Concurrent Realizability on Conjunctive Structures

Emmanuel Beffara
Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, 38000 Grenoble, France

Félix Castro
IRIF, Université Paris Cité, France
IMERL, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay

Mauricio Guillermo
IMERL, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay

Étienne Miquey
Aix-Marseille Université, CNRS, I2M, Marseille, France

Abstract

This work aims at exploring the algebraic structure of concurrent processes and their behavior independently of a particular formalism used to define them. We propose a new algebraic structure called conjunctive involutive monoidal algebra (CIMA) as a basis for an algebraic presentation of concurrent realizability, following ideas of the algebrization program already developed in the realm of classical and intuitionistic realizability. In particular, we show how any CIMA provides a sound interpretation of multiplicative linear logic. This new structure involves, in addition to the tensor and the orthogonal map, a parallel composition. We define a reference model of this structure as induced by a standard process calculus and we use this model to prove that parallel composition cannot be defined from the conjunctive structure alone.

2012 ACM Subject Classification Theory of computation → Process calculi; Theory of computation → Proof theory; Theory of computation → Linear logic

Keywords and phrases Realizability, Process Algebras, Concurrent Processes, Linear Logic

Digital Object Identifier 10.4230/LIPIcs.FSCD.2023.28

Related Version *Extended Version*: `https://hal.science/hal-04083002`

Funding This work was partially supported by projects STIC-AmSud *Qapla'* and *APCoRe*.

1 Introduction

1.1 Realizability and its algebrization

Realizability provides a well-established and general set of techniques for studying the relationships between programs and proofs. In the traditional presentation of intuitionistic realizability, one starts from a set A of realizers, which are objects with computational meaning, programs in some formalism (codes of recursive functions, λ-terms, etc). Logic is then interpreted in the powerset of A, in such a way that the meaning of a formula is essentially a set of programs sharing a computational behavior dictated by the formula. From an algebraic viewpoint, the set A induces in its powerset a *Heyting algebra*, which in turn induces a *topos* [12, 20].

Classical realizability adapts these principles to classical logic, building on different foundations [13]. The computational part is based on the duality between programs (potential proofs) and environments (counter-proofs). The proper categorical structure underlying classical realizability was discovered by Streicher [22] and involves an *ordered combinatory algebra* (OCA) induced by terms and stacks. This construction was later generalized in a different setting, namely Krivine ordered combinatory algebras [6], with the feature that

Streicher's construction can be carried out in a purely axiomatic context. In particular, both truth values and realizers belong to the underlying set of these structures and the partial order subsumes subtyping, term reduction and the realizability relation. A similar approach was later followed within Miquel's implicative algebras [15] (and some variants by the last author's [16]), a slightly different structure which allows to encompass intuitionistic and classical models of forcing and realizability into a general framework, keeping the feature that truth values and realizers (and also forcing conditions) both belong to the underlying set of the implicative algebra. All these works therefore advocate for foundations of realizability seeking to transform a rather complex and operational definition into a much simpler and algebraic one.

1.2 Logic for concurrency

Process calculi form a wide range of formalisms designed to model concurrent systems and reason about them by means of term rewriting. Their applications are diverse, from the semantics of proof systems to the conception of concrete programming languages. Type systems for such calculi are therefore a wide domain, with systems of different kinds designed to capture different behaviors and ensure different properties of processes: basic interfacing, input-output discipline, linearity, lock-freeness, termination, respect of communication protocols, etc. To better understand the diversity of calculi and uncover basic structures and general patterns, many authors have searched for languages with simpler or more general theory in which the most features could be expressed by means of restrictions or encodings. Similar unification has been searched for in the realm of types, but no system can yet claim to be as basic and universal as, *e.g.*, simple types for the λ-calculus, which perfectly capture the abstraction and application mechanism in the logical world.

Concurrent realizability, developed by Beffara [2], transposes the ideas of classical realizability in a context of interacting concurrent processes. Following the constructions of phase semantics from linear logic [8] and ludics [9], the combinatory algebra is replaced with a variant of the π-calculus, endowed with a structure of commutative monoid under parallel composition. The pole, which defines orthogonality as in classical realizability, is seen as a testing protocol between processes, and the sets of processes that are closed by biorthogonality act both as truth values and behaviors. However, in the original construction, processes are subject to polarization and interfacing constraints which restrict the use of names, in order to avoid ambiguity when composing and to ensure consistency. As a consequence, operations over behaviors inherit such restrictions, which makes them partial in general.

The aim of the present work is to set the basis for an algebraic presentation of concurrent realizability, as a way to study processes and their types in a well-structured algebraic framework. We propose that, as with sequential models, the algebraic presentation is based on an ordered algebraic structure which must allow interpreting truth values and concurrent programs in its underlying set, subsuming the reduction semantics in the order relation. Furthermore, we want to avoid imposing a priori restrictions on processes.

1.3 Principles of the construction

We aim for an algebraic structure in which we can represent processes, types and operators in a uniform setting and we will refer to elements of such a structure as *behaviors*. A guiding intuition is to see behaviors as particular sets of terms in a process algebra, with some notion of closure by observational equivalence (a formal implementation is presented in Section 4). We postulate three fundamental structures over behaviors:

- a complete lattice structure, taking a comparison $a \leq b$ to mean that b exhibits more possible behaviors than a;
- a binary operator $\otimes$, continuous with respect to the lattice structure, that represents an operation of parallel composition without interaction.
- an anti-monotone unary operator $(\cdot)^\perp$ s.t. $a^\perp$ represents the tests that a passes.

This yields a variant of the last author's conjunctive structure [16], including notions of arrow and application and allowing for a sound interpretation of multiplicative linear logic. Within this framework, we study parallel composition as an additional continuous binary operation in the conjunctive structure.

Note that the notion of *name*, though pervasive in process algebra, is not taken to be primitive because different calculi and type systems make different choices on the use of names and actions (monadicity vs polyadicity, polarization, synchrony, etc). Instead, we consider that the multiplicative structure includes some way of managing connections between processes so that the axioms are satisfied.

1.4 Outline of the paper

We start by introducing in Section 2 the conjunctive structures that we identified to be conducive to the analysis of concurrent computation from an algebraic viewpoint. In Section 3, we develop a construction over process algebras to formally extend a language of processes with generalized fusions (PGF), on top of which we build a realizability model in Section 4 with the expected algebraic structure. We then explain in Section 5 how to equip conjunctive structures with parallelism and we illustrate it in the case of PGF. In Section 6, we prove that parallel composition cannot have an internal representation in the axioms of purely conjunctive structures, using a model that validates these axioms but does not have an operator for general parallel compisition.

2 Logic in conjunctive structures

Following the program of algebrization of realizability models that was mostly undertaken in the realm of Krivine realizability [22, 6, 15, 16] and previous work by the first author on concurrent computation [2], we introduce a particular class of *conjunctive algebras* which we identify as the key algebraic structures underlying realizability models induced by process calculi. We first define the notion of *conjunctive structure*, which reflects the algebraic structure of these models (*how are the truth values defined?*), and then define a notion of *separator* that allows to capture the logical content (*which processes define valid realizers?*).

2.1 Conjunctive structures

▶ **Definition 1.** *A conjunctive structure (CS) is a tuple* $(\mathbb{C}, \preccurlyeq, \otimes, (\cdot)^\perp)$ *such that*

1. $(\mathbb{C}, \preccurlyeq)$ *is a complete lattice;*
2. $\otimes$ *is a binary monotone operation of* $\mathbb{C}$ *and* $(\cdot)^\perp$ *is a unary antimonotone function on* $\mathbb{C}$*;*
3. $\otimes$ *distributes over the join operation* $\curlyvee$*, i.e. for any* $a \in \mathbb{C}$ *and* $\mathfrak{B} \subseteq \mathbb{C}$ *we have* $\curlyvee_{b\in\mathfrak{B}}(a \otimes b) = a \otimes \left(\curlyvee_{b\in\mathfrak{B}} b\right)$ *and* $\curlyvee_{b\in\mathfrak{B}}(b \otimes a) = \left(\curlyvee_{b\in\mathfrak{B}} b\right) \otimes a$*;*
4. *the orthogonal map* $(\cdot)^\perp$ *satisfies De Morgan's law* $\left(\curlyvee_{b\in\mathfrak{B}} b\right)^\perp = \curlywedge_{b\in\mathfrak{B}} b^\perp$.

We say that the structure is involutive *(CIS) if* $(\cdot)^\perp$ *is involutive:* $a^{\perp\perp} = a$ *for all* $a \in \mathbb{C}$*. Also we call* unitary *every CS (resp. CIS) with a distinguished element* $1 \in \mathbb{C}$.

To draw the comparison with the conjunctive structures defined in the last author's work [16], the only difference lies in the use of an orthogonal map instead of a negation $\neg$ that was meant to convey a computational content. In particular, even in the classical case, a and $\neg\neg a$ are logically equivalent but not necessarily equal.

▶ **Example 2.** Any complete Boolean algebra $(\mathbb{B}, \preccurlyeq, \wedge, \vee, \neg)$ defines a CIS using the conjunction as tensor ($a \otimes b \triangleq a \wedge b$) and the negation for orthogonal map ($a^\perp \triangleq \neg a$).

▶ **Example 3.** A phase space [8] is defined by a commutative monoid M and a subset $\perp \subseteq M$. For $a \subseteq M$, define the dual $a^\perp \triangleq \{x : \forall y \in a, xy \in \perp\}$; define $a \otimes b \triangleq \{xy : x \in a, y \in b\}^{\perp\perp}$ for $a, b \subseteq M$. Then the set of subsets $a \subseteq M$ such that $a^{\perp\perp} = a$ forms a CIS.

Given a conjunctive structure $(\mathbb{C}, \preccurlyeq, \otimes, (\cdot)^\perp)$, we define the usual connectives and quantifiers of multiplicative linear logic (MLL) on $\mathbb{C}$ as follows, where F is a function over $\mathbb{C}$:

$$a \parr b \triangleq (a^\perp \otimes b^\perp)^\perp \qquad a \multimap b \triangleq (a \otimes b^\perp)^\perp \qquad \exists F \triangleq \bigcurlyvee_{a \in \mathbb{C}} F(a) \qquad \forall F \triangleq \bigcurlywedge_{a \in \mathbb{C}} F(a)$$

These definitions induce a canonical interpretation of MLL formulas within any CIS.

▶ **Definition 4.** *Consider a unitary CS $(\mathbb{C}, \preccurlyeq, \otimes, (\cdot)^\perp, 1)$. The interpretation of closed MLL formulas with parameters $P \in \mathbb{C}$ is defined as follows:*

$$\begin{array}{lll} [\![\mathbb{1}]\!] \triangleq 1 & [\![A \otimes B]\!] \triangleq [\![A]\!] \otimes [\![B]\!] & [\![\exists X.A]\!] \triangleq \exists\big(P \mapsto [\![A]\!]\{X := P\}\big) \\ [\![\perp]\!] \triangleq 1^\perp & [\![A \parr B]\!] \triangleq [\![A]\!] \parr [\![B]\!] & [\![\forall X.A]\!] \triangleq \forall\big(P \mapsto [\![A]\!]\{X := P\}\big) \\ [\![P]\!] \triangleq P & & \end{array}$$

Given a sequent $\vdash A_1, \ldots, A_k$, the interpretation $[\![A_1, \ldots, A_k]\!]$ is defined as $[\![A_1]\!]$ if $k = 1$ and $[\![A_1]\!] \parr [\![A_2, \ldots, A_k]\!]$ otherwise.

▶ Remark 5. As usual, the interpretation $[\![A]\!]$ of a formula A depends only upon the assignment of the free variables of A. The same is valid for sequents.

▶ Remark 6. If $\mathbb{C}$ is a CIS, the involutivity of $(\cdot)^\perp$ actually implies that $(\mathbb{C}, \preccurlyeq, \multimap)$ is an implicative structure in the sense of Miquel [15] since $\multimap$ satisfies the expected variance and continuity properties: in particular $\big(\bigcurlyvee_{b \in B} b\big) \multimap a = \bigcurlywedge_{b \in B}(b \multimap a)$ for any $a \in \mathbb{C}$ and $B \subseteq \mathbb{C}$.

Note that this applies to *structures*, which only define operators and their relationships. The *algebras* defined below, which include a notion of logical validity, will actually diverge since implicative algebras model intuitionistic logic, while our algebras apply to linear logic.

2.2 Separators and internal logic

Conjunctive structures are suited to interpret formulas of MLL. To account for a validity notion, we need to introduce the notion of separator [15, 16], a subset of the structure which intuitively distinguishes valid formulas, similarly to a filter in a Boolean algebra. Separators are built upon a set of combinators that one can understand as axioms for the internal logic.

▶ **Definition 7.** *Consider a unitary CS $(\mathbb{C}, \preccurlyeq, \curlyvee, \otimes, (\cdot)^\perp, 1)$. The MLL combinators are:*

- $S_3 \triangleq \bigcurlywedge_{a,b \in \mathbb{C}} (a \otimes b) \multimap (b \otimes a)$
- $S_4 \triangleq \bigcurlywedge_{a,b,c \in \mathbb{C}} (a \multimap b) \multimap (a \otimes c) \multimap (b \otimes c)$
- $S_5 \triangleq \bigcurlywedge_{a,b,c \in \mathbb{C}} ((a \otimes b) \otimes c) \multimap (a \otimes (b \otimes c))$
- $S_6 \triangleq \bigcurlywedge_{a \in \mathbb{C}} a \multimap (1 \otimes a)$
- $S_7 \triangleq \bigcurlywedge_{a \in \mathbb{C}} (1 \otimes a) \multimap a$
- $S_8 \triangleq \bigcurlywedge_{a,b \in \mathbb{C}} (a \multimap b) \multimap (b^\perp \multimap a^\perp)$

The definitions (and names) for S_3, S_4 and S_5 come from the notion of separator in conjunctive algebras [16]. The original S_1 and S_2 are replaced here by S_6, S_7 and S_8 to account for linearity. The set of combinators $\{S_3, \ldots, S_8\}$ is known to be complete for MLL [1].

$$\frac{}{\vdash 1 : \mathbb{1}}\,(\mathbb{1}) \qquad \frac{}{\vdash \mathbf{I} : A^{\perp}, A}\,(\textsc{Ax}) \qquad \frac{\vdash a : A_1, \dots, A_k}{\vdash \mathbf{ex}(\sigma) * a : A_{\sigma(1)}, \dots, A_{\sigma(k)}}\,(\textsc{Ex}) \qquad \frac{\vdash a : \Gamma, A \quad \vdash b : B, \Delta}{\vdash \mathbf{t} * a * b : \Gamma, A \otimes B, \Delta}\,(\otimes)$$

$$\frac{\vdash a : \Gamma, A \quad \vdash b : A^{\perp}, \Delta}{\vdash \mathbf{c} * a * b : \Gamma, \Delta}\,(\textsc{Cut}) \qquad \frac{\vdash a : \Gamma, A\{X := B\}}{\vdash a : \Gamma, \exists X.A}\,(\exists) \qquad \frac{\vdash a : \Gamma, A \quad X \text{ not free in } \Gamma}{\vdash a : \Gamma, \forall X.A}\,(\forall)$$

Figure 1 Semantic typing rules for MLL.

▶ **Definition 8.** *Let us consider* $(\mathbb{C}, \preccurlyeq, \curlyvee, \otimes, (\cdot)^{\perp}, 1)$ *a unitary CIS. A* monoidal separator *on* $\mathbb{C}$ *is an upwards closed set* $\mathcal{S} \subseteq \mathbb{C}$ *such that:*

- $\mathcal{S}$ *contains the MLL-combinators* $S_3, \dots, S_8$.
- *For any* $a, b \in \mathbb{C}$, *if* $a \multimap b \in \mathcal{S}$ *and* $a \in \mathcal{S}$ *then* $b \in \mathcal{S}$.

A conjunctive involutive monoidal algebra (CIMA) is a unitary CIS $\mathbb{C}$ *together with a monoidal separator* $\mathcal{S}$.

▶ **Example 9.** In any CIS induced by a complete Boolean algebra $\mathbb{B}$ (see Example 2), all the combinators are tautologies trivially interpreted by the maximal element $\top$. Therefore, the singleton $\{\top\}$ (or alternatively any filter on $\mathbb{B}$) defines a separator for this CIS.

▶ **Definition 10.** *Let* $\mathbb{C}$ *be a CS. A* semantic judgement *is a statement* $\vdash a : \Gamma$ *where* $a \in \mathbb{C}$ *and* Γ *is a sequent of formulas with parameters in* $\mathbb{C}$. *Such a judgement is* sound *if* $a \preccurlyeq [\![\Gamma]\!]$. *A* semantic typing rule *is an inference rule where the premises and conclusion are semantic judgements, possibly involving free variables. Such a rule is* sound *if the soundness of its premises entails that of its conclusion for any instantiation of the variables in* $\mathbb{C}$.

We provide semantic typing rules for MLL in Figure 1, Using the following terms, where $\sigma : [1;k] \to [1;k]$ is a permutation and $a * b \triangleq \curlywedge\{c \in \mathbb{C} \mid a \preccurlyeq b \multimap c\}$ is the usual application from implicative algebras:

$$\begin{aligned}
\mathbf{I} &\triangleq \curlywedge_{a \in \mathbb{C}} (a \multimap a)\\
\mathbf{t} &\triangleq \curlywedge_{a,b,g,d \in \mathbb{C}} ((g \parr a) \multimap (b \parr d) \multimap g \parr ((a \otimes b) \parr d))\\
\mathbf{c} &\triangleq \curlywedge_{a,g,d \in \mathbb{C}} ((g \parr a) \multimap (a^{\perp} \parr d) \multimap (g \parr d))\\
\mathbf{ex}(\sigma) &\triangleq \curlywedge_{a_1,\dots,a_k \in \mathbb{C}} \big((a_1 \parr \cdots \parr a_k) \multimap (a_{\sigma(1)} \parr \cdots \parr a_{\sigma(k)})\big)
\end{aligned}$$

▶ **Proposition 11.** *For any CIMA* $(\mathbb{C}, \mathcal{S})$ *and any permutation* σ, *the combinators* $\mathbf{I}$, $\mathbf{t}$, $\mathbf{c}$ *and* $\mathbf{ex}(\sigma)$ *belong to the separator* $\mathcal{S}$

Proof. These proofs are essentially simple combinatorial manipulations within conjunctive involutive structures. We detail here the proof that $\mathbf{I} \in \mathcal{S}$ as an example. Let us consider a CIMA $(\mathbb{C}, \mathcal{S})$ and define:

$$S_4' \triangleq \curlywedge_{a,b,c \in \mathbb{C}} (a \multimap b) \multimap (b \multimap c) \multimap a \multimap c \qquad\qquad t \circ s \triangleq S_4' * s * t$$

It is an easy exercise to check that $S_4' \in \mathcal{S}$ (this follows from combinatorial manipulations using the closure under modus ponens and the fact that $S_4, S_8 \in \mathcal{S}$). Besides, similarly to what happens for implicative algebras, separators of CIMAs are closed under application $*$, and as such they are also closed under composition $\circ$. To prove that $\mathbf{I} \in \mathcal{S}$, observe that for any $a \in \mathbb{C}$ we have $S_6 \preccurlyeq a \multimap a \otimes \mathbf{1}$ and $S_7 \preccurlyeq a \otimes \mathbf{1} \multimap a$. Then $S_7 \circ S_6 \preccurlyeq a \multimap a$ (uniformly on $a \in \mathbb{C}$) and thus we get $S_7 \circ S_6 \in \mathcal{S}$ and then $\curlywedge_{a \in \mathbb{C}} a \multimap a \in \mathcal{S}$. ◀

▶ **Theorem 12.** *The semantic typing rules of Figure 1 are sound for all unitary CS. Moreover, for any CIMA $(\mathbb{C}, \mathcal{S})$ if $\vdash a : \Gamma$ is provable, then $a \in \mathcal{S}$.*

Proof. The soundness of rule $(\mathbb{1})$ is just reflexivity of $\preccurlyeq$ and that of rules (Ax), $(\forall)$ and $(\exists)$ holds by definition of the meet operation. For the other rules, remark that if for some $a, b, t, u \in \mathbb{C}$ we have $t \preccurlyeq a \multimap b$ and $u \preccurlyeq a$, then $t \preccurlyeq u \multimap b$, hence $t * u \preccurlyeq b$ by definition of $*$. This entails the soundness of (Ex), $(\otimes)$ and (Cut) by definition of $\mathbf{I}$, $\mathbf{t}$ and $\mathbf{c}$.

Besides, by definition of $*$, we also have $a \preccurlyeq b \multimap a * b$ for all a and b, which implies that separators are closed under $*$. This, with the results of Proposition 11, entails that the left-hand sides of derivable judgements are always in $\mathcal{S}$. ◀

As a consequence, we conclude that all provable MLL formulas are interpreted by elements in the separator (i.e. validated by the model).

As stated in Remark 6, unitary CISs are implicative structures while in general CIMAs are not implicative algebras. Due to the chosen notion of separator, CIMAs do not (in general) model intuitionistic logic. Nevertheless, some specific CIMAs can model intuitionistic or classical logics if their separators contains non linear terms.

3 Processes with global fusions

In this section, we define a computational structure that will serve as our reference. We first recall the basic definitions of a standard π-calculus, then we build processes with fusions as a formal extension of this language. These fusions, being an additional structure to extend the expressiveness of an existing algebra (as opposed to a feature that we would impose on the underlying calculus, which would restrict the range of calculi our study applies to), bring the necessary connections to realize MLL axioms.

Hence, it should be made clear that the point of this section is not to define "yet another π-calculus" but, on the contrary, to show that our study applies to any process calculus. The only technical constraint is that the combinators from Definition 7 should not be degenerate (i.e. equal to $\bot$), otherwise there is no meaningful seperator. This essentially requires to have realizers that act as substitutions (or "forwarders"). The construction below consists in formally completing a process algebra with such objects.

So the choice of a particular variant of the π-calculus is essentially arbitrary, we pick this one for the purpose of illustration. The only features we need are the ability to perform renaming and the availability of parallel composition and hiding with usual properties. The point of the construction is to extend the language with name fusions at top-level while preserving this composition structure. Keeping fusions at top-level is the only generic choice that does not impose to reason on the syntax of the calculus we build on, thus allowing for working with fusions in the algebraic structure without imposing constraints on the underlying process language.

3.1 Basic processes

The following recalls the standard definitions of the polyadic asynchronous π-calculus with its operational semantics by reduction [21]. We leave out replication and addition because these will not be used in this paper, but including them is not a problem.

▶ **Definition 13.** *The set Π of π-processes is defined by the following grammar:*

$$P, Q ::= \mathbb{1} \mid (P \mid Q) \mid u(\vec{x}).P \mid \bar{u}\,\langle\vec{v}\rangle \mid (\nu y)P$$

where $\vec{x}$ is a finite sequence of pairwise distinct names that range over the set $\mathbb{N}$ of natural numbers, $\vec{v}$ is a finite sequence of names, y and u are names. The term $\mathbb{1}$ is the unit *and the operators $|, u(\vec{x}), \bar{u}\langle v\rangle, (\nu y)$ are respectivelly called* parallel composition, reception, emission *and* hiding.

The names in $\vec{x}$ are bound in $u(\vec{x}).P$, the name y is bound in $(\nu y)P$. The set of free names *of a term P is denoted by $\mathrm{FN}(P)$. Terms are considered up to renaming of bound names, also called α-conversion, written $\equiv_\alpha$.*

Structural equivalence *is the smallest congruence over terms such that*

- $(\Pi, |, \equiv)$ *is an commutative monoid, i.e.* $\mathbb{1}|P \equiv P$, $P|Q \equiv Q|P$ *and* $(P|Q)|R \equiv P|(Q|R)$,
- $P \equiv (\nu x)P$ *and* $P \mid (\nu x)Q \equiv (\nu x)(P \mid Q)$ *whenever* $x \notin \mathrm{FN}(P)$,
- $(\nu x)(\nu y)P \equiv (\nu y)(\nu x)P$ *and we write* $(\nu xy)P \triangleq (\nu x)(\nu y)P$.

▶ **Definition 14.** *A* substitution *is a function $\sigma : \mathbb{N} \to \mathbb{N}$. Provided that the names in $\vec{x}$ are pairwise distinct and that $\vec{x}$ and $\vec{v}$ have the same length, $\{\vec{x} := \vec{v}\}$ denotes the substitution which replaces $\vec{x}$ by $\vec{v}$ and leaves all other names unchanged.*

Given $X \subseteq \mathbb{N}$ we denote by $\sigma_{\restriction X}$ the substitution which coincides with σ on X and is the identity outside. We write $\sigma \backslash X$ for the substitution $\sigma_{\restriction X^c}$.

We denote by P^σ the result of applying σ to $P \in \Pi$, defined inductively in the standard way. Given substitutions $\sigma_1, \dots, \sigma_k$ we denote by $P^{\sigma_1 \cdots \sigma_k}$ the term $(\dots(P^{\sigma_1})\dots)^{\sigma_k}$.

▶ **Definition 15.** *One-step reduction is the smallest binary relation $\longrightarrow_1$ over Π such that*

- $\bar{u}\langle\vec{v}\rangle \mid u(\vec{x}).P \longrightarrow_1 P^{\{\vec{x}:=\vec{v}\}}$,
- $\longrightarrow_1$ *is compatible with* $\equiv$, *i.e. if* $P \equiv P'$, $Q \equiv Q'$ *and* $P \longrightarrow_1 Q$ *then* $P' \longrightarrow_1 Q'$,
- $\longrightarrow_1$ *is compatible with parallel composition and hiding, i.e. if* $x \in \mathbb{N}$ *and* $P \longrightarrow_1 P'$, *then* $P \mid Q \longrightarrow_1 P' \mid Q$ *and* $(\nu x)P \longrightarrow_1 (\nu x)P'$.

We do not define a higher-level semantics at this point. The semantics induced by the conjunctive structures defined in the following sections will be a generic form of testing and most of the construction will be parametric in the particular testing protocol.

3.2 Processes with fusions

A *fusion* is, intuitively, a connection between channels. The defining property of a fusion $u{\leftrightarrow}v$ is that its presence within a process allows actions on channel u to synchronize with actions on channel v. Several works have defined extensions of the π-calculus with this feature [7, 19, 23] and studied its effects on the theory and the expressiveness of the language.

Instead, the construction presented in this section rather aims at extending an arbitrary process calculus with fusions *without* affecting its theory, for our study to be compatible with any particular process language. This is made possible by the fact that we only need fusions at top-level, so that they can live side-by-side with standard processes. Intuitively, a fusion $u{\leftrightarrow}v$ at top level in a process can be interpreted as the information that u and v will eventually get substituted with the same name.

▶ **Definition 16.** *A fusion is an equivalence relation over $\mathbb{N}$. The set of fusions is written $\mathcal{E}$. Given $e \in \mathcal{E}$ we write $x \underset{e}{\sim} y$ instead of $(x, y) \in e$. For $e, f \in \mathcal{E}$ we write ef for the smallest equivalence that contains both e and f.*

▶ Remark 17. The set $\mathcal{E}$ is a complete lattice under inclusion, with the identity $\Delta_\mathbb{N} \triangleq \{(x,x) \mid x \in \mathbb{N}\}$ as minimum and the full relation $\nabla_\mathbb{N} \triangleq \mathbb{N} \times \mathbb{N}$ as maximum. The meet of a set $S \subseteq \mathcal{E}$ is the intersection $\bigcap S$ and its join $\bigvee S$ is the transitive closure of the union $\bigcup S$.

▶ **Definition 18.** *Given $e \in \mathcal{E}$, $x, y \in \mathbb{N}$, $X \subseteq \mathbb{N}$ and $\sigma : \mathbb{N} \to \mathbb{N}$, define*

- *the* equivalence class *of x as $[x]_e \triangleq \{y \in \mathbb{N} \mid x \underset{e}{\sim} y\}$, extended as $[X]_e \triangleq \bigcup_{x \in X} [x]_e$;*
- *the* domain *of e as $|e| \triangleq \{x \in \mathbb{N} \mid [x]_e \neq \{x\}\}$;*
- *the* restriction *of e to X as $e \cap X \triangleq (e \cap (X \times X)) \cup \Delta_{\mathbb{N}}$;*
- *the* hiding *of X in e as $e \setminus X \triangleq e \cap X^c$;*
- *the* elementary fusion *of x and y as $x{\leftrightarrow}y \triangleq \{(x, y), (y, x)\} \cup \Delta_{\mathbb{N}}$.*

An equivalence e can be induced by a function σ, considering that equivalent elements are the ones with the same image under σ. Such a function can always be deduced from e by picking a representative in each equivalence class. The following definitions formalize these constructions, exploiting the well-ordering over $\mathbb{N}$ to get canonical representatives.

▶ **Definition 19.** *Given a fusion $e \in \mathcal{E}$, $x \in \mathbb{N}$ and a function $\tau : \mathbb{N} \to \mathbb{N}$, we define:*

- $x^{\bullet}_e \triangleq \min[x]_e$ *and* $x^*_e \triangleq \begin{cases} x & \text{if } [x]_e = \{x\} \\ \min([x]_e \backslash \{x\}) & \text{if } [x]_e \neq \{x\} \end{cases}$
- *$\sigma^{\bullet}_e : x \mapsto x^{\bullet}_e$ for all x, the substitution induced by e;*
- *$\varepsilon_\tau \triangleq \bigvee_{x \in \mathbb{N}} (x{\leftrightarrow}\tau(x))$, the fusion induced by τ;*
- *$e^\tau \triangleq \bigvee_{x \underset{e}{\sim} y} (\tau(x){\leftrightarrow}\tau(y))$, the composition of e with τ.*

We are now ready to introduce the notion of processes with global fusions, and extend the constructions on processes to this generalization. This is similar to that of unitization in algebras, where a non-unital algebra is extended with a formal extra element and saturated to preserve the structure and endow the extra element with desired properties.

▶ **Definition 20.** *The set of* processes with global fusions *(PGF) is defined as*

$$\bar{\Pi} \triangleq \Pi \times \mathcal{E} = \{(P, e) \mid P \in \Pi, e \in \mathcal{E}\}$$

Over $\bar{\Pi}$, define substitution, free names, α-equivalence and structural equivalence $\equiv$ as

$$\begin{aligned} (P, e)^\tau &\triangleq (P^\tau, e^\tau) & (P, e) \equiv_\alpha (Q, f) &\iff e = f \text{ and } P^{\sigma^{\bullet}_e} \equiv_\alpha Q^{\sigma^{\bullet}_f} \\ \mathrm{FN}(P, e) &\triangleq \mathrm{FN}(P) \cup |e| & (P, e) \equiv (Q, f) &\iff e = f \text{ and } P^{\sigma^{\bullet}_e} \equiv Q^{\sigma^{\bullet}_f} \end{aligned}$$

When it is not necessary to specify process and fusion, we will use the lowercase letters p, q, r, s to refer to processes with fusions.

Remark that the definitions of α-equivalence and structural equivalence allow renaming of free names as long as the equivalence class of each name is unchanged, so that we have for instance $(a(x).P, a{\leftrightarrow}b) \equiv (b(x).P, a{\leftrightarrow}b)$. Including $|e|$ in $\mathrm{FN}(P, e)$ ensures that, despite such renaming, FN is invariant under $\equiv_\alpha$ as expected. Note also that if we identify each term P with the pair $(P, \Delta_{\mathbb{N}})$, then the operations defined above coincide with those of Π.

▶ **Definition 21.** *One-step reduction in $\bar{\Pi}$ is the relation $\longrightarrow_1$ such that $(P, e) \longrightarrow_1 (Q, f)$ if and only if $e = f$ and $P^{\sigma^{\bullet}_e} \longrightarrow_1 Q^{\sigma^{\bullet}_f}$. Multistep reduction $\longrightarrow$ is the reflexive and transitive closure of $\longrightarrow_1$.*

Again, this extends the associated relation from Π to $\bar{\Pi}$ and allows more reductions when the fusion relates two names on which compatible actions occur. For instance we have $(u(x).P \mid \bar{v}\,\langle y \rangle\,, u{\leftrightarrow}v) \longrightarrow_1 (P\{x := y\}, u{\leftrightarrow}v)$.

▶ **Definition 22.** *For $(P, e), (Q, f) \in \bar{\Pi}$, define $(P, e) \mid (Q, f) \triangleq (P \mid Q, ef)$.*

Identifying a pure fusion e with the pair $(\mathbb{1}, e)$, we have $(P, e) \equiv P \mid e$ for every P, since $P \mid \mathbb{1} \equiv P$ in Π. So in this sense Π and $\mathcal{E}$ generate $\bar{\Pi}$ under parallel composition. Besides, reduction $\longrightarrow_1$ is compatible with $\equiv$ and parallel composition, as in the basic calculus.

We now need to extend name hiding to $\bar{\Pi}$. The result of $(\nu x)(P, e)$ must be (Q, f) where $x \notin \mathrm{FN}(Q, f)$, hence $x \notin \mathrm{FN}(Q) \cup |f|$. Consider a pair $(\nu x)(\bar{x}, x{\leftrightarrow}y)$: even if x must be bound in $(\nu x)(\bar{x}, x{\leftrightarrow}y)$, since x is fused with y, we have $(\bar{x}, x{\leftrightarrow}y) \equiv_\alpha (\bar{y}, x{\leftrightarrow}y)$ so the only reasonable definition for $(\nu x)(\bar{x}, x{\leftrightarrow}y)$ is $(\bar{y}, \Delta_\mathbb{N})$, which is structurally equivalent to $((\nu x)\bar{y}, \Delta_\mathbb{N})$. On the other hand, if we consider $(\nu x)(\bar{x}, \Delta_\mathbb{N})$, we expect to get simply $((\nu x)\bar{x}, \Delta_\mathbb{N})$. Definition 19 provides the notation x_e^* to represent this in a general way.

▶ **Definition 23.** *For $(P, e) \in \bar{\Pi}$ and $x \in \mathbb{N}$, define $(\nu x)(P, e) \triangleq \big((\nu x)P\{x := x_e^*\}, e \setminus \{x\}\big)$.*

▶ Remark 24. Whenever x is not equivalent to another name, *i.e.* $[x]_e = \{x\}$, we have $x = x_e^*$ and therefore the binder νx on a PGF process only computes as expected on the process side: $(\nu x)(P, e) = ((\nu x)P, e)$. If, instead, x is equivalent to other names, *i.e.* $[x]_e \neq \{x\}$, then $x \neq x_e^*$ and νx will disconnect x from its fusion-side equivalents, while replacing it by an equivalent name on the process side: $((\nu x)P\{x := x_e^*\}, e\backslash\{x\}) \equiv (P\{x := x_e^*\}, e\backslash\{x\})$.

To extend the definition of ν to finite sets, we first need to ensure that, up to α-equivalence, the order of application of ν is irrelevant.

▶ **Lemma 25.** *Consider $(P, e) \in \bar{\Pi}$ and $x, y \in \mathbb{N}$, then $(\nu x)(\nu y)(P, e) \equiv_\alpha (\nu y)(\nu x)(P, e)$.*

Proof. On the one hand

$$(\nu x)(\nu y)(P, e) = (\nu x)((\nu y)P\{y := y_e^*\}, e\backslash\{y\}) = ((\nu xy)P\{y := y_e^*\}\{x := x_{e\backslash\{y\}}^*\}, e\backslash\{x, y\}).$$

On the other hand, it can be shown that the last process is α-equivalent to

$$((\nu yx)P\{x{:=}x_e^*\}\{y{:=}y_{e\backslash\{x\}}^*\}, e\backslash\{x, y\}) = (\nu y)((\nu x)P\{x{:=}x_e^*\}, e\backslash\{x\}) = (\nu y)(\nu x)(P, e). \blacktriangleleft$$

▶ **Definition 26.** *Given a* finite *set of names $X = \{x_1, \dots, x_k\}$ with $x_1 < \cdots < x_k$, define $(\nu X)(P, e) \triangleq (\nu x_k) \cdots (\nu x_1)(P, e)$.*

Later on, we will need to extend this definition for a possibly infinite set X. Even if, for any PGF p, there exist only finitely many free names of p in X which we need to hide, for each of them we need to treat them differently depending on whether their equivalence class is included in X or not. The following definitions generalize the unary definition of ν in that regards.

▶ **Definition 27.** *Consider a set $X \subseteq \mathbb{N}$ and a fusion $e \in \mathcal{E}$. Define the substitution*

$$\mu_{(X,e)}(x) \triangleq \begin{cases} x_e^\bullet & \text{if } [x]_e \subseteq X \\ \min([x]_e \setminus X) & \text{if } [x]_e \setminus X \neq \emptyset \end{cases}$$

▶ **Definition 28.** *Let $(P, e) \in \bar{\Pi}$ and $X \subseteq \mathbb{N}$. Since $\mathrm{FN}(P)$ is finite, there exists classes $[x_1]_e, \dots, [x_{h+k}]_e$ in the quotient $\mathbb{N}/e$ such that for each $i \leq h$ we have $[x_i]_e \subseteq X$ and for each $j > h$ we have $[x_j] \setminus X \neq \emptyset$, and moreover $\mathrm{FN}(P) \cap X \subseteq \bigcup_{i=1}^{h+k} [x_i]_e$. Define*

$$(\nu X)(P, e) \triangleq \big((\nu x_{1,e}^\bullet, \dots, x_{h,e}^\bullet)P^{\mu_{(X,e)}}, e \setminus X\big)$$

Let us denote by $\bar{\nu}$ the binder $(\nu\mathbb{N})$.

Finally, let us state a technical result that emphasizes how we can embed a substitution within the calculus by means of the corresponding fusion.

▶ **Proposition 29.** *Consider a substitution $\tau : X \to X^c$ for a set $X \subset \mathbb{N}$ and $p, q \in \bar{\Pi}$ such that $\mathrm{FN}(p) \subseteq X$ and $\mathrm{FN}(q) \subseteq X^c$. Then $(\nu X)(p \,|\, q \,|\, \varepsilon_\tau) \equiv_\alpha p^\tau \mid q$.*

We defer the proof to the extended version of this paper, as it requires a number of technical lemmas whose statements and proofs are of little interest by themselves. The property above will be a crucial ingredient later on to build appropriate realizers. Obtaining it is the point of the PGF construction and we will not rely on more details when studying the conjunctive structure.

4 Conjunctive structure of processes

Following the first author's work [2], we shall now see how to define realizability models which interpret MLL formulas as sets of processes. These models actually induce a conjunctive involutive monoidal algebra and as such provides a sound interpretation of MLL. In particular, we would like to emphasize that this illustrates a significant benefit of our approach, which allows us to isolate logical properties that are shared by *all* concurrent realizability models sharing this algebraic structure, ensuring that these properties are indeed insensitive to implementation details such as the choice of a particular syntax for processes.

4.1 Concurrent realizability with PGF

We build upon the main insight of Krivine's realizability [14], by parameterizing the interpretation by a *pole*, a set of processes somewhat characterizing valid interactions between terms. In this setting, we will consider poles that are simply closed under structural equivalence to characterize the soundness of interaction with respect to the renaming mechanism that is necessary for pure parallel composition, without focusing so far on the reduction of processes. From now on, we denote by $\mathbb{P}$ the set of all sets of processes $\mathbb{P} \triangleq \mathcal{P}(\bar{\Pi})$.

▶ **Definition 30.** *A* pole *is a set of closed processes $\perp\!\!\!\perp \subseteq \{p \in \bar{\Pi} \mid \mathrm{FN}(p) = \emptyset\}$ that is closed under $\equiv$,* i.e. *if $p \equiv q$ and $q \in \perp\!\!\!\perp$ then $p \in \perp\!\!\!\perp$.*

Given a pole $\perp\!\!\!\perp$, we say that two processes p, q are *orthogonal*, which we write $p \perp q$, whenever $\bar{\nu}(p \,|\, q) \in \perp\!\!\!\perp$. This naturally induces an orthogonal operator $(\cdot)^\perp : \mathbb{P} \to \mathbb{P}$ on sets of processes by defining $A^\perp \triangleq \{p \in \bar{\Pi} : \forall q{\in}A,\ p{\perp}q\}$. This operator satisfies the usual property of orthogonality.

▶ **Proposition 31.** *For all $A, B \in \mathbb{P}$ and for any non-empty $\mathfrak{B} \subseteq \mathbb{P}$ we have:*

1. *If $A \subseteq B$ then $B^\perp \subseteq A^\perp$*
2. *$A \subseteq A^{\perp\perp}$ and $A^\perp = A^{\perp\perp\perp}$*
3. *$\left(\bigcup_{X\in\mathfrak{B}} X\right)^\perp = \bigcap_{X\in\mathfrak{B}} X^\perp$ and if $\mathfrak{B} \neq \emptyset$ then $\left(\bigcap_{X\in\mathfrak{B}} X\right)^\perp \supseteq \left(\bigcup_{X\in\mathfrak{B}} X^\perp\right)$*
4. *$\left(\bigcup_{X\in\mathfrak{B}} X^{\perp\perp}\right)^\perp = \left(\bigcup_{X\in\mathfrak{B}} X\right)^\perp$*

Proof. Part 1, 2, 3 directly follow from the definition, in particular $(\bar{\Pi}, \bar{\Pi}, \perp)$ where $\perp = \{(p, q) \mid p \perp q\}$ is called a *polarity* in [6] and it defines a Galois connection [3]. For part 4, it suffices to see that $\left(\bigcup_{X\in\mathfrak{B}} X^{\perp\perp}\right)^\perp = \bigcap_{X\in\mathfrak{B}} X^{\perp\perp\perp} = \bigcap_{X\in\mathfrak{B}} X^\perp = \left(\bigcup_{X\in\mathfrak{B}} X\right)^\perp$. ◀

Our realizability interpretation is inspired by the semantics of phases, in particular we interpret formulas in the set of *behaviors* $\mathbb{B} \triangleq \{A \in \mathbb{P} : A^{\perp\perp} = A\}$, *i.e.* the sets that are closed under double orthogonal. This set can be endowed with a structure of complete lattice, as the next proposition shows.

▶ **Proposition 32.** *Let us define* $\bigvee \mathfrak{B} \triangleq (\bigcup \mathfrak{B})^{\perp\perp}$ *for all* $\mathfrak{B} \subseteq \mathbb{B}$. *Then:*

1. $(\mathbb{P}, \bigcap, \bigcup, \subseteq)$ *is a complete lattice with top element* $\overline{\Pi}$ *and bottom element* $\emptyset$.
2. $(\mathbb{B}, \bigcap, \bigvee, \subseteq)$ *is a complete lattice with top element* $\overline{\Pi}$ *and bottom element* $\emptyset^{\perp\perp}$.

In order to interpret MLL formulas in this framework, the main idea consists in seeing the tensor $\otimes$ as defining the parallel composition of two processes in such a way that they do not communicate. This is achieved by making extensive use of substitutions to assign each process a disjoint space of names. Technically, we rely on the fact that $\mathbb{N}$ is in bijection with the set of even (or odd) natural numbers, somehow reflecting the tree structure of the formulas into the space of names.

▶ **Definition 33.** *We fix injections* $\iota_1, \iota_2 : \mathbb{N} \to \mathbb{N}$ *such that*[1] $\iota_1(\mathbb{N}) \cap \iota_2(\mathbb{N}) = \emptyset$ *and* $\iota_1(\mathbb{N}) \cup \iota_2(\mathbb{N}) = \mathbb{N}$. *We use the following notations:* $\mathbb{N}^i \triangleq \iota_i(\mathbb{N})$, $n.i := \iota_i(n)$, $p^i \triangleq p^{\iota_i}$ *and* $e^i \triangleq e^{\iota_i}$. *Given* $p \in \bar{\Pi}$ *s.t.* $\mathrm{FN}(p) \subseteq \mathbb{N}^i$ *we denote as* p^{-i} *the process* $p^{\iota_i^{-1}}$. *Similarly, if* e *is a fusion s.t.* $|e| \subseteq \mathbb{N}^i$, *we denote as* e^{-i} *the fusion* $e^{\iota_i^{-1}}$.

We can now define the semantic counterpart of MLL connectives on processes, which can then lift to behaviors.

For technical purposes, we also define an operator $\cdot * \cdot$, which in fact coincides with the usual application, *i.e.* the left adjoint of the linear implication $\multimap$ as Proposition 37 shows.

▶ **Definition 34.** *For all* $p = (P, e)$ *and* $q = (Q, f)$ *define the following operations*[2]*:*

- $p \bullet q \triangleq p^1 \mid q^2 = (P^1|Q^2, e^1 f^2)$
- $p * q \triangleq ((\nu\mathbb{N}^1)(p \mid q^1))^{-2} = ((\nu\mathbb{N}^1)(P|Q^1, ef^1))^{-2}$.

▶ **Definition 35.** *We define the following operations on* $\mathbb{P}$*:*

$$\begin{array}{rcl|rcl}
1 & := & \{(\mathbb{1}, \Delta_{\mathbb{N}})\}^{\perp\perp} & A \otimes B & := & (A \bullet B)^{\perp\perp} \\
A \bullet B & := & \{p \bullet q \mid p \in A,\ q \in B\} & A \parr B & := & (A^\perp \otimes B^\perp)^\perp \\
A \mid B & := & \{p \mid q \mid p \in A,\ q \in B\} & A \multimap B & := & (A \otimes B^\perp)^\perp \\
A * B & := & \{p * q \mid p \in A,\ q \in B\}^{\perp\perp} & A \parallel B & := & (A \mid B)^{\perp\perp} \\
 & & & A \upharpoonright X & := & \{p \in A \mid \mathrm{FN}(p) \subseteq X\}
\end{array}$$

Observe that the definitions of $1, *, \otimes, \parr, \multimap, \parallel$ define behaviors, and besides, they are insensitive to double orthogonal:

▶ **Proposition 36.** *For any* $A, B \in \mathbb{P}$, *it holds that* $(A \bullet B)^\perp = (A^{\perp\perp} \bullet B^{\perp\perp})^\perp$. *As a consequence,* $A \otimes B = A^{\perp\perp} \otimes B^{\perp\perp}$, $A \parr B = A^{\perp\perp} \parr B^{\perp\perp}$ *and* $A \multimap B = A^{\perp\perp} \multimap B^{\perp\perp}$.

Proof. The inclusion $(A^{\perp\perp} \bullet B^{\perp\perp})^\perp \subseteq (A \bullet B)^\perp$ is true by anti monotonicity of the orthogonal operator. For the reverse inclusion, let us consider $r \in (A \bullet B)^\perp$ and let us prove that $r \in (A^{\perp\perp} \bullet B^{\perp\perp})^\perp$. First, observe that for any processes $p, q,$, we have the following equivalences: $\bar{\nu}(p^1 \mid q^2 \mid r) \equiv \bar{\nu}(((\nu\mathbb{N}^2)(r \mid q^2))^{-1} \mid p) \equiv \bar{\nu}(((\nu\mathbb{N}^1)(r \mid p^1))^{-2} \mid q)$. In particular, for $p \in A$ and $q \in B$, we have $\bar{\nu}(((\nu\mathbb{N}^2)(r|q^2))^{-1}|p) \equiv \overline{\nu}(p^1|q^2|r) \in \perp\!\!\!\perp$. Then $((\nu\mathbb{N}^2)(r|q^2))^{-1} \in A^\perp = A^{\perp\perp\perp}$ and thus $\overline{\nu}(p^1 \mid q^2 \mid r) \in \perp\!\!\!\perp$ for all $p \in A^{\perp\perp}, q \in B$. Using the same observation, we get that $((\nu\mathbb{N}^1)(r \mid p^1))^{-2} \in B^{\perp\perp\perp}$ for all $p \in A^{\perp\perp}$. Finally, using this observation again we obtain that $\overline{\nu}(p^1 \mid q^2 \mid r) \in \perp\!\!\!\perp$ for all $p \in A^{\perp\perp}, q \in B^{\perp\perp}$, which is what we wanted to prove. ◀

[1] For instance, we can take $\iota_1(n) \triangleq 2n + 1$ and $\iota_2(n) \triangleq 2n$.
[2] Observe that $\mathrm{FN}((\nu\mathbb{N}^1)((P, e)|(Q, f)^1) \subseteq \mathbb{N}^2$ and thus we can apply ι_2^{-1} to this process.

▶ **Proposition 37.** *Let us consider $A, B, C \in \mathbb{P}$. Then:*

1. $C \subseteq A \multimap B \iff C * A \subseteq B^{\perp\perp}$.
2. $C * A = \bigcap\{B \in \mathbb{B} \mid C \subseteq A \multimap B\} = \min\{B \in \mathbb{B} \mid C \subseteq A \multimap B\}$.

(In particular, if $B \in \mathbb{B}$ then $C \subseteq A \multimap B \iff C * A \subseteq B$).

Proof.

1. Observe that $A \multimap B = (A \bullet B^\perp)^\perp$. Therefore, we have:

$$\begin{array}{rcl} C \subseteq (A \bullet B^\perp)^\perp & \Leftrightarrow & \forall p \in C. \forall q \in A. \forall r \in B^\perp . \, p \perp q \bullet r \\ & \Leftrightarrow & \forall p \in C. \forall q \in A. \forall r \in B^\perp . \, \bar{\nu}(p|q^1|r^2) \in \perp\!\!\!\perp \\ & \Leftrightarrow & \forall p \in C. \forall q \in A. \forall r \in B^\perp . \, (\nu\mathbb{N}^2)\Big((\nu\mathbb{N}^1)(p|q^1)|r^2\Big) \in \perp\!\!\!\perp \\ & \Leftrightarrow & \forall p \in C. \forall q \in A. \forall r \in B^\perp . \, \bar{\nu}\Big((\nu\mathbb{N}^1)(p|q^1)^{-2}|r\Big) \in \perp\!\!\!\perp \\ & \Leftrightarrow & \forall p \in C. \forall q \in A. \forall r \in B^\perp . \, p * q \perp r \\ & \Leftrightarrow & C * A \subseteq B^{\perp\perp} \end{array}$$

2. Using Part 1, we get that $C * A \subseteq C * A \iff C \subseteq A \multimap (C * A)$ from which we deduce that $C * A \in \{B \mid C \subseteq A \multimap B\}$. On the other hand, if $C \subseteq A \multimap B$, then $C * A \subseteq B$. ◀

To complete the definition of the concurrent realizability interpretation of MLL using processes, it only remains to define the realizability relation.

▶ **Definition 38.** *We say that a PGF p* realizes *a behavior A, which we write $p \Vdash A$, if $p \in A$.*

A detailed study of this interpretation and of the computational content of realizers, which mostly ensure adequate communication between adequate channels using an appropriate fusion, is out of the scope of this paper. We simply illustrate how the identity can be realized.

▶ **Proposition 39.** *For any $A \in \mathbb{P}$, we have that $I \triangleq \bigvee_{n \in \mathbb{N}} (n.1 \leftrightarrow n.2) \Vdash A \multimap A$.*

Proof. We have $A \multimap A = (A \bullet A^\perp)^\perp$ (using Proposition 37). Let us consider $p \in A$ and $q \in A^\perp$, it suffices to prove that $I \perp\!\!\!\perp p^1 \mid q^2$. Using Proposition 29, we get that $\bar{\nu}(p^1 \mid q^2 \mid I) \equiv_\alpha (\nu_2)(\nu_1)(p^1 \mid q^2 \mid I) \equiv_\alpha (\overline{\nu})(p \mid q)$ and we conclude by observing that $(\overline{\nu})(p \mid q) \in \perp\!\!\!\perp$. ◀

4.2 The induced conjunctive involutive monoidal algebra

Most importantly, we can prove that this construction actually defines a conjunctive involutive monoidal algebra. In particular, this entails (via Theorem 12) that it defines a valid interpretation of MLL. The proof is in two parts, we prove that the interpretation induces a CIS, then we show that the set of realized behaviors defines a valid separator.

▶ **Proposition 40.** *The tuple $(\mathbb{B}, \subseteq, \otimes, (\cdot)^\perp)$, where $(\mathbb{B}, \subseteq)$ is the lattice defined in Proposition 32, is a conjunctive involutive structure.*

Proof. We have to check that the axioms of Definition 1 are satisfied. The variance properties and De Morgan law directly follow from the definition of $\mathbb{B}$, $\otimes$ and from Proposition 31. We prove the first distributivity law, the other one being similar. For any $\mathfrak{B} \subseteq \mathbb{B}$, we have:

$$\begin{aligned} A \otimes \bigvee_{B \in \mathfrak{B}} B &\underset{\text{def } \vee}{=} A \otimes \Big(\bigcup_{B \in \mathfrak{B}} B\Big)^{\perp\perp} \underset{\text{Prop } 36}{=} A \otimes \bigcup_{B \in \mathfrak{B}} B \underset{\text{def } \otimes}{=} \Big(A \bullet \bigcup_{B \in \mathfrak{B}} B\Big)^{\perp\perp} \\ &= \Big(\bigcup_{B \in \mathfrak{B}} (A \bullet B)\Big)^{\perp\perp} \underset{\text{Prop } 31.4}{=} \Big(\bigcup_{B \in \mathfrak{B}} (A \bullet B)^{\perp\perp}\Big)^{\perp\perp} \underset{\text{def } \otimes}{=} \Big(\bigcup_{B \in \mathfrak{B}} (A \otimes B)\Big)^{\perp\perp} \underset{\text{def } \vee}{=} \bigvee_{B \in \mathfrak{B}} (A \otimes B) \end{aligned}$$

◀

$$\frac{}{F(a,\mathbb{1}) \equiv_F a} \qquad \frac{}{F(\mathbb{1},a) \equiv_F a} \qquad \frac{}{F(a,b) \equiv_F F(b,a)} \qquad \frac{}{F(a,F(b,c)) \equiv_F F(F(a,b),c)}$$

$$\frac{a \equiv_F a' \quad b \equiv_F b'}{F(a,b) \equiv_F F(a',b')} \qquad \frac{a \equiv_F a' \quad b \equiv_F b'}{a \otimes b \equiv_F a' \otimes b'} \qquad \frac{a \equiv_F a'}{a^{\perp} \equiv_F a'^{\perp}} \qquad \frac{a \equiv_F a' \quad b \equiv_F b' \quad a \preccurlyeq b}{a' \preccurlyeq b'}$$

Figure 2 Axioms for the compositional structure associated to F.

▶ **Proposition 41.** *The following behaviors are realized by pure fusions:*
1. $\bigcap_{A\in\mathbb{B}} A \multimap A$.
2. $\bigcap_{A,B\in\mathbb{B}} (A \otimes B) \multimap (B \otimes A)$.
3. $\bigcap_{A,B,C\in\mathbb{B}} (A{\multimap}B){\multimap}(B{\multimap}C){\multimap}A{\multimap}C$.
4. $\bigcap_{A,B,C\in\mathbb{B}} ((A \otimes B) \otimes C) \multimap (A \otimes (B \otimes C))$.
5. $\bigcap_{A\in\mathbb{B}} A \multimap (1 \otimes A)$.
6. $\bigcap_{A\in\mathbb{B}} (1 \otimes A) \multimap A$.
7. $\bigcap_{A,B\in\mathbb{B}} (A \multimap B) \multimap (B^{\perp} \multimap A^{\perp})$.

Proof. The first statement is Proposition 39, and the other statements follow similar ideas. We give here a realizer for the second statement as an example.

By Definition 35, we have $(A \otimes B) \multimap (B \otimes A) = \big((A \bullet B) \bullet (B \bullet A)^{\perp}\big)^{\perp}$. Consider $p \in (B \bullet A)^{\perp}$, $q \in A$ and $r \in B$, in particular, they satisfy $\overline{\nu}(r^1 \mid q^2 \mid p) \in \perp\!\!\!\perp$. By definition $q^{1.1} \mid r^{2.1} \mid p^2 \in (A \otimes B) \multimap (B \otimes A)$. To get a realizer, we define the substitution $\tau : \mathbb{N}^1 \to \mathbb{N}^2$ s.t. $\tau(n.1.1) := n.2.2$ and $\tau(n.2.1) := n.1.2$. It is now enough to prove that $\varepsilon_\tau \perp\!\!\!\perp q^{1.1} \mid r^{2.1} \mid p^2$, i.e.: $\overline{\nu}(q^{1.1} \mid r^{2.1} \mid p^2 \mid \varepsilon_\tau) \in \perp\!\!\!\perp$. Moreover $\overline{\nu}(q^{1.1} \mid r^{2.1} \mid p^2 \mid \varepsilon_\tau) \equiv (\nu\mathbb{N}^2)(\nu\mathbb{N}^1)(q^{1.1} \mid r^{2.1} \mid p^2 \mid \varepsilon_\tau) \equiv (\nu\mathbb{N}^2)(q^{2.2} \mid r^{1.2} \mid p^2) \equiv \overline{\nu}(r^1 \mid q^2 \mid p) \in \perp\!\!\!\perp$, which ends the proof. ◀

▶ **Proposition 42.** *The set of non-empty behaviors* $\mathcal{S}_{\mathbb{B}} \triangleq \mathbb{B}\backslash\emptyset$ *defines a separator for the conjunctive structure* $(\mathbb{B}, \subseteq, \otimes, (\cdot)^{\perp})$.

Proof. $\mathcal{S}_{\mathbb{B}}$ is upwards closed by construction, and the proof that all the combinators are inhabited by fusions is given in Proposition 41. To prove that it is compatible with the modus ponens, let $a, b \in \mathbb{B}$ be behaviors and p, q be processes such that $p \in a \multimap b$ and $q \in a$. By Proposition 37, we get $(a \multimap b) * a \subseteq b$, and thus in particular $p * q \in b$, *i.e.* $b \in \mathcal{S}_{\mathbb{B}}$. ◀

5 Conjunctive algebras for concurrency

5.1 Parallel composition

So far, we identified the conjunctive part of the realizability model, that is, processes handling disjoint namespaces. We shall now investigate the necessary structure to algebrize parallelism.

▶ **Definition 43.** *Let us consider a CS* $(\mathbb{C}, \otimes, (\cdot)^{\perp}, \preccurlyeq)$*. A* composition *on* $\mathbb{C}$ *is an increasing and* $\curlyvee$*-continuous function* $F : \mathbb{C} \times \mathbb{C} \to \mathbb{C}$*.*

Given a composition F *on* $\mathbb{C}$*, define the induced* compositional structure *associated to* F *as the quotient* $\mathbb{C}/\equiv_F$ *where the equivalence relation* $\equiv_F$ *is the minimum equivalence relation that satisfies the rules of Figure 2.*

Observe that the first four rules express that $(\mathbb{C}, F, \mathbb{1}, \equiv_F)$ is an commutative monoid while the last ones express that the structure of the CS $\mathbb{C}$ is lifted to the quotient $\mathbb{C}/\equiv_F$, thus inducing respectively the operations $\otimes_F$, $(\cdot)^{\perp}_F$ and the preorder $\preccurlyeq_F$. It is a direct verification that the structure $(\mathbb{C}/{\equiv_F}, \otimes_F, (\cdot)^{\perp}_F, \preccurlyeq_F)$ is also a CS. Provided there is no ambiguity, we will avoid the subscripts F on the induced operations. The structure $(\mathbb{C}/{\equiv_F}, \otimes, (\cdot)^{\perp}, F, \preccurlyeq)$ induced by F on $\mathbb{C}$ is denoted as $\mathbb{C}_F$.

▶ **Definition 44.** *Let us consider* $(\mathbb{C}, \otimes, (\cdot)^{\perp}, 1, F, \preccurlyeq)$ *a CS with composition* F*. We define* $\rhd_F : \mathbb{C} \times \mathbb{C} \to \mathbb{C}$ *by means of* $b \rhd_F c \triangleq \curlyvee\{x \in \mathbb{C} \mid xFb \preccurlyeq c\}$*.*

It is straightforward to check that $\rhd_F$ defines a right adjoint to F.

▶ **Proposition 45.** *Let us consider* $(\mathbb{C}, \otimes, (\cdot)^{\perp}, 1, F, \preccurlyeq)$ *a CS with composition* F*. Then* $aFb \preccurlyeq c \iff a \preccurlyeq b \rhd_F c$ *for all* $a, b, c \in \mathbb{C}$*.*

Proof. For the direct implication, $aFb \preccurlyeq c$ implies that $a \in \{x \in \mathbb{C} \mid xFb \preccurlyeq c\}$ hence $a \preccurlyeq b \rhd_F c$. For the reverse implication, since F is monotone, we have

$$aFb \preccurlyeq (b \rhd_F c)Fb \;=\; \Big(\bigcurlyvee\{x \in \mathbb{C} \mid xFa \preccurlyeq c\}\Big) \mid b \;\preccurlyeq\; \bigcurlyvee\{xFb \mid x \in \mathbb{C}, xFb \preccurlyeq c\} \;\preccurlyeq\; c. \quad ◀$$

▶ **Proposition 46.** *Consider the CS* $(\mathbb{B}, \otimes, (\cdot)^{\perp}, 1, \subseteq)$ *of behaviors over* $\bar{\Pi}$*.*

1. *The operation* $\parallel$ *(from Definition 35) is a composition over* $\mathbb{B}$*, the equivalence* $\equiv_{\parallel}$ *is equality and its right adjoint is* $B \rhd_{\parallel} C := (B \parallel C^{\perp})^{\perp}$*.*
2. *The operation* $\otimes$ *(from Definition 35) is a composition over* $\mathbb{B}$ *and its right adjoint is* $B \rhd_{\otimes} C := \big(((B^2 \parallel C^{\perp})^{\perp})_{\restriction \mathbb{N}^1}\big)^{-1}$*.*

Proof.

1. Since $(\overline{\Pi}, |, \mathbb{1})$ is a commutative monoid, then $(\mathbb{B}, \|, \mathbf{1})$ is also a commutative monoid with unit $\mathbf{1} := \{\mathbb{1}\}^{\perp\perp}$. The function $\|$ is increasing by definition. For the continuity, we can check that $(\bigvee \mathfrak{B})\,|A \subseteq \bigvee\{B\|A \mid B \in \mathfrak{B}\}$ for all $\mathfrak{B} \subseteq \mathbb{B}$ and $A \in \mathbb{B}$, and we prove the continuity in the right argument similarly. Besides, $A\|B \subseteq C$ is equivalent to $C^{\perp} \subseteq (A\|B)^{\perp}$, which is equivalent to $\forall p \in A, q \in B, r \in C^{\perp}, \overline{\nu}(p \mid q \mid r) \in \perp\!\!\!\perp$. The last condition is equivalent to $A \subseteq (B\|C^{\perp})^{\perp}$, which proves that $B \rhd_{\|} C := (B\|C^{\perp})^{\perp}$ is the right adjoint of $\|$.
2. We have $(\bigvee \mathfrak{B}) \otimes A = \bigvee\{B \otimes A \mid B \in \mathfrak{B}\}$ and $A \otimes (\bigvee \mathfrak{B}) = \bigvee\{A \otimes B \mid B \in \mathfrak{B}\}$ from Proposition 40, and by construction $\otimes$ is increasing. Moreover, $A \otimes B \subseteq C$ is equivalent to $C^{\perp} \subseteq (A \otimes B)^{\perp}$, which is equivalent to $\forall p \in A, q \in B, r \in C^{\perp}, \overline{\nu}(p^1 \mid q^2 \mid r) \in \perp\!\!\!\perp$. The last condition is equivalent to $A^1 \subseteq ((B^2 \mid C^{\perp})^{\perp})_{\restriction \mathbb{N}^1}$ hence $A \subseteq \big(((B^2 \mid C^{\perp})^{\perp})_{\restriction \mathbb{N}^1}\big)^{-1}$, which proves that $B \rhd_{\otimes} C := \big(((B^2 \mid C^{\perp})^{\perp})_{\restriction \mathbb{N}^1}\big)^{-1}$ is the right adjoint of $\otimes$. ◀

5.2 Embedding the π-calculus through Honda-Yoshida combinators

Following previous works on algebraic structures for realizability models based on sequential calculi [15, 16], we face two options to show that the π-calculus can be faithfully embedded with conjunctive involutive structure with compositions. Either we find a way to soundly embed the different constructs $u(\vec{x}).P, \bar{u}\,\langle\vec{v}\rangle\,, |\,(\nu y)P$, or we can define a set of combinators that is complete for this calculus (as are S and K for the λ-calculus). We opt for the second approach, building on Honda & Yoshida combinators for the π-calculus [10].

▶ **Definition 47.** *A* Honda-Yoshida structure *(HYS) is a CIS* $(\mathbb{C}, \preccurlyeq, \otimes, |, (\cdot)^{\perp}, 1)$ *with composition* $|$ *together with a function* $\mathsf{M} : \mathbb{N} \times \mathbb{N} \to \mathbb{C}$*. Given such a structure, we define the* Honda-Yoshida combinators *of* $\mathbb{C}$ *by means of:*

$$\begin{array}{rl|rl}
\mathsf{K}(a) \triangleq & \curlywedge_{x\in\mathbb{N}} \big(\mathsf{M}(a,x) \rhd 1\big) & \mathsf{Br}(a,b) \triangleq & \curlywedge_{x\in\mathbb{N}} \big(\mathsf{M}(a,x) \rhd \mathsf{F}(b,x)\big) \\
\mathsf{F}(a,b) \triangleq & \curlywedge_{x\in\mathbb{N}} \big(\mathsf{M}(a,x) \rhd \mathsf{M}(b,x)\big) & \mathsf{D}(a,b,c) \triangleq & \curlywedge_{x\in\mathbb{N}} \big(\mathsf{M}(a,x) \rhd \mathsf{M}(b,x)|\mathsf{M}(c,x)\big) \\
\mathsf{Bl}(a,b) \triangleq & \curlywedge_{x\in\mathbb{N}} \big(\mathsf{M}(a,x) \rhd \mathsf{F}(x,b)\big) & \mathsf{S}(a,b,c) \triangleq & \curlywedge_{x\in\mathbb{N}} \big(\mathsf{M}(a,x) \rhd \mathsf{F}(b,c)\big)
\end{array}$$

These definitions are sound with respect to the expected reductions of these combinators, in the sense that each reduction rule $p \to q$ (in [10]) directly yields an inequality $p \preccurlyeq q$.

▶ **Proposition 48.** *Given a HYS* $(\mathbb{C}, \preccurlyeq, \otimes, |, (\cdot)^{\perp}, 1, \mathsf{M})$ *we have:*

$$\begin{array}{rcl|rcl} \mathsf{K}(a)|\mathsf{M}(a,x) & \preccurlyeq & 1 & \mathsf{Bl}(a,b)|\mathsf{M}(a,x) & \preccurlyeq & \mathsf{F}(x,b) \\ \mathsf{F}(a,b)|\mathsf{M}(a,x) & \preccurlyeq & \mathsf{M}(b,x) & \mathsf{Br}(a,b)|\mathsf{M}(a,x) & \preccurlyeq & \mathsf{F}(b,x) \\ \mathsf{D}(a,b,c)|\mathsf{M}(a,x) & \preccurlyeq & \mathsf{M}(b,x)|\mathsf{M}(c,x) & \mathsf{S}(a,b,c)|\mathsf{M}(a,x) & \preccurlyeq & \mathsf{F}(b,c) \end{array}$$

Proof. Straightforward from the definitions. ◀

▶ **Definition 49.** *A* Honda-Yoshida algebra *(HYA) is a HYS* $(\mathbb{C}, \preccurlyeq, \otimes, |, (\cdot)^{\perp}, 1, \mathsf{M})$ *together with a monoidal separator* $\mathcal{S} \subseteq \mathbb{C}$ *s.t. all Honda-Yoshida combinators belong to* $\mathcal{S}$.

So far we have not needed to introduce a notion of reduction because the constructors of Linear Logic in our types reflect the properties of connections and handling of names. However, to obtain a model of behaviors $\mathcal{B}$ where the Honda-Yoshida combinators are inhabited, we ask the poles to be closed by anti-reduction. This choice follows the first author's design when defining the *regular poles* in [2].

▶ **Definition 50.** *Let us consider a pole* $\perp\!\!\!\perp \subseteq \bar{\Pi}$. *We say that* $\perp\!\!\!\perp$ *is* regular *iff for all* $(P,e),(Q,f)$ *(PGF)-processes and names* $u, \vec{x}$, $\{(P \mid Q, ef)\}^{\perp} \subseteq \{(\bar{u}\langle\vec{x}\rangle.P \mid v(\vec{x}).Q, ef)\}^{\perp}$ *whenever* $u \underset{ef}{\sim} v$.

Following [11], the combinators of Honda & Yoshida can be encoded into PGF using the usual π-calculus (pure) processes:

$$\begin{array}{rcl|rcl} \mathsf{m}(a,x) & \triangleq & \bar{a}\langle x\rangle & & & \\ \mathsf{k}(a) & \triangleq & a(x) & \mathsf{br}(a,b) & \triangleq & a(x).\mathsf{f}(b,x) \\ \mathsf{f}(a,b) & \triangleq & a(x).\bar{b}\langle x\rangle & \mathsf{d}(a,b,c) & \triangleq & a(x).(\mathsf{m}(b,x) \mid \mathsf{m}(c,x)) \\ \mathsf{bl}(a,b) & \triangleq & a(x).\mathsf{f}(x,b) & \mathsf{s}(a,b,c) & \triangleq & a(x).\mathsf{f}(b,c) \end{array}$$

▶ **Proposition 51.** *A CIMA of behaviours* $(\mathbb{B}, \preccurlyeq, \otimes, \|, (\cdot)^{\perp}, 1, \mathcal{S}_{\mathbb{B}})$ *induced by a regular pole* $\perp\!\!\!\perp$ *is a HYA with the definition* $\mathsf{M}(a,b) \triangleq \{(\bar{a}\langle b\rangle.\mathbb{1}, \Delta_{\mathbb{N}})\}^{\perp\perp}$.

Proof. It suffices to prove that all the combinators of Definition 47 are inhabited. For that we consider the standard encoding of the negative combinators into the π-calculus (which is embedded into PGF). Then, using the regularity of the pole, we prove that the encoding belongs to the corresponding combinator defined in $\mathbb{C}$. Since $\mathsf{M}(a,b)$ is not empty, we have $\mathsf{M}(a,b) \in \mathcal{S}_{\mathbb{B}}$ by definition of $\mathcal{S}_{\mathbb{B}}$. The other combinators $\mathsf{K}(a), \mathsf{F}(a,b)$, $\mathsf{Bl}(a,b), \mathsf{Br}(a,b), \mathsf{D}(a,b,c)$ and $\mathsf{S}(a,b,c)$ are given by Definition 49 and they are inhabited by their respective encoding on (PGF).

Let us illustrate the technique by proving the case of $\mathsf{K}a$. Observe that since the pole is regular, we have $\{\mathbb{1}\}^{\perp} \subseteq \{\mathsf{k}(a)|\mathsf{m}(a,x)\}^{\perp} = \Big(\{\mathsf{k}(a)\}^{\perp\perp}|\mathsf{M}(a,x)\Big)^{\perp} = \Big(\{\mathsf{k}(a)\}^{\perp\perp} \parallel \mathsf{M}(a,x)\Big)^{\perp}$. Applying the orthogonal map we get: $\{\mathsf{k}(a)\}^{\perp\perp} \parallel \mathsf{M}(a,x) \subseteq \{\mathbb{1}\}^{\perp\perp} \overset{\text{def}}{=} 1$. By Proposition 45 we deduce that $\{\mathsf{k}(a)\}^{\perp\perp} \subseteq \mathsf{M}(a,x) \rhd 1$ for all names a,b,x. Taking the meet over x, we obtain $\mathsf{k}(a) \in \{\mathsf{k}(a)\}^{\perp\perp} \subseteq \bigwedge_{x\in\mathbb{N}} (\mathsf{M}(a,x) \rhd 1) = \mathsf{K}(a)$ and thus $\mathsf{k}(a) \in \mathsf{K}(a)$. ◀

6 Parallel composition cannot be derived

We shall now see that the extra structure presented in the previous section to encompass parallelism within conjunctive involutive monoidal algebra is actually necessary, in the sense that parallelism cannot be derived from the ground structure of a conjunctive involutive

$$\frac{(x : P^{\perp}) \in \Gamma}{\vdash x : P \mid \Gamma}\,(\textsc{Ax}_+) \qquad \frac{\vdash t : A \mid \Gamma \quad \vdash u : B \mid \Gamma}{\vdash (t,u) : A \otimes B \mid \Gamma}\,(\otimes) \qquad \frac{\vdash t : A[P/X] \mid \Gamma}{\vdash t : \exists X.A \mid \Gamma}\,(\exists)$$

$$\frac{(\alpha : N^{\perp}) \in \Gamma}{\vdash \alpha : N \mid \Gamma}\,(\textsc{Ax}_-) \qquad \frac{c : (\vdash \kappa : A, \kappa' : B, \Gamma)}{\vdash \mu(\kappa,\kappa').c : A \parr B \mid \Gamma}\,(\parr) \qquad \frac{\vdash V : A \mid \Gamma \quad X \notin FV(\Gamma)}{\vdash V : \forall X.A \mid \Gamma}\,(\forall)$$

$$\frac{c : (\vdash \kappa : A, \Gamma)}{\vdash \mu\kappa.c : A \mid \Gamma}\,(\mu) \qquad \frac{\vdash t : A \mid \Gamma \quad \vdash u : A^{\perp} \mid \Gamma}{\langle t \parallel u \rangle : (\vdash \Gamma)}\,(\textsc{Cut})$$

Figure 3 System L, typing rules.

monoidal algebra. Indeed, we observed that in the CIMA induced by PGF behaviors, the parallel composition is preexistent since there exists not only an external[3] continuous function, but even a term Φ s.t. $\Phi * t * u = t \| u$ for all t, u. This explains why, in this case, there is no need for a quotient (see Proposition 46.1).

In the general case, when there is no such preexisting term, it is always possible to follow Definition 43 to recover the expected structure through a quotient. We will now highlight that this is, in general, necessary, in the sense that parallel composition can not always be defined with only terms and application in an arbitrary CIMA. We do so by providing a concrete example of a conjunctive involutive monoidal algebra in which we prove that there is no term satisfying the axioms of parallel composition.

We build on a realizability interpretation based on a fragment of Munch-Maccagnoni's system L [17], a polarised sequent calculus tailored to give a term language to Girard's classical logic LC. To draw the comparison with other sequent calculi such as Curien-Herbelin's $\bar{\lambda}\mu\tilde{\mu}$-calculus [4], the main difference lies in the use of polarisation, which makes lazy and strict qualify logical connectives rather than evaluation strategies: instead of having to enforce globally a strategy to avoid critical pairs, lazy and strict evaluations coexist and are dictated locally by the polarities of the corresponding connectives. For a more detailed introduction on the rationale of L, we refer the interested reader to [18].

We work here with a fragment of L, keeping only the connectives necessary to induce a CIMA. As explained above, formulas are divided into positive and negative ones:

$$\begin{array}{ll|l} \textbf{Positive formulas} & P ::= X \mid A \otimes B \mid \exists X.A & \\ \textbf{Negative formulas} & N ::= X^{\perp} \mid A \parr B \mid \forall X.A & \textbf{Formulas} \quad A, B ::= P \mid N \end{array}$$

To each positive formula P is associated a dual negative formula $P^{\perp}$ (and vice-versa), defined by induction on the syntax of formulas. As is usual in linear logic, the map $(\cdot)^{\perp}$ defines an involutive function on formulas:

$$\begin{array}{r|r|r} (X)^{\perp} = X^{\perp} & (A \otimes B)^{\perp} = A^{\perp} \parr B^{\perp} & (\forall X.A)^{\perp} = \exists X.A^{\perp} \\ (X^{\perp})^{\perp} = X & (A \parr B)^{\perp} = A^{\perp} \otimes B^{\perp} & (\exists X.A)^{\perp} = \forall X.A^{\perp} \end{array}$$

Similarly, the syntax of terms and values reflect these distinction between negative and positive connectives. To ease the connection with [17], we stick to the same presentation:

$$\begin{array}{lrcl|lrcl} \textbf{Variables} & \kappa & ::= & x \mid \alpha & \textbf{Commands} & c & ::= & \langle t_+ \parallel t_- \rangle \mid \langle t_- \parallel t_+ \rangle \\ \textbf{Terms} & t_- & ::= & \alpha \mid \mu x.c \mid \mu(\kappa,\kappa).c & \textbf{Values} & V_+ & ::= & x \mid (V, V') \\ & t_+ & ::= & x \mid \mu\alpha.c \mid (t,u) & & V & ::= & V_+ \mid t_- \\ & t & ::= & t_+ \mid t_- & & & & \end{array}$$

[3] That is a meta-theoretic function, as opposed to an *internal* term definable in the language of the algebraic structure at play.

In the sequel, we write $\mathcal{T}_+$ (resp. $\mathcal{T}_-$, $\mathcal{C}$, $\mathbb{V}$) for the set of positive terms (resp. negative terms, commands, values), and denote by $\mathcal{T}_+^0$, etc... the corresponding set of closed terms. As in [17], we use monolateral sequents, following Girard's tradition to have all the formulas on the right. To that end, we need to quotient the syntax with a new α-equivalence $\langle t \,\|\, u\rangle \equiv \langle u \,\|\, t\rangle$, which will simplify overall the presentation. The type system, which distinguishes between sequents of the shape $\vdash t : A \mid \Gamma$ to type terms (where A is the formula *in the stoup*) and sequents of the shape $c : (\vdash \Gamma)$ to type commands, is given in Figure 3. The reader may observe that the rule for the universal quantifier has to be restricted to values in order to avoid the usual inconsistency of polymorphism in presence of side-effects [17].

The operational semantics is defined as a weak-head reduction relation on commands, by means of the following reduction rules:

$$\begin{array}{lcl|rcl}
\langle \mu\alpha.c \,\|\, V\rangle & \to_\mu & c[V/\alpha] & \langle (V,V') \,\|\, \mu(\kappa,\kappa').c\rangle & \to_\beta & c[V/\kappa, V'/\kappa'] \\
\langle V \,\|\, \mu x.c\rangle & \to_\mu & c[V/x] & \langle (t,t') \,\|\, u\rangle & \to_\xi & \left\langle t \,\middle\|\, \mu\kappa.\langle t' \,\|\, \mu\kappa'.\langle (\kappa,\kappa') \,\|\, u\rangle\rangle \right\rangle
\end{array}$$

The different binders only reduce in front of values, while the $\to_\xi$ rules specify how constructors should be expanded when they are built on top of a term instead of a value. In particular, it is worth noting that the reduction $\to \triangleq \to_\mu \cup \to_\beta \cup \to_\xi$ has no critical pair, which entails the Church-Rosser property when considering its extension to subcommands.

Following [17], we can now define a realizability interpretation *à la* Krivine relying on this calculus. As is usual in Krivine realizability, we say that a *pole* is any subset of $\mathcal{C}^0$ that is closed by anti-reduction. This induces an orthogonality relation on terms, which we extend to any sets $T \in \mathcal{P}(\mathcal{T}_+^0)$ or $U \in \mathcal{P}(T_-^0)$ by defining:

$$T^\perp \triangleq \{t_- \in \mathcal{T}_-^0 : \forall t_+ \in T, \langle t_+ \,\|\, t_-\rangle \in \perp\!\!\!\perp\} \quad\Big|\quad U^\perp \triangleq \{t_+ \in \mathcal{T}_+^0 : \forall t_- \in U, \langle t_+ \,\|\, t_-\rangle \in \perp\!\!\!\perp\}$$

In this context, we call *behavior* any subset T of $\mathcal{T}_+^0$ or $\mathcal{T}_-^0$ such that $T = T^{\perp\perp}$ and we denote by $\mathbf{H}$ the set of all behaviors. Observe that $(\cdot)^\perp$ defines an involutive map on $\mathbf{H}$.

We are now ready to define a realizability interpretation that will associate to any formula a behavior. More precisely, we extend the language of formulas to consider formulas with (positive) parameters R in the set $\Pi \triangleq \mathcal{P}(\mathcal{T}_+^0 \cap \mathbb{V})$, and to any such positive formula P we associate a behavior $|P| \in \mathcal{P}(\mathcal{T}_+^0)$ while to any negative formula N we associate a behavior $|N| \in \mathcal{P}(\mathcal{T}_-^0)$. The interpretation $|\cdot|$ is defined by induction on formulas:

$$\begin{array}{rcl|rcl}
|R| & \triangleq & R^{\perp\perp} & |R^\perp| & \triangleq & R^\perp \\
|A \otimes B| & \triangleq & (|A| \otimes |B|)^{\perp\perp} & |A \parr B| & \triangleq & (|A^\perp| \otimes |B^\perp|)^\perp \\
|\exists X.A| & \triangleq & \left(\bigcup_{R\in\mathbb{V}} |A[R/X]|_\mathbb{V}\right)^{\perp\perp} & |\forall X.A| & \triangleq & \left(\bigcap_{R\in\mathbb{V}} |A[R/X]|_\mathbb{V}\right)^{\perp\perp}
\end{array}$$

where $|A|_\mathbb{V} \triangleq |A| \cap \mathbb{V}$ and $T \otimes U \triangleq \{(t,u) : t \in T \wedge u \in U\}$. Observe that by construction, for any closed formula A, it holds that $|A^\perp| = |A|^\perp$.

▶ **Definition 52.** *For any term t, one says that t realizes a formula A whenever $t \in |A|$, which we denote by $t \Vdash A$. Similarly, we write $\sigma \Vdash \Gamma$ to denote that a substitution σ realizes a context $\Gamma \equiv x_1 : A_1, ..., x_n : A_n$ when for any $1 \le i \le n$, $\sigma(x_i) \Vdash A_i$.*

We say that t is a universal realizer *of A and we write $t \Vvdash A$ when a term t realizes A for any possible choice of pole.*

We will not develop any further the properties of this construction, but we shall at least emphasize that the typing rules defined in Figure 3 are adequate with respect to this interpretation, which is proven as usual by induction over typing derivations [17].

▶ **Proposition 53** (Adequacy). *Let Γ be a typing context, $\perp\!\!\!\perp$ a pole and σ be a substitution such that $\sigma \Vdash \Gamma$, then for any term t and formula A, if $\vdash t : A \mid \Gamma$, then $t^\sigma \Vdash A$.*

Even though the fragment of L we have selected does not account for linearity, the last proposition shows that the realizability interpretation still defines a model for MLL. In particular, as was observed in [16], algebraically it induces a conjunctive structure. This observation can be adapted to our framework, to show that the realizability interpretation defined above actually induces a conjunctive involutive monoidal algebra.

▶ **Proposition 54.** *The tuple $(\boldsymbol{H}, \subseteq, \otimes, (\cdot)^\perp, \mathbb{1}, \mathcal{S})$, where $\mathbb{1} \triangleq \mathcal{T}_+^0$ and $\mathcal{S} = \boldsymbol{H}\backslash\emptyset$, is a CIMA.*

Proof. The proof is analogous to the proof in [16], up to the observation that $(\cdot)^\perp$ is involutive on $\mathbf{H}$ (by definition of $\mathbf{H}$). We first show that $(\mathbf{H}, \subseteq, \otimes, (\cdot)^\perp)$ is a conjunctive involutive structure. Indeed, the set of behaviors, ordered by inclusion, defines a complete lattice whose join is given by the union, and the map $\cdot \otimes \cdot$ is clearly covariant in its two arguments. while it is clear that $(\cdot)^\perp$ is anti-monotonic. Regarding De Morgan law, we have for any set $\mathcal{U} \subseteq \mathcal{T}^0$:

$$\Big(\bigcup_{T\in\mathcal{U}} T\Big)^\perp = \Big(\bigcup_{T_+\in\mathcal{U}\cap\mathcal{T}_+} T_+ \;\cup \bigcup_{T_-\in\mathcal{U}\cap\mathcal{T}_-} T_-\Big)^\perp = \Big(\bigcup_{T_+\in\mathcal{U}\cap\mathcal{T}_+} T_+\Big)^\perp \cap \Big(\bigcup_{T_-\in\mathcal{U}\cap\mathcal{T}_-} T_-\Big)^\perp$$
$$= \bigcap_{T_+\in\mathcal{U}\cap\mathcal{T}_+} T_+^\perp \;\cap \bigcap_{T_-\in\mathcal{U}\cap\mathcal{T}_-} T_-^\perp \;= \bigcap_{T\in\mathcal{U}} T^\perp$$

Then we prove that, when considering the unit $\mathbb{1} \triangleq \mathcal{T}_+^0$ (which is indeed a behavior), the set of non-empty behaviors $\mathcal{S} = \mathbf{H}\backslash\emptyset$ defines a valid separator. This mostly amounts to finding a realizer for each of the different axioms, since $\mathcal{S}$ is clearly upwards-closed. and is compatible with modus ponens. Indeed, if $A, B \in \mathbf{H}$ are such that $A \multimap B \in \mathcal{S}$ and $A \in \mathcal{S}$, by definition of $\mathcal{S}$ it means that there exists two terms t and u such that $t \in A \multimap B$ and $u \in A$. Then $t\,u = \mu\kappa.\langle t \parallel (u, \alpha)\rangle \Vdash B$, using the typing rule for application above and Proposition 53. To find realizers for the axiom, it suffices to find a term of the expected type, and to use adequacy (cf. Proposition 53). For instance, we can pick:

- $\mu(x,\alpha).\langle x \parallel \mu(\kappa_1,\kappa_2).\langle(\kappa_2,\kappa_1) \parallel \alpha\rangle\rangle \Vdash (A \otimes B) \multimap (B \otimes A)$
- $\lambda z.\mu(x,\beta).\langle x \parallel \mu(\kappa,\kappa').\langle(z\,\kappa,\kappa') \parallel \beta\rangle\rangle \Vdash (A \multimap B) \multimap (A \otimes C) \multimap (B \otimes C)$
- $\mu(x,\alpha).\Big\langle x \,\Big\Vert\, \mu(y,\kappa_3).\langle y \parallel \mu(\kappa_1,\kappa_2).\langle(\kappa_1,(\kappa_2,\kappa_3)) \parallel \alpha\rangle\rangle\Big\rangle \Vdash ((A\otimes B)\otimes C) \multimap (A\otimes(B\otimes C))$
- For any $t_+ \in \mathcal{T}_+^0$, $\lambda\kappa.(t_+,\kappa) \Vdash A \multimap \mathbb{1} \otimes A$
- $\mu(x,\kappa).\langle x \parallel \mu(\_,\kappa').\langle\kappa' \parallel \kappa\rangle\rangle \Vdash \mathbb{1} \otimes A \multimap A$
- $\mu(x,\alpha).\langle x \parallel \mu(\kappa,\kappa').\langle(\kappa,\kappa') \parallel \alpha\rangle\rangle \Vdash (A \multimap B) \multimap (B^\perp \multimap A^\perp)$ ◀

We have shown so far that the realizability interpretation based on L defines a conjunctive involutive monoidal algebra as expected. Besides, we can show that this particular kind of CIMA does not admit parallelism, thus justifying the necessity of the extra requirements formulated in Definition 43. Indeed, to obtain a compositional structure in such a model without adding some extra structure would mean the desired composition F could be defined by means of a term t (*i.e.* $F(u,v) \triangleq t\,u\,b$) such that the expected axioms are satisfied.

We prove that this is not possible, in that the first two axioms ($F(a,\mathbb{1}) \equiv a$ and $F(\mathbb{1},a) \equiv a$) cannot be satisfied together. The proof, which is folklore in Krivine realizability, mostly consists in proving that a term that realizes both of these axioms should essentially behave like a parallel ⫛ operator, which does not exist in L. This should not come as a surprise, since, to put it differently, it essentially means that to get the algebraic structure corresponding to parallelism, we actually need to have parallel computations in our calculus.

▶ **Proposition 55.** *There is no $t \in \mathcal{T}_+^0$ s.t. $t \Vvdash \forall X.(X \multimap \mathbb{1} \multimap X)$ and $t \Vvdash \forall X.(\mathbb{1} \multimap X \multimap X)$.*

Proof. To simplify the proof, let us assume that the calculus is extended with two inert (positive) constants $\boldsymbol{\kappa_1}, \boldsymbol{\kappa_2}$ and a negative one $\boldsymbol{\alpha}$ (alternatively, one could consider any terms $u_1, u_2 \in \mathcal{T}_+^0$ and $v \in \mathcal{T}_-^0$ such that the commands $\langle u_1 \parallel v\rangle$ and $\langle u_2 \parallel v\rangle$ are blocked).

We reason by contradiction by assuming that such a term t exists. Let $u \in \mathcal{T}_-^0$ be any term, and $\mathbb{X} \triangleq \{u\} \in \Pi$. Consider the pole $\perp\!\!\!\perp_1 = \{c : c \to \langle \kappa_1 \parallel u\rangle\}$. By construction, we have that $\boldsymbol{\kappa_1} \in |\mathbb{X}|$, $\boldsymbol{\kappa_2} \in |\mathbb{1}|$ and $\boldsymbol{\alpha} \in |\mathbb{X}|^{\perp}$. Since by assumption $t \Vdash \mathbb{X} \multimap \mathbb{1} \multimap \mathbb{X}$, we get $\langle t \parallel (\boldsymbol{\kappa_1}, (\boldsymbol{\kappa_2}, \boldsymbol{\alpha}))\rangle \in \perp\!\!\!\perp_1$, *i.e.* $\langle t \parallel (\boldsymbol{\kappa_1}, (\boldsymbol{\kappa_2}, \boldsymbol{\alpha}))\rangle \to \langle \boldsymbol{\kappa_1} \parallel \boldsymbol{\alpha}\rangle$.

By taking $\perp\!\!\!\perp_2 = \{c : c \to \langle \boldsymbol{\kappa_2} \parallel \boldsymbol{\alpha}\rangle\}$, we prove similarly that $\langle t \parallel (\boldsymbol{\kappa_1}, (\boldsymbol{\kappa_2}, \boldsymbol{\alpha}))\rangle \to \langle \boldsymbol{\kappa_2} \parallel \boldsymbol{\alpha}\rangle$. Since both commands $\langle \boldsymbol{\kappa_1} \parallel \boldsymbol{\alpha}\rangle$ and $\langle \boldsymbol{\kappa_2} \parallel \boldsymbol{\alpha}\rangle$ are blocked and the calculus is deterministic, we indeed found a contradiction. ◀

7 Conclusion and future work

7.1 Exponentials

In addition to studying in more depth the properties of the various structures introduced in this paper, a natural prolongation of this work would be to investigate the possibility to build on conjunctive involutive monoidal algebra to account for more expressive logical systems, in particular MALL with exponentials. This would provide us with a complete algebraic presentation of the first author's work [2]. To that end, we could rely on Honda & Yoshida combinatorial approach to replication [11]. Another path to follow in that direction would consist in exploring the connections with the Geometry of Interaction as it is presented in Duchesne's work [5], where MLL is interpreted using partial permutations and the tensor corresponds to an appropriate disjoint union. This bears similarities with our interpretation using injections $\iota_i : \mathbb{N} \to \mathbb{N}$ for the tensor to ensure disjoint namespaces.

7.2 Concurrency

A central element in our study of parallelism and its axiomatization is the role of non-determinism. Indeed, the presence in a process of several actions that may synchronize together in different ways is both a source of expressiveness, as witnessed by the famous "parallel or" which relies on this feature, but also an obstacle for logical study. In some sense, our search for an axiomatization of parallel composition aims at identifying the part of the logical structure that is responsible for such phenomena. An analogy can be made with the computational interpretation of classical logic: it is well known that the classical sequent calculus LK is not confluent, to the point that any direct denotational semantics of proofs is degenerated, but interesting semantics can be obtained by means of translations into better behaved logics. Such translations are effective because they impose constraints on proof reduction, like call-by-name or call-by-value policies in languages with control. Our system with parallelism could be envisaged as a linear analogue of full classical logic with no a priori constraints on strategies.

$$\begin{tikzcd}
\text{process calculus} \arrow[r, dotted, leftrightarrow] \arrow[d, "\text{determinization}"'] & \text{axiomatization of concurrency?} \arrow[d, dotted] & \text{LK} \arrow[l, dotted] \arrow[dl, "\text{various}"'] \arrow[d, "\neg\neg \text{ and CPS translations}"] \\
\text{deterministic processes} \arrow[r, leftrightarrow, "\text{typing}"'] & \text{linear logic} & \text{LJ} \arrow[l, "\text{linearization}"]
\end{tikzcd}$$

In this line of thought, the conjunctive structures we are after will provide the fundamental structure for logics of concurrent behaviours, which we can call an axiomatization of concurrency. Particular logics or type systems would be built on top of this structure, through the choice of appropriate operators, and possibly extra axioms, so that soundness of a particular type system for a particular calculus reduces to the fact that the considered calculus does provide an appropriate conjunctive structure.

References

1 Samson Abramsky, Esfandiar Haghverdi, and Philip Scott. Geometry of interaction and linear combinatory algebras. *Mathematical Structures in Computer Science*, 12(5):625–665, 2002. doi:10.1017/S0960129502003730.

2 Emmanuel Beffara. *Logique, réalisabilité et concurrence*. PhD Thesis, Université Paris 7, December 2005.

3 G. Birkhoff. *Lattice theory*, volume 25 of *Colloquium publications*. American Mathematical Society, 1940.

4 Pierre-Louis Curien and Hugo Herbelin. The duality of computation. In *Proceedings of ICFP 2000*, SIGPLAN Notices 35(9), pages 233–243. ACM, 2000. doi:10.1145/351240.351262.

5 Étienne Duchesne. *La localisation en logique : géométrie de l'interaction et sémantique dénotationnelle*. Thèse de doctorat, Université Aix-Marseille 2, 2009. URL: http://www.theses.fr/2009AIX22080.

6 Walter Ferrer Santos, Jonas Frey, Mauricio Guillermo, Octavio Malherbe, and Alexandre Miquel. Ordered combinatory algebras and realizability. *Mathematical Structures in Computer Science*, 27(3):428–458, 2017. doi:10.1017/S0960129515000432.

7 Yuxi Fu. The χ-Calculus. In *1997 Advances in Parallel and Distributed Computing Conference (APDC)*, pages 74–81. Computer Society Press, 1997.

8 Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–101, 1987. doi:10.1016/0304-3975(87)90045-4.

9 Jean-Yves Girard. Locus Solum: From the rules of logic to the logic of rules. *Mathematical Structures in Computer Science*, 11(3):301–506, June 2001. doi:10.1017/S096012950100336X.

10 Kohei Honda and Nobuko Yoshida. Combinatory representation of mobile processes. In *ACM-SIGACT Symposium on Principles of Programming Languages*, 1994.

11 Kohei Honda and Nobuko Yoshida. Replication in concurrent combinators. In Masami Hagiya and John C. Mitchell, editors, *Theoretical Aspects of Computer Software*, pages 786–805, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

12 J.M.E. Hyland. The effective topos. *Studies in Logic and the Foundations of Mathematics*, 110:165–216, 1982. The L. E. J. Brouwer Centenary Symposium. doi:10.1016/S0049-237X(09)70129-6.

13 J.-L. Krivine. Realizability in classical logic. In Interactive models of computation and program behaviour. *Panoramas et synthèses*, 27:197–229, 2009. Publisher: Société Mathématique de France.

14 Jean-Louis Krivine. Realizability algebras II : new models of ZF + DC. *Logical Methods in Computer Science*, 8(1):10, February 2012. 28 p. doi:10.2168/LMCS-8(1:10)2012.

15 Alexandre Miquel. Implicative algebras: a new foundation for realizability and forcing. *Mathematical Structures in Computer Science*, 30(5):458–510, 2020. doi:10.1017/S0960129520000079.

16 Étienne Miquey. Revisiting the Duality of Computation: An Algebraic Analysis of Classical Realizability Models. In Maribel Fernández and Anca Muscholl, editors, *28th EACSL Annual Conference on Computer Science Logic (CSL 2020)*, volume 152 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 30:1–30:18, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.CSL.2020.30.

17 Guillaume Munch-Maccagnoni. Focalisation and Classical Realisability. In Erich Grädel and Reinhard Kahle, editors, *Computer Science Logic '09*, volume 5771 of *Lecture Notes in Computer Science*, pages 409–423. Springer, Heidelberg, 2009. doi:10.1007/978-3-642-04027-6_30.

18 Guillaume Munch-Maccagnoni. *Syntax and Models of a non-Associative Composition of Programs and Proofs.* PhD thesis, Univ. Paris Diderot, 2013.

19 Joachim Parrow and Björn Victor. The Fusion Calculus: Expressiveness and Symmetry in Mobile Processes. In *13th IEEE Symposium on Logic in Computer Science (LICS)*, pages 176–185, 1998.

20 Andrew M. Pitts. Tripos theory in retrospect. *Mathematical Structures in Computer Science*, 12(3):265–279, 2002. doi:10.1017/S096012950200364X.

21 Davide Sangiorgi and David Walker. *The π-Calculus: A Theory of Mobile Processes.* Cambridge University Press, 2001.

22 Thomas Streicher. Krivine's classical realisability from a categorical perspective. *Mathematical Structures in Computer Science*, 23(6):1234–1256, 2013. doi:10.1017/S0960129512000989.

23 Lucian Wischik and Philippa Gardner. Explicit fusions. *Theoretical Computer Science*, 340(3):606–630, August 2005. doi:10.1016/j.tcs.2005.03.017.