_1$ and consequently $a_0a_7 \models q * r$ and $a_0 \models p -\!\!* (q * r)$. To show that $\tilde{a}_0 \not\models (p -\!\!* q) * (p -\!\!* r)$, we must show either $a_6 \not\models p -\!\!* q$ or $\tilde{a}_6 \not\models p -\!\!* r$ because $a_6\tilde{a}_6 = \tilde{a}_0$. We have $a_{10} \models p$ and thus $\tilde{a}_6a_{10} \not\models r$ since $\tilde{a}_6a_{10} = \tilde{a}_{10}$ and $\tilde{a}_{10} \not\models r$.

We have generated a countermodel of $(p -\!\!* (q * r)) -\!\!* ((p -\!\!* q) * (p -\!\!* r))$ from the labelled formula tree with existing connections before the failure of the construction attempts. Such a structure can be seen as a graphical representation of the countermodel like it is the case with a resource graph with the related tableau method in BI [7].

Our new connection method for MILL, that corresponds to a method of proof nets construction, is well-adapted to avoid several kinds of redundancy one has to deal with in more standard backward reasoning methods. In addition, it allows to efficiently detect the non-provability of formulae because the initial labelled formula tree contains the necessary semantic information. It also generate a countermodel in case of non-provability. It is a key point of the approach to say that such labelled proof structures or nets eliminate some bureaucracy from deductive systems but also are central structures with enough semantics in order to generate either proofs or countermodels.

References

1. M. Abrusci and P. Ruet. Non-commutative logic I : the multiplicative fragment. *Annals of Pure and Applied Logic*, 101:29–64, 2000.
2. W. Bibel. On matrices with connections. *Journal of ACM*, 28(4):633–645, 1981.
3. C. Faggian and M. Hyland. Designs, disputes and strategies. In *16th Int. Workshop on Computer Science Logic, CSL 2002, LNCS 2471*, pages 442–457, September 2002. Edinburgh, Scotland.
4. D. Galmiche. Connection Methods in Linear Logic and Proof nets Construction. *Theoretical Computer Science*, 232(1-2):231–272, 2000.
5. D. Galmiche and D. Méry. Connection-based proof search in propositional BI logic. In *18th Int. Conference on Automated Deduction, CADE-18, LNAI 2392*, pages 111–128, 2002. Copenhagen, Danemark.
6. D. Galmiche and D. Méry. Semantic labelled tableaux for propositional BI without bottom. *Journal of Logic and Computation*, 13(5):707–753, 2003.
7. D. Galmiche and D. Méry. Resource graphs and countermodels in resource logics. In *Int Workshop on Disproving: Non-theorems, Non-validity, Non-Provability*, pages 59–75, Cork, Ireland, July 2004.
8. D. Galmiche, D. Méry, and D. Pym. Resource Tableaux (extended abstract). In *16th Int. Workshop on Computer Science Logic, CSL 2002, LNCS 2471*, pages 183–199, September 2002. Edinburgh, Scotland.
9. D. Galmiche and J.M. Notin. Connection-based Proof Construction in Non-commutative Logic. In *10th Int. Conference on Logic for Programming, Artificial Intelligence, and Reasoning, LPAR'03, LNCS 2850*, pages 422–436, September 2003. Almaty, Kazakhstan.
10. D. Galmiche and G. Perrier. Foundations of proof search strategies design in linear logic. In *Logic at St Petersburg '94, Symposium on Logical Foundations of Computer Science, LNCS 813*, pages 101–113, St Petersburg, Russia, July 1994.
11. J.Y. Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–102, 1987.
12. 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.
13. C. Kreitz and H. Mantel. A matrix characterization for multiplicative exponential linear logic. *Journal of Automated Reasoning*, 32(2):121–166, 2004.
14. 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.
15. D. Méry. *Preuves et Sémantiques dans des Logiques de Ressources.* PhD thesis, Université Henri Poincaré, Nancy 1, 2004.
16. A. S. Murawski and C.-H. L. Ong. Fast verification of MLL proof nets via IMLL. *ACM Transactions on Computational Logic*, 2004. To appear.
17. P.W. O'Hearn and D. Pym. The Logic of Bunched Implications. *Bulletin of Symbolic Logic*, 5(2):215–244, 1999.
18. D.J. Pym. *The Semantics and Proof Theory of the Logic of Bunched Implications*, volume 26 of *Applied Logic Series.* Kluwer Academic Publishers, 2002.
19. D.J. Pym, P.W. O'Hearn, and H. Yang. Possible worlds and resources: The semantics of BI. *Theoretical Computer Science*, 315(1):257–305, 2004.
20. J. Reynolds. Separation logic: A logic for shared mutable data structures. In *IEEE Symposium on Logic in Computer Science*, pages 55–74, Copenhagen, Danemark, July 2002.
21. A. Urquhart. Semantics for Relevant Logic. *Journal of Symbolic Logic*, 37:159–169, 1972.
22. L.A. Wallen. *Automated Proof search in Non-Classical Logics.* MIT Press, 1990.