Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
Electronic Notes in Theoretical Computer Science 265 (2010) 161–176
www.elsevier.com/locate/entcs

From Focalization of Logic to the Logic of Focalization[1]

Michele Basaldella[a,2] Alexis Saurin[b,3] Kazushige Terui[a,4]

[a] *RIMS, Kyoto University, Japan*

[b] *PPS & INRIA πr^2, Paris, France*

Abstract

Andreoli originally discovered focalization as a concrete proof search strategy in proof theory of linear logic, putting to the foreground the role of polarity in logic.
The aim of the present paper is to give a more abstract account on focalization in the framework of ludics. We describe focalization as a map (embodied by an untyped proof/design) from an unsynthesized to a synthesized type/behaviour. The map turns out to be a retraction of another map, that is related to invertibility of negative connectives. In this way we formalize the common intuition that focalization of positive connectives is dual to invertibility of negative ones.

Keywords: Linear logic, ludics, focalization, proof-search, interaction.

1 Introduction

Focalization is a deep outcome of linear logic (**LL**) proof theory, putting to the foreground the role of polarity in logic. It resulted in important advances in various fields ranging from proof search (the original motivation for Andreoli's study [2] of focalization) to game semantical analysis of logic.

1.1 Focalization in linear logic

In linear logic, one distinguishes two classes of logical connectives: *positive* $(\otimes, \oplus, \mathbf{0}, \mathbf{1}, \exists, !)$ and *negative* $(\parr, \&, \top, \bot, \forall, ?)$ connectives. The distinction can be easily understood in terms of proof search in (one-sided) sequent calculus. The introduction rules for negative connectives $\parr, \&, \top, \bot, \forall$ are *invertible*: in the bottom-up

[1] A preliminary version of this work has been presented in [3].

[2] Email: mbasalde@kurims.kyoto-u.ac.jp

[3] Email: alexis.saurin@pps.jussieu.fr

[4] Email: terui@kurims.kyoto-u.ac.jp

doi:10.1016/j.entcs.2010.08.010

reading, the rule is deterministic, i.e., there is no choice to make and provability of the conclusion implies provability of the premisses. On the other hand, the introduction rules for positive connectives involve choices: e.g., splitting the context in $\otimes$ rule, or choosing between $\oplus_L$ and $\oplus_R$ rules, resulting in possibly erroneous choices during proof search. Still, positive connectives satisfy a strong property called *focalization* [2]: let us consider a sequent $\vdash F_1, \ldots, F_n$ containing no negative formulas, then there is (at least) one formula F_i which can be used as a *focus* for the search by hereditarily selecting F_i and its positive subformulas as principal formulas up to the first negative subformulas. This property induces the following strategy of proof search called *focalization discipline*:

The sequent $\vdash \Gamma$ contains a negative formula : choose any negative formula (e.g. the leftmost one) and decompose it using the only possible negative rule.

The sequent $\vdash \Gamma$ contains no negative formula : choose some positive formula and decompose it (and its subformulas) hereditarily until we get to atoms or negative subformulas.

It is proven in [2] that the focalization discipline is a complete proof search strategy. Other approaches to focalization consider proof transformation techniques [4,12,15].

Focalization is not only concerned with efficiency of proof search, but also a key to understand the CPS translation [6,16,18] and the space compression in computational complexity [17]. It also underlies game semantics [11,14]. In view of its importance, it would be worthwhile to study focalization from as many perspectives as possible. While focalization is originally explained by permutability of inference rules, we look for a more abstract, “algebraic” account in the framework of ludics.

Some other recent works provide an enlightening view of the structure of focalization and its connections with interaction-based semantics. Though focalization induces a sequential view of logic, it is worth mentioning Abramsky’s discussion [1] on sequential and concurrent approaches to games and logic: pursuing the concurrent game direction, Melliès developed a model of linear logic based on asynchronous games [13]. Melliès and Tabareau recently suggested to consider tensor logic (*i.e.*, a logic equipped with a tensorial, noninvolutive negation) as being more primitive than linear logic. This leads to consider *dialogue categories* [14]. Finally, one can also mention Munch-Maccagnoni’s recent work on providing a polarized approach [16] to Curien and Herbelin’s *duality of computation* and shed interesting light on focalization. This work uses Krivine’s classical realizability.

1.2 Focalization in ludics

Ludics [9] is a pre-logical framework proposed by Girard which aims to analyze various logical and computational phenomena at a foundational level.

One reason why we adopt ludics is that types (called *behaviours*) are built as sets of untyped proofs (called *designs*), as in some realizability interpretations and semantic/operational types. This view of types as proof sets allows us to analyze various properties of proofs, including focalization, at the level of types. Furthermore, ludics has the following prominent features that are relevant for focalization.

1. Synthetic connectives.

According to the focalization discipline, one decomposes a positive formula persistently until reaching atomic or negative subformulas. This strategy is internally expressed in ludics by considering *synthetic connectives* that combine several connectives of the same polarity into one, and demanding that every behaviour (type) be a strict alternation of positive and negative synthetic connectives. For instance, a compound formula $\mathbf{N} \otimes (\mathbf{M} \oplus \mathbf{K})$ of **LL** with two positive connectives $\oplus$, $\otimes$ is expressed by a behaviour $\otimes\oplus\,(\mathbf{N}, \mathbf{M}, \mathbf{K})$ of ludics with one synthetic connective $\otimes\oplus$. With $\otimes$ and $\oplus$ inseparable, the latter only admits focalized proof search. On the other hand, unfocalized proof search can be simulated by considering $\mathbf{N} \otimes \uparrow(\mathbf{M} \oplus \mathbf{K})$, where a dummy negative connective $\uparrow$ is artificially inserted between two positive connectives. This allows us to think of focalization as a map f from $\mathbf{P} = \mathbf{N} \otimes \uparrow(\mathbf{M} \oplus \mathbf{K})$ to $\mathbf{P}^f = \otimes\oplus\,(\mathbf{N}, \mathbf{M}, \mathbf{K})$.

2. Behaviours and biorthogonality.

Notice that focalization is intrinsically a context-sensitive phenomenon, as it states that one can obtain a proof of $\vdash \Gamma, \mathbf{P}^f$ from that of $\vdash \Gamma, \mathbf{P}$ for *any* context Γ. In ludics, interaction with contexts is taken into account by demanding that behaviours be closed under the biorthogonal operation: $\mathbf{P} = \mathbf{P}^{\perp\perp}$. Indeed, this biorthogonal-closedness (together with associativity of normalization) implies the *closure principle*: informally speaking, a design D belongs to a "sequent of behaviour" $\vdash \Gamma, \mathbf{P}$ if and only if $D[E/x]$ (D applied to E) belongs to $\mathbf{P}$ for all designs E in $\Gamma^{\perp}$. The effect is sort of modularity: even though focalization involves contexts, one can "project" them on a single behaviour $\mathbf{P}$.

3. Internal completeness.

A difficulty of biorthogonal closure is that it may obscure the content of a behaviour $\mathbf{P}$, and make it hard to verify the correctness of the focalization map f directly. Fortunately, ludics enjoys *internal completeness*, that allows us to remove biorthogonal closure and to give a concrete description to the designs in $\mathbf{P}$. This is a key to ensure that the focalization map $f : \mathbf{P} \longrightarrow \mathbf{P}^f$ is a total function.

Putting these three features together, we can informally explain focalization as follows: given an unfocalized proof D of $\vdash \Gamma, \mathbf{P}$ (seen as design), we derive:

$$\begin{aligned} D \in {\vdash} \Gamma, \mathbf{P} &\Longrightarrow D[E/x] \in \mathbf{P} && \text{for every } E \in \Gamma^{\perp} \\ &\Longrightarrow f(D)[E/x] = f(D[E/x]) \in \mathbf{P}^f && \text{for every } E \in \Gamma^{\perp} \\ &\Longrightarrow f(D) \in {\vdash} \Gamma, \mathbf{P}^f \end{aligned}$$

to obtain a focalized proof $f(D)$ of $\vdash \Gamma, \mathbf{P}^f$.

In this paper, we shall describe the focalization map f as a retraction of a map u which may explain the invertibility of negative inference rules. This way we promote an "algebraic" view of focalization in the setting of ludics: *focalization as a retraction of invertibility*.

1.3 Organization of the paper

In Section 2, we first recall the basic objects of ludics, namely *designs*, and introduce *functionals*, which work as morphisms between behaviours. In Section 3, we define synthetic signatures and synthetic connectives, which are the starting point for studying *focalizing* and *inverting* functionals, that we first introduce in the untyped setting in Section 4. In Section 5, we move to a (semantically) typed setting and reconsider functionals as functions between behaviours (types). Logical behaviours are described in Section 6 together with internal completeness. Finally, we complete our study in Section 7 providing a ludics account of focalization.

2 Untyped designs

2.1 Syntax

We recall the (identity-free) syntax of designs following the notation of [17], inspired by the close relations between ludics and linear π-calculus [7].

Designs are built over a given **signature** $\mathcal{A} = (A, \mathsf{ar})$, where A is a set of **names** $a, b, c, \ldots$ and $\mathsf{ar} : A \longrightarrow \mathbb{N}$ a function which assigns an *arity* $\mathsf{ar}(a)$ to each name a. Let $\mathcal{V}$ be a countable set of variables $\mathcal{V} = \{x, y, z, \ldots\}$. Over a fixed signature $\mathcal{A}$, a *positive action* is $\overline{a}$ with $a \in A$, and a *negative action* is $a(x_1, \ldots, x_n)$ where $x_1, \ldots, x_n$ are distinct variables and $\mathsf{ar}(a) = n$. In the sequel, an expression of the form $a(\boldsymbol{x})$ always stands for a negative action.

Definition 2.1 The positive P (resp. negative N) **designs** are *coinductively* generated by the following grammar:

$$
\begin{array}{rll}
P ::= & \Omega & \text{(partiality, divergence)},\\
| & \maltese & \text{(daimon, termination)},\\
| & x|a\langle N_1, \ldots, N_n\rangle & \text{(head normal form)},\\
| & N|a\langle N_1, \ldots, N_n\rangle & \text{(cut)},\\
N ::= & \sum a(\boldsymbol{x}).P_a & \text{(abstraction)},
\end{array}
$$

where $\mathsf{ar}(a) = n$, $\boldsymbol{x} = x_1, \ldots, x_n$ and the formal sum $\sum a(\boldsymbol{x}).P_a$ is built from $|A|$-many components $\{a(\boldsymbol{x}).P_a\}_{a \in A}$.

Designs may be considered as infinitary λ-terms with *named* applications and *named and superimposed* abstractions. We use meta-variables $P, Q, \ldots$ (resp. N, M, $\ldots$, resp. $D, E, \ldots$) to denote positive (resp. negative, resp. arbitrary) designs. Any subterm E of D is called a **subdesign** of D.

We use Ω to encode *partial* sums: given a set $\alpha = \{a(\boldsymbol{x}), b(\boldsymbol{y}), \dots\}$ of negative actions, we write $a(\boldsymbol{x}).P_a + b(\boldsymbol{y}).P_b + \cdots$ or $\sum_\alpha a(\boldsymbol{x}).P_a$ to denote the design $\sum a(\boldsymbol{x}).R_a$, where $R_a = P_a$ if $a(\boldsymbol{x}) \in \alpha$, and $R_a = \Omega$ otherwise.

A design D may contain free and bound variables. An occurrence of subterm $a(\boldsymbol{x}).P_a$ *binds* the free-variables $\boldsymbol{x}$ in P_a. Variables which are not under the scope of any binder $a(\boldsymbol{x})$ are *free.* $\mathsf{fv}(D)$ denotes the set of free variables occurring in D. Designs are always considered *up to α-equivalence*, that is up to renaming of bound variables (see [17] for further details).

A positive design which is neither Ω nor ✠ is either of the form $(\sum a(\boldsymbol{x}).P_a)| \overline{a}\langle N_1, \dots, N_n\rangle$ and called a **cut** or of the form $x|\overline{a}\langle N_1, \dots, N_n\rangle$ and called a **head normal form**. The *head variable* x in the design above plays the same role as a pointer does in a strategy from Hyland-Ong's games model and an address (or locus) in Girard's ludics.

In the first case a cut reduces to another positive design via the following *reduction rule*, written $\longrightarrow$:

$$\big(\sum a(x_1, \dots, x_n).P_a\big)|\overline{a}\langle N_1, \dots, N_n\rangle \longrightarrow P_a[N_1/x_1, \dots, N_n/x_n];$$

where the expression $D[N_1/x_1, \dots, N_n/x_n]$ denotes the design obtained by the simultaneous and capture-free substitution of N_i for x_i, $1 \le i \le n$, in D.

Example 2.2 Let $N = a(x).✠ + b(x).y|\overline{d}\langle M\rangle + c(x).x|\overline{e}\langle M\rangle$, with $x \notin \mathsf{fv}(M)$. We have:

- $N|\overline{a}\langle L\rangle \longrightarrow$ ✠ (termination);
- $N|\overline{d}\langle L\rangle \longrightarrow \Omega$ (divergence);
- $N|\overline{b}\langle L\rangle \longrightarrow y|\overline{d}\langle M\rangle$ (reduction to head normal form);
- $N|\overline{c}\langle L\rangle \longrightarrow L|\overline{e}\langle M\rangle$ (reduction to another cut).

We write $\longrightarrow^*$ for the transitive reflexive closure of $\longrightarrow$.

Given a design D, we define its **normal form** $[\![D]\!]$ by *corecursion* as follows:

- $[\![P]\!] =$ ✠, if $P \longrightarrow^*$ ✠;
- $[\![P]\!] = x|\overline{a}\langle [\![N_1]\!], \dots, [\![N_n]\!]\rangle$, if $P \longrightarrow^* \ x|\overline{a}\langle N_1, \dots, N_n\rangle$;
- $[\![P]\!] = \Omega$, otherwise (*i.e.*, if either $P \longrightarrow^* \ \Omega$ or $P \longrightarrow\longrightarrow \cdots$ diverges);
- $[\![\sum a(\boldsymbol{x}).P_a]\!] = \sum a(\boldsymbol{x}).[\![P_a]\!]$.

An important property of ludics is **associativity** of normalization:

$$[\![D[N_1/x_1, \dots, N_n/x_n]]\!] = [\![[\![D]\!][[\![N_1]\!]/x_1, \dots, [\![N_n]\!]/x_n]]\!].$$

A design is said:

- **total**, if $D \neq \Omega$ (not to be confused with totality of strategies in game semantics);
- **linear** (or **affine**), if for any subdesign of the form $N_0|a\langle N_1, \dots, N_n\rangle$, the sets $\mathsf{fv}(N_0)$, $\dots$, $\mathsf{fv}(N_n)$ are pairwise disjoint;
- **cut-free**, if it does not contain a cut as a subdesign.

Designs which are total, linear and cut-free correspond to the original notion of design [9] and for this reason we call them **standard**.

A very important subclass of standard designs is the one consisting of **atomic designs**. A positive standard design P is **atomic** if $\mathsf{fv}(P) \subseteq \{x_0\}$ for a certain *fixed* variable x_0 (the variable x_0 plays the same role as the empty address "$\langle\rangle$" does in [9], *i.e.*, it is a fixed and predetermined "location"). A negative standard design N is **atomic** if $\mathsf{fv}(N) = \emptyset$. In the sequel, we denote by $\mathcal{D}$ the set of the *atomic designs*, by $\mathcal{D}^+$ (resp. $\mathcal{D}^-$) its restriction to positive (resp. negative) designs.

2.2 Functionals

We now introduce a class of designs that work as morphisms between behaviours.

Definition 2.3 A **functional** $(f, g, h, \ldots)$ is any negative standard design N such that $\mathsf{fv}(N) \subseteq \{x_0\}$.

A functional f can be thought as a polarity preserving map $f : \mathcal{D} \longrightarrow \mathcal{D}$. Indeed, given an atomic positive design P, we can apply f to P by $f^*(P) := [\![P[f/x_0]]\!]$. The result is either a positive atomic design or Ω, which can be seen as a coding of "undefined." So, the operation f^* can be seen as a partial map $f^* : \mathcal{D}^+ \rightharpoonup \mathcal{D}^+$.

Similarly, given an atomic negative design N, $f_*(N) := [\![f[N/x_0]]\!]$ is an atomic negative design. So, $f_* : \mathcal{D}^- \longrightarrow \mathcal{D}^-$.

By associativity, we immediately have the following duality principle:

Lemma 2.4 *For any $P \in \mathcal{D}^+$ and $N \in \mathcal{D}^-$, $[\![f^*(P)[N/x_0]]\!] = [\![P[f_*(N)/x_0]]\!]$.*

We now introduce some basic functionals and their notation.

Identity: We call *identity* (also called *fax* in [9], or copycat strategy in game semantics) the functional *id* corecursively defined by the equation:

$$id := \sum a(x_1, \ldots, x_n).x_0|\overline{a}\langle id(x_1), \ldots id(x_n)\rangle,$$

where $id(x_k) := id[x_k/x_0]$ for any $1 \leq k \leq n$. The design *id* plays the role of the identity function for (standard) designs, in particular: $id^*(P) = P$, for any $P \in \mathcal{D}^+$ and $id_*(N) = N$, for any $N \in \mathcal{D}^-$ (see [17]).

Renaming: Given two n-ary names a, b, we call *renaming* the functional $rn_{(a,b)}$ defined as:

$$rn_{(a,b)} := a(x_1, \ldots, x_n).x_0|\overline{b}\langle id(x_1), \ldots id(x_n)\rangle.$$

When $rn_{(a,b)}$ is applied to a positive atomic design P, it works as follows. For $P = x_0|\overline{c}\langle N_1, \ldots, N_n\rangle$, we have $rn^*_{(a,b)}(P) = x_0|\overline{b}\langle N_1, \ldots, N_n\rangle$ if $c = a$ (in this case $rn_{(a,b)}$ just "renames" the first action of P); $rn^*_{(a,b)}(P) = \Omega$, *i.e.*, "undefined" when $c \neq a$. We use the special notation id_a for the identical renaming $rn_{(a,a)}$. Notice that $\sum id_a = id$.

Functionals can be composed: for any f, g, we define the *positive composition* $g \circ^* f := [\![f[g/x_0]]\!]$ and the *negative composition* $g \circ_* f := [\![g[f/x_0]]\!]$.

Example 2.5 $rn_{(a,b)} \circ^* rn_{(b,a)} = rn_{(b,a)} \circ_* rn_{(a,b)} = id_b$.

The operations $\circ^*$ and $\circ_*$ are associative with unit id. Furthermore,

Proposition 2.6 (Untyped composition) *For any $P \in \mathcal{D}^+$ and $N \in \mathcal{D}^-$,*

$$(g \circ^* f)^*(P) = g^*(f^*(P)), \qquad (g \circ_* f)_*(N) = g_*(f_*(N)).$$

We have thus constructed two categories $\mathcal{P}_0$ and $\mathcal{N}_0$: $\mathcal{P}_0$ (resp. $\mathcal{N}_0$) has $\mathcal{D}^+$ (resp. $\mathcal{D}^-$) as the unique object and functionals as morphisms. While the set of morphisms is the same in $\mathcal{P}_0$ and $\mathcal{N}_0$, composition is defined differently as described above. The fundamental principle is that $\mathcal{P}_0^{op} \cong \mathcal{N}_0$. We shall later refine them to the categories of positive and negative behaviours.

3 Synthetic connectives

3.1 Synthetic signature

Let $\mathcal{A} = (A, \mathsf{ar})$ be a signature. Let A^n be the set of names of arity n, *i.e.*, $A^n := \{a \in A : \mathsf{ar}(a) = n\}$. A signature is **synthetic** if:

- for any $a \in A^n$, $b \in A^m$ and $1 \leq i \leq n$, there exists a name $a[b/i] \in A^{n+m-1}$;
- $a[b/i] = c[d/j]$ only if $a = c$, $b = d$ and $i = j$.

From now on, we assume that our signature $\mathcal{A} = (A, \mathsf{ar})$ is synthetic and equipped with a unary name $\uparrow$, that we call the *dummy shift operator*. We denote by $\downarrow$ the positive action $\overline{\uparrow}$, and abbreviate $\uparrow(x).x|\overline{a}\langle \boldsymbol{N}\rangle$ by $\uparrow\overline{a}\langle \boldsymbol{N}\rangle$.

As a convention, given disjoint sequences of variables $\boldsymbol{x} = x_1, \dots, x_n$ and $\boldsymbol{y}$ and $1 \leq i \leq n$, we denote by $\boldsymbol{x}[\boldsymbol{y}/i]$ the sequence $x_1, \dots, x_{i-1}, \boldsymbol{y}, x_{i+1}, \dots, x_n$.

3.2 Logical and synthetic connectives

Informally, a logical connective is specified by *(i) placeholders* for subformulas, and *(ii) inference rules* associated to the connective. In our setting, *(i)* is embodied by a sequence of variables and *(ii)* by a set of negative actions as follows:

Definition 3.1 Let $\boldsymbol{z} = z_1, \dots, z_n$ be a sequence of distinct variables. An **n-ary logical connective** $\alpha(z_1, \dots, z_n)$ is a finite set of negative actions $\{a_1(\boldsymbol{x}_1), \dots, a_m(\boldsymbol{x}_m)\}$, such that $a_1, \dots, a_m$ are distinct names and $\{\boldsymbol{x}_i\} \subseteq \{\boldsymbol{z}\}$, for any $1 \leq i \leq m$.

Since variables are just used as placeholders, we naturally identify two logical connectives if one is obtained from another by renaming the variables. In other words, the variables $\boldsymbol{z}$ are bound in the expression $\alpha(\boldsymbol{z})$. Hence, given two logical connectives $\alpha(\boldsymbol{z})$ and $\beta(\boldsymbol{u})$, we may always assume that $\boldsymbol{z}$ and $\boldsymbol{u}$ are disjoint. When the variables $\boldsymbol{z}$ are clear from the context, we often write α instead of $\alpha(\boldsymbol{z})$.

A synthetic signature allows us to *synthesize* two logical connectives.

Definition 3.2 Let $\alpha(z_1, \dots, z_n)$ and $\beta(u_1, \dots, u_m)$ be logical connectives. Given $1 \leq i \leq n$, we call **synthetic connective** associated to (α, β, i), noted by $\mathsf{synth}(\alpha, \beta, i)$, the logical connective $\gamma(z_1, \dots, z_{i-1}, u_1, \dots, u_m, z_{i+1}, \dots, z_n)$ consisting of negative actions:

(a) $a[b/j](\boldsymbol{x}[\boldsymbol{y}/j])$ such that $a(\boldsymbol{x}) \in \alpha$, $b(\boldsymbol{y}) \in \beta$, $\boldsymbol{x} = x_1, \ldots, x_k$ and $z_i = x_j$;

(b) $a\beta(\boldsymbol{x})$ such that $a(\boldsymbol{x}) \in \alpha$ and $z_i \notin \{\boldsymbol{x}\}$, where $a\beta$ is a fresh name of arity $\mathsf{ar}(a)$.

Observe that the freshness condition in (b) above ensures that the actions in $\mathsf{synth}(\alpha, \beta, i)$ have pairwise distinct names.

Standard connectives, shifts and synthetic connectives of **MALL** can be expressed in a synthetic signature containing unary names $\uparrow, \pi_1, \pi_2$ and a binary name $\wp$. We define:

$$\top := \emptyset, \qquad \boldsymbol{⅋}(x_1, x_2) := \{\wp(x_1, x_2)\},$$

$$\boldsymbol{\uparrow}(x) := \{\uparrow(x)\}, \qquad \boldsymbol{\&}(y_1, y_2) := \{\pi_1(y_1), \pi_2(y_2)\}.$$

We may build a synthetic connective $\mathsf{synth}(\boldsymbol{⅋}, \boldsymbol{\&}, 1) = \gamma(y_1, y_2, x_2) = \{\wp[\pi_1/1](y_1, x_2), \wp[\pi_2/1](y_2, x_2)\}$, which is a logical connective with inference rules:

$$\frac{\vdash \Gamma, \mathbf{P}, \mathbf{R} \quad \vdash \Gamma, \mathbf{Q}, \mathbf{R}}{\vdash \Gamma, \gamma(\mathbf{P}, \mathbf{Q}, \mathbf{R})}\,\gamma \qquad \frac{\vdash \Gamma, \mathbf{N} \quad \vdash \Delta, \mathbf{K}}{\vdash \Gamma, \Delta, \overline{\gamma}(\mathbf{N}, \mathbf{M}, \mathbf{K})}\,\overline{\gamma}_1 \qquad \frac{\vdash \Gamma, \mathbf{M} \quad \vdash \Delta, \mathbf{K}}{\vdash \Gamma, \Delta, \overline{\gamma}(\mathbf{N}, \mathbf{M}, \mathbf{K})}\,\overline{\gamma}_2$$

It is clear that the rule γ is a combination of the standard **MALL** rules for & and ⅋, while $\overline{\gamma}_1$ and $\overline{\gamma}_2$ are combinations of the rules for $\otimes$, $\oplus_1$ and $\oplus_2$. Indeed, thinking of $\gamma(\mathbf{P}, \mathbf{Q}, \mathbf{R})$ as $(\mathbf{P}\&\mathbf{Q})⅋\mathbf{R}$ and $\overline{\gamma}(\mathbf{N}, \mathbf{M}, \mathbf{K})$ as $(\mathbf{N}\oplus\mathbf{M})\otimes\mathbf{K}$, we have:

$$\frac{\dfrac{\vdash \Gamma, \mathbf{P}, \mathbf{R} \quad \vdash \Gamma, \mathbf{Q}, \mathbf{R}}{\vdash \Gamma, \mathbf{P}\&\mathbf{Q}, \mathbf{R}}\,\&}{\vdash \Gamma, (\mathbf{P}\&\mathbf{Q})⅋\mathbf{R}}\,⅋ \qquad \frac{\dfrac{\vdash \Gamma, \mathbf{N}}{\vdash \Gamma, \mathbf{N} \oplus \mathbf{M}}\,\oplus_1 \quad \vdash \Delta, \mathbf{K}}{\vdash \Gamma, \Delta, (\mathbf{N}\oplus\mathbf{M})\otimes\mathbf{K}}\,\otimes \qquad \frac{\dfrac{\vdash \Gamma, \mathbf{M}}{\vdash \Gamma, \mathbf{N} \oplus \mathbf{M}}\,\oplus_2 \quad \vdash \Delta, \mathbf{K}}{\vdash \Gamma, \Delta, (\mathbf{N}\oplus\mathbf{M})\otimes\mathbf{K}}\,\otimes$$

4 Focalizing designs

In the sequel, given a sequence of variables $\boldsymbol{z} = z_1, \ldots, z_k$ we denote by the expression $id(\boldsymbol{z})$ the sequence of functionals $id(z_1), \ldots, id(z_k)$. With this notation, id can be succinctly expressed by $\sum a(\boldsymbol{x}).x_0|\overline{a}\langle id(\boldsymbol{x})\rangle$.

Given logical connectives $\alpha(z_1, \ldots, z_n)$, $\beta(u_1, \ldots, u_m)$ and $1 \leq i \leq n$, we define two functionals: **focalizing design** $f_{(\alpha,\beta,i)}$ and **inverting design** $u_{(\alpha,\beta,i)}$ as follows.

The focalizing design is built from $|\alpha|$ components indexed by (names of) actions $a(\boldsymbol{x}) \in \alpha$, whereas the inverting one from $|\gamma|$ components indexed by (names of) actions $c(\boldsymbol{v}) \in \gamma$, where $\gamma = \mathsf{synth}(\alpha, \beta, i)$. We have two sorts of components (LHS and RHS of the table below), corresponding to the two cases (a), (b) of Definition 3.2.

For $a(\boldsymbol{x}) \in \alpha$ with $\boldsymbol{x} = x_1, \ldots, x_k$, we set:

if $z_i = x_j$	if $z_i \notin \{\boldsymbol{x}\}$
$f_a := a(\boldsymbol{x}).z_i|\downarrow\langle\sum_\beta b(\boldsymbol{y}).x_0|a[b/j]\langle id(\boldsymbol{x}[\boldsymbol{y}/j])\rangle\rangle$	$f_a := rn_{(a,a\beta)}$
$u_{a[b/j]} := a[b/j](\boldsymbol{x}[\boldsymbol{y}/j]).x_0|a\langle id(\boldsymbol{x}_l), \uparrow b\langle id(\boldsymbol{y})\rangle, id(\boldsymbol{x}_r)\rangle$	$u_{a\beta} := rn_{(a\beta,a)}$

where $\boldsymbol{x} = \boldsymbol{x}_l, x_j, \boldsymbol{x}_r$ in the definition of $u_{a[b/j]}$, and $rn_{(a,a\beta)}$, $rn_{(\beta a,a)}$ are renaming functionals. We finally take the formal sum of the components: $f_{(\alpha,\beta,i)} := \sum_\alpha f_a$ and $u_{(\alpha,\beta,i)} := \sum_\gamma u_c$.

To see how they work, consider:

$$f = f_{(\parr,\&,1)} = \wp(x_1,x_2).x_1|{\downarrow}\langle \textstyle\sum_{i=1,2} \pi_i(y_i).x_0|\wp[\pi_i/1]\langle id(y_i), id(x_2)\rangle\rangle,$$

$$u = u_{(\parr,\&,1)} = \textstyle\sum_{i=1,2} \wp[\pi_i/1](y_i,x_2).x_0|\overline{\wp}\langle \uparrow\overline{\pi}_i\langle id(y_i)\rangle, id(x_2)\rangle.$$

Consider also the following atomic designs:

$$P_1 := x_0|\overline{\wp}\langle\uparrow \overline{\pi_1}\langle M\rangle, N\rangle,\ N_1 := \wp(x_1,x_2).x_1|{\downarrow}\langle \pi_1(y_1).Q_1 + \pi_2(y_2).Q_2\rangle,$$

$$P_2 := x_0|\wp[\pi_1/1]\langle M, N\rangle,\ N_2 := \wp[\pi_1/1](y_1,x_2).Q_1 + \wp[\pi_2/1](y_2,x_2).Q_2.$$

We can calculate $f^*(P_1)$ by normalization:

$$\begin{aligned} f^*(P_1) &= [\![f \mid \overline{\wp}\langle\uparrow\overline{\pi_1}\langle M\rangle, N\rangle]\!] \\ &= [\![(\uparrow\pi_1\langle M\rangle) \mid {\downarrow}\langle \pi_1(y_1).x_0|\wp[\pi_1/1]\langle id(y_1), id(x_2)[N/x_2]\rangle\rangle]\!] \\ &= [\![(\pi_1(y_1).x_0|\wp[\pi_1/1]\langle id(y_1), id(x_2)[N/x_2]\rangle) \mid \pi_1\langle M\rangle]\!] \\ &= [\![x_0|\wp[\pi_1/1]\langle id(y_1)[M/y_1], id(x_2)[N/x_2]\rangle]\!] \\ &= x_0|\wp[\pi_1/1]\langle id_*(M), id_*(N)\rangle = P_2. \end{aligned}$$

Similarly, we obtain $u^*(P_2) = P_1$, $u_*(N_1) = N_2$, $f_*(N_2) = N_1$.

Observe that f^* (resp. u_*) "collapses" three polarity layers into one when applied to a positive (resp. negative) design, while u^* (resp. f_*) "cancels" the effect of f^* (resp. u_*). In particular, we could informally claim that f^* *internalize* focalization as morphism. We shall see later that u_* can be seen as an internal expression of the invertibility of negative connectives.

Proposition 4.1 (Focalization-inversion) *Let* $f = f_{(\alpha,\beta,i)}$ *and* $u = u_{(\alpha,\beta,i)}$.

- $f \circ^* u = id_\gamma$, *where* $\gamma = \mathsf{synth}(\alpha,\beta,i)$ *and* $id_\gamma = \sum_{c(\boldsymbol{x})\in\gamma} id_c$.
- $u \circ^* f$ *is idempotent:* $(u \circ^* f) \circ^* (u \circ^* f) = u \circ^* f$.

The equation $f \circ^* u = id_\gamma$ roughly states that f and u are opposite operations. Later we shall state more precisely that focalizing designs are retractions of inverting designs. That will formally verify the intuition that focalization of positive rules is dual to invertibility of negative rules [9].

5 Functionals on behaviours

We first recall the notion of behaviour and then discuss functionals in the typed setting.

5.1 Orthogonality and behaviours

Two atomic designs P, N of opposite polarities are said **orthogonal** (written $P \perp N$) when $[\![P[N/x_0]]\!] = \maltese$. If $\mathbf{X}$ is a set of atomic designs of the same polarity, then its **orthogonal set** is defined by $\mathbf{X}^{\perp} := \{E : \forall D \in \mathbf{X},\ D \perp E\}$.

The duality of Lemma 2.4 is nicely expressed in terms of orthogonality as:

$$f^*(P) \perp N \text{ if and only if } P \perp f_*(N). \tag{1}$$

A **behaviour** is a set $\mathbf{X}$ of atomic designs of the same polarity such that $\mathbf{X}^{\perp\perp} = \mathbf{X}$; according to the polarity of its designs, it is positive and noted by letters $\mathbf{P}, \mathbf{Q}, \mathbf{R}, \dots$ or negative and noted by $\mathbf{N}, \mathbf{M}, \mathbf{K} \dots$.

There are the least and the greatest behaviours among all positive (resp. negative) behaviours with respect to set inclusion:

$\mathbf{0} := \{\maltese\}$	$\underline{\mathbf{0}} := \{\maltese^-\}$	$\underline{\top} := \underline{\mathbf{0}}^{\perp} (= \mathcal{D}^+)$	$\top := \mathbf{0}^{\perp} (= \mathcal{D}^-)$

where $\maltese^- := \sum a(\boldsymbol{x}).\maltese$ is called *negative daimon* in [9].

We are now ready to assign "types" to functionals.

Definition 5.1 Let $\mathbf{P}, \mathbf{Q}$ be positive behaviours. We define the positive **function space** as the set of functionals $\mathbf{P} \longrightarrow \mathbf{Q} := \{f : \forall P \in \mathbf{P}, f^*(P) \in \mathbf{Q}\}$. Analogously, given negative behaviours $\mathbf{N}, \mathbf{M}$, we define $\mathbf{N} \longrightarrow \mathbf{M} := \{f : \forall N \in \mathbf{N}, f_*(N) \in \mathbf{M}\}$. We write $f : \mathbf{X} \longrightarrow \mathbf{Y}$ whenever $f \in \mathbf{X} \longrightarrow \mathbf{Y}$.

For instance, we have $id : \mathbf{P} \longrightarrow \mathbf{P}$ and $id : \mathbf{N} \longrightarrow \mathbf{N}$ for any $\mathbf{P}, \mathbf{N}$.

We also have the following characterization of function spaces:

$$f : \mathbf{P} \longrightarrow \mathbf{Q} \iff \forall N \in \mathbf{Q}^{\perp}, \forall P \in \mathbf{P}, [\![P[f[N/x_0]/x_0]]\!] = \maltese;$$

$$f : \mathbf{N} \longrightarrow \mathbf{M} \iff \forall P \in \mathbf{M}^{\perp}, \forall N \in \mathbf{N}, [\![P[f[N/x_0]/x_0]]\!] = \maltese.$$

As an immediate consequence, we have $f : \mathbf{P} \longrightarrow \mathbf{Q} \iff f : \mathbf{Q}^{\perp} \longrightarrow \mathbf{P}^{\perp}$. Also, the above makes clear that every function space is a "nonatomic" behaviour.

Composition of typed functionals is naturally typed: if $f : \mathbf{P} \longrightarrow \mathbf{Q}$, $g : \mathbf{Q} \longrightarrow \mathbf{R}$ and $f' : \mathbf{N} \longrightarrow \mathbf{M}$, $g' : \mathbf{M} \longrightarrow \mathbf{K}$, we have by Proposition 2.6:

Proposition 5.2 (Composition) *$g \circ^* f : \mathbf{P} \longrightarrow \mathbf{R}$ and $g' \circ_* f' : \mathbf{N} \longrightarrow \mathbf{K}$.*

Given identity and composition, it is tempting to think of a category in which objects are positive (or negative) behaviours and morphisms are functionals. This, however, does not work, since the identity morphism is not unique. As an extreme case, consider the "minimal" positive function space $\mathbf{0} \longrightarrow \mathbf{0}$. In the usual sense, there is only one (total) function here: the one which sends $\maltese$ to $\maltese$. On the other hand, $\mathbf{0} \longrightarrow \mathbf{0}$ contains all the functionals, since we have $f^*(\maltese) = \maltese$ for any f. They all play the same role, sending $\maltese$ to $\maltese$. Hence they are *equivalent from the viewpoint of* $\mathbf{0} \longrightarrow \mathbf{0}$.

To obtain a category, we therefore need to quotient functionals $f : \mathbf{X} \longrightarrow \mathbf{Y}$ by such an equivalence relation depending on $\mathbf{X}$ and $\mathbf{Y}$. Similarly, elements of a behaviour $\mathbf{X}$ can be equipped with such a relation: D and E are equivalent in $\mathbf{X}$ whenever they only differ in *useless parts i.e.*, subdesigns which play no active role when normalizing against designs of $\mathbf{X}^{\perp}$. Interestingly, ludics is already equipped with such a relation, "equality up to *materiality*" [9].

5.2 Materiality, section, retraction and isomorphism

Informally, two functionals $f, g \in \mathbf{X} \longrightarrow \mathbf{Y}$, are "equal up to materiality" in $\mathbf{X} \longrightarrow \mathbf{Y}$ if they share the "minimal" part h which is really necessary during *any* computation with designs of $\mathbf{X}$ and $\mathbf{Y}^{\perp}$. For example, given $f : \mathbf{0} \longrightarrow \mathbf{0}$, no part of f is necessary for computations, because $[\![\maltese[f/x_0]]\!]$ immediately gives $\maltese$, whatever f is.

To formalize this concept, we first recall the notion of **stable ordering** $\sqsubseteq$ between designs [9,5,17]. Informally, $D \sqsubseteq E$ whenever D is obtained from E by replacing some positive subdesign with Ω. Formally, $\sqsubseteq$ is the largest binary relation on standard designs such that:

- if $\Omega \sqsubseteq D$, then D is a positive design;
- if $\maltese \sqsubseteq D$, then $D = \maltese$;
- if $x|\overline{a}\langle N_1, \dots, N_n\rangle \sqsubseteq D$, then $D = x|\overline{a}\langle M_1, \dots, M_n\rangle$, $N_i \sqsubseteq M_i$, $\forall i, 1 \leq i \leq n$;
- if $\sum a(\boldsymbol{x}).P_a \sqsubseteq D$ then $D = \sum a(\boldsymbol{x}).Q_a$ and $P_a \sqsubseteq Q_a$ for every $a \in A$.

An important property of the stable ordering is **monotonicity**: given standard designs D, E, N, M such that $D \sqsubseteq E$ and $N \sqsubseteq M$, we have $[\![D[N/x_0]]\!] \sqsubseteq [\![E[M/x_0]]\!]$ (see [17]).

We also define the **intersection** $\cap$ of standard designs by corecursion as:

- $P \cap \Omega = \Omega \cap P = \Omega$;
- $\maltese \cap \maltese = \maltese$;
- $x|\overline{a}\langle N_1, \dots, N_n\rangle \cap x|\overline{a}\langle M_1, \dots, M_n\rangle = x|\overline{a}\langle N_1 \cap M_1, \dots, N_n \cap M_n\rangle$ if $N_i \cap M_i$ are defined for every $1 \leq i \leq n$;
- $\sum a(\boldsymbol{x}).P_a \cap \sum a(\boldsymbol{x}).Q_a = \sum a(\boldsymbol{x}).(P_a \cap Q_a)$ if $P_a \cap Q_a$ is defined $\forall a \in A$;
- $D \cap E$ is not defined otherwise.

Let $\mathbf{A}$ be a behaviour $\mathbf{X}$ or function space $\mathbf{X} \longrightarrow \mathbf{Y}$. We define the **material part** of $D \in \mathbf{A}$ as $|D|_{\mathbf{A}} := \bigcap\{E \sqsubseteq D : E \in \mathbf{A}\}$. Designs D, E are said **equal up to materiality** in $\mathbf{A}$, $D \sim_{\mathbf{A}} E$, whenever $|D|_{\mathbf{A}} = |E|_{\mathbf{A}}$.

The definition of materiality is justified by the fact that $|D|_{\mathbf{A}}$ is *the minimal design* in $\mathbf{A}$ such that $|D|_{\mathbf{A}} \sqsubseteq D$ (see [9,17] for a proof). So, each equivalence class induced by $\sim_{\mathbf{A}}$ has a canonical and unique representative in $\mathbf{A}$.

For example, in the function space $\mathbf{0} \longrightarrow \mathbf{0}$ all the functionals are equal up to materiality. Indeed, for any f we have $|f|_{\mathbf{0} \longrightarrow \mathbf{0}} = 0$, where $0 := \sum a(\boldsymbol{x}).\Omega$ (called *negative skunk* in [9]) is the minimal negative design *w.r.t.* $\sqsubseteq$.

Lemma 5.3 *$D \sim_{\mathbf{A}} E$ if and only if $\exists F \in \mathbf{A}$ such that $F \sqsubseteq D$ and $F \sqsubseteq E$.*

Theorem 5.4 (Preservation of $\sim$)

(i) *If $P \sim_{\mathbf{P}} Q$ and $f : \mathbf{P} \longrightarrow \mathbf{Q}$ then $f^*(P) \sim_{\mathbf{Q}} f^*(Q)$;*

(ii) *If $P \in \mathbf{P}$ and $f \sim_{\mathbf{P} \longrightarrow \mathbf{Q}} g$ then $f^*(P) \sim_{\mathbf{Q}} g^*(P)$.*

Similarly for negative behaviours and negative function spaces.

Proof. (i) By Lemma 5.3, if $P \sim_{\mathbf{P}} Q$ then $\exists R \in \mathbf{P}$ such that $R \sqsubseteq P$ and $R \sqsubseteq Q$. Applying f, we have that $f^*(R) \in \mathbf{Q}$ and by monotonicity $f^*(R) \sqsubseteq f^*(P)$ and $f^*(R) \sqsubseteq f^*(Q)$. By Lemma 5.3 again, we conclude $f^*(P) \sim_{\mathbf{Q}} f^*(Q)$. For (ii) and for negatives we use a similar reasoning. □

We are now ready to build the category $\mathcal{P}$ of positive behaviours: the objects of $\mathcal{P}$ are positive behaviours and $Hom_{\mathcal{P}}(\mathbf{P}, \mathbf{Q})$ consists of equivalence classes of functionals $f : \mathbf{P} \longrightarrow \mathbf{Q}$ with respect to $\sim_{\mathbf{P} \longrightarrow \mathbf{Q}}$. the category $\mathcal{N}$ of negative behaviours is defined similarly. As before, we have $\mathcal{P}^{op} \cong \mathcal{N}$ where the isomorphism is given by $\mathbf{P} \mapsto \mathbf{P}^{\perp}$ on objects and identity on morphisms.

We may thus employ categorical concepts in ludics. In particular, we have the notions of section, retraction and isomorphism, which may be described in elementary terms as follows:

Definition 5.5 Let $\mathbf{P}, \mathbf{Q}$ be positive behaviours and r, s be functionals such that $r : \mathbf{P} \longrightarrow \mathbf{Q}$ and $s : \mathbf{Q} \longrightarrow \mathbf{P}$. When $r \circ^* s \sim_{\mathbf{Q} \longrightarrow \mathbf{Q}} id$, we say that s is a **section** of r and r is a **retraction** of s. If $s \circ^* r \sim_{\mathbf{P} \longrightarrow \mathbf{P}} id$ holds in addition, s and r are called **isomorphisms**. Similarly for negatives.

In Section 7, we shall show that $f_{(\alpha,\beta,i)}$ is a retraction of $u_{(\alpha,\beta,i)}$. To state it precisely, we shall however need to clarify in which function spaces they live.

6 Logical behaviours

We now describe how behaviours can be generated by the logical connectives we gave in Section 3.2. Let $\alpha(z_1, \ldots, z_n) = \{a_1(\boldsymbol{x}_1), \ldots, a_m(\boldsymbol{x}_m)\}$ be an n-ary logical connective. Since $\{\boldsymbol{x}_i\} \subseteq \{\boldsymbol{z}\}$ for any $1 \le i \le m$, $\boldsymbol{x}$ it is of the form $z_{i_1}, \ldots, z_{i_k}$ with $k = \mathsf{ar}(a_i)$ and $i_1, \ldots, i_k \in \{1, \ldots, n\}$. Let $\mathbf{N}_1, \ldots, \mathbf{N}_n, \mathbf{P}_1, \ldots, \mathbf{P}_n$ be arbitrary behaviours. We define:

- $\alpha\langle \mathbf{N}_1, \ldots, \mathbf{N}_n \rangle := \left(\bigcup_{1 \le i \le m} a_i \langle \mathbf{N}_{i_1}, \ldots, \mathbf{N}_{i_k)} \rangle \right)^{\perp\perp}$, where indices $i_1, \ldots, i_k$ are given by $\boldsymbol{x}_i = z_{i_1}, \ldots, z_{i_k}$ for each $1 \le i \le m$ and $a_i\langle \mathbf{N}_{i_1}, \ldots, \mathbf{N}_{i_k} \rangle := \{x_0 | a_i\langle N_1, \ldots, N_k \rangle : N_1 \in \mathbf{N}_{i_1}, \ldots, N_k \in \mathbf{N}_{i_k}\}$;
- $\alpha(\mathbf{P}_1, \ldots, \mathbf{P}_n) := \alpha\langle \mathbf{P}_1^{\perp}, \ldots, \mathbf{P}_n^{\perp} \rangle^{\perp}$.

Categorically, every n-ary logical connective α defines (covariant) functors $\alpha : \mathcal{P}^n \longrightarrow \mathcal{N}$ and $\alpha : \mathcal{N}^n \longrightarrow \mathcal{P}$.

A remarkable property of logical connectives is **internal completeness** [9]: we can give a precise and direct description of the elements in logical behaviours without using the orthogonality nor referring to any proof system:

- $\overline{\alpha}\langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle = \bigcup_{1 \le i \le m} \overline{a}_i \langle \mathbf{N}_{i_1}, \dots, \mathbf{N}_{i_k} \rangle \cup \{\maltese\}$.
- $\alpha(\mathbf{P}_1, \dots, \mathbf{P}_n) = \{\sum a_i(\boldsymbol{x}_i).P_{a_i} : P_{a_i} \models z_{i_1} : \mathbf{P}_{i_1}, \dots, z_{i_k} : \mathbf{P}_{i_k}$ for every $1 \le i \le m\}$, where the expression $P_{a_i} \models z_{i_1} : \mathbf{P}_{i_1}, \dots, z_{i_k} : \mathbf{P}_{i_k}$ is a short for $\forall N_1 \in \mathbf{P}_{i_1}{}^{\perp}, \dots, \forall N_k \in \mathbf{P}_{i_k}{}^{\perp}$, $[\![P_{a_i}[N_1/z_{i_1}, \dots, N_k/z_{i_k}]]\!] = \maltese$. Notice that the components $b(\boldsymbol{y}).P_b$'s can be arbitrary when $b(\boldsymbol{y}) \notin \alpha$.

Recall that we have expressed the standard **MALL** connectives $\top, \uparrow, \invamp, \&$ as logical connectives in Section 3.2. With these logical connectives we can build (semantic versions of) usual linear logic types. For sake of readability, we use the notation $\mathbf{0} = \overline{\top}$, $\downarrow = \overline{\uparrow}$, $\bullet = \overline{\wp}$, $\otimes = \overline{\invamp}$, $\iota_i = \overline{\pi_i}$, and $\oplus = \overline{\&}$.

By using the infix notation and taking into account the internal completeness, we obtain concrete descriptions (irrelevant components of sums are suppressed by "$\cdots$"):

$$
\begin{aligned}
\mathbf{N} \otimes \mathbf{M} &= \bullet\langle \mathbf{N}, \mathbf{M} \rangle \cup \{\maltese\}, & \mathbf{P} \invamp \mathbf{Q} &= \{\wp(x_1, x_2).P + \cdots : P \models x_1 : \mathbf{P}, x_2 : \mathbf{Q}\},\\
\mathbf{N} \oplus \mathbf{M} &= \iota_1\langle \mathbf{N} \rangle \cup \iota_2\langle \mathbf{M} \rangle \cup \{\maltese\}, & \mathbf{P} \,\&\, \mathbf{Q} &= \{\pi_1(x_0).P + \pi_2(x_0).Q + \cdots : P \in \mathbf{P}, Q \in \mathbf{Q}\},\\
\downarrow \mathbf{N} &= \downarrow \langle \mathbf{N} \rangle \cup \{\maltese\}, & \uparrow \mathbf{P} &= \{\uparrow(x_0).P + \cdots : P \in \mathbf{P}\},\\
\mathbf{0} &= \{\maltese\}, & \top &= \mathcal{D}^-.
\end{aligned}
$$

As to the synthetic connective $\gamma = \mathsf{synth}(\invamp, \&, 1)$, we have:

$$
\begin{aligned}
\overline{\gamma}(\mathbf{N}, \mathbf{M}, \mathbf{K}) &= \wp[\pi_1/1]\langle \mathbf{N}, \mathbf{K} \rangle \cup \wp[\pi_2/1]\langle \mathbf{M}, \mathbf{K} \rangle \cup \{\maltese\};\\
\gamma(\mathbf{P}, \mathbf{Q}, \mathbf{R}) &= \{\wp[\pi_1/1](y_1, x_2).P + \wp[\pi_2/1](y_2, x_2).Q + \cdots\\
&\qquad : P \models y_1 : \mathbf{P}, x_2 : \mathbf{R} \text{ and } Q \models y_2 : \mathbf{Q}, x_2 : \mathbf{R}\}.
\end{aligned}
$$

7 An analysis of focalization in ludics

Let us collect all we have done so far in order to obtain our main results.

7.1 Focalization in behaviours

Let $\alpha(z_1, \dots, z_n)$, $\beta(u_1, \dots, u_m)$ be logical connectives, $1 \le i \le n$ and $\mathbf{N}_1, \dots, \mathbf{N}_{i-1}, \mathbf{M}_1, \dots, \mathbf{M}_m, \mathbf{N}_{i+1}, \dots, \mathbf{N}_n$ behaviours. We can form an "unsynthesized" behaviour $\mathbf{P}$ and a "synthesized" one $\mathbf{Q}$ by:

$$
\begin{aligned}
\mathbf{P} &= \alpha\langle \mathbf{N}_1, \dots, \mathbf{N}_{i-1}, \uparrow\left(\beta\langle \mathbf{M}_1 \dots \mathbf{M}_m \rangle\right), \mathbf{N}_{i+1}, \dots, \mathbf{N}_n \rangle,\\
\mathbf{Q} &= \gamma\langle \mathbf{N}_1, \dots, \mathbf{N}_{i-1}, \mathbf{M}_1 \dots \mathbf{M}_m, \mathbf{N}_{i+1}, \dots, \mathbf{N}_n \rangle,
\end{aligned}
$$

where $\gamma = \mathsf{synth}(\alpha, \beta, i)$. As we have noted in the introduction, the internal completeness theorem provides an easy proof to the following:

Proposition 7.1 *For* $\mathbf{P}$ *and* $\mathbf{Q}$ *as above, we have that*

$$
f_{(\alpha,\beta,i)} : \mathbf{P} \longrightarrow \mathbf{Q}, \qquad u_{(\alpha,\beta,i)} : \mathbf{Q} \longrightarrow \mathbf{P}.
$$

Hence, if $P \sim_{\mathbf{P}} P'$ *then* $f^*_{(\alpha,\beta,i)}(P) \sim_{\mathbf{Q}} f^*_{(\alpha,\beta,i)}(P')$. *The same for* $u_{(\alpha,\beta,i)}$.

Proof. For instance, suppose that $\mathbf{P} = (\uparrow(\mathbf{M}_1 \oplus \mathbf{M}_2)) \otimes \mathbf{N}$. If $P \in \mathbf{P}$ and $P \neq \maltese$, then by internal completeness P must be of the form

$$x_0 | \bullet \langle \uparrow(y).y|\iota_j\langle M\rangle, N\rangle,$$

where $j = 1$ or 2, $M \in \mathbf{M}_j$ and $N \in \mathbf{N}$. Given such a concrete description, it is easy to calculate $f_{(\parr,\&,1)}(P)$ and to verify it belongs to $\overline{\gamma}\langle \mathbf{M}_1, \mathbf{M}_2, \mathbf{N}_2\rangle$. The argument is similar for $u_{(\alpha,\beta,i)} : \mathbf{Q} \longrightarrow \mathbf{P}$. □

Before we proceed to the next theorem, let us explain why the functional $u_{(\alpha,\beta,i)}$ is related to invertibility of negative connectives. Consider $\gamma = \mathsf{synth}(\parr, \&, 1)$. Then the behaviour $\gamma(\mathbf{P}, \mathbf{Q}, \mathbf{R})$ can be thought as if it were a "sequent" $\models \mathbf{P} \,\&\, \mathbf{Q}, \mathbf{R}$ by identifying $\parr$ with comma. As we have seen above, internal completeness tells us that every design N in $\gamma(\mathbf{P}, \mathbf{Q}, \mathbf{R})$ *uniformly* arises as $N = \wp[\pi_1/1](y_1, x_2).P + \wp[\pi_2/1](y_2, x_2).Q + \cdots$ from designs $P \models y_1 : \mathbf{P}, x_2 : \mathbf{R}$ and $Q \models y_2 : \mathbf{Q}, x_2 : \mathbf{R}$. This is nothing but invertibility of $\&$ (under the above identification). Notice that we do not have such a uniform description for the unsynthesized behaviour $\downarrow(\mathbf{P} \,\&\, \mathbf{Q}) \parr \mathbf{R}$. On the other hand, the inverting functional $u = u_{(\parr,\&,1)}$ sends a design in the latter to one in $\gamma(\mathbf{P}, \mathbf{Q}, \mathbf{R})$ for which we have a uniform description. This is the reason why we call u an *inverting* design. We finally arrive at a formal statement of our slogan: *focalization is a retraction of invertibility.*

Theorem 7.2 (Section-retraction) *For* $\mathbf{P}$ *and* $\mathbf{Q}$ *as above,* $f_{(\alpha,\beta,i)} : \mathbf{P} \longrightarrow \mathbf{Q}$ *is a retraction of* $u_{(\alpha,\beta,i)} : \mathbf{Q} \longrightarrow \mathbf{P}$.

Proof. By Proposition 7.1, we have that $f = f_{(\alpha,\beta,i)} : \mathbf{P} \longrightarrow \mathbf{Q}$ and $u = u_{(\alpha,\beta,i)} : \mathbf{Q} \longrightarrow \mathbf{P}$ and by Proposition 5.2 that $f \circ^* u : \mathbf{Q} \longrightarrow \mathbf{Q}$. By Proposition 4.1, $f \circ^* u = id_\gamma$ and since $id_\gamma \sqsubseteq id$, $f \circ^* u \sim_{\mathbf{Q} \longrightarrow \mathbf{Q}} id$. □

As we have noted in the introduction, focalization is a context-sensitive phenomenon. Hence in order to claim that the focalizing designs really capture the essence of focalization, we must ensure that they work in context as well. Rather than dealing with "sequents of behaviours" $\models \mathbf{\Gamma}, \mathbf{P}$ (for which the closure principle would be enough), we consider focalization inside a logical connective $\alpha(\mathbf{P}_1, \ldots, \mathbf{P}_n)$.

Theorem 7.3 (Focalization in context) *Let* $\{s_i : \mathbf{Q}_i \longrightarrow \mathbf{P}_i\}_{1 \leq i \leq n}$ *be a family of sections and* $\{r_i : \mathbf{P}_i \longrightarrow \mathbf{Q}_i\}_{1 \leq i \leq n}$ *be the family of corresponding retractions. For any logical connective* $\alpha(z_1, \ldots, z_n)$*, there exists a section* $s : \alpha(\mathbf{Q}_1, \ldots, \mathbf{Q}_n) \longrightarrow \alpha(\mathbf{P}_1, \ldots, \mathbf{P}_n)$ *with a corresponding retraction* r.

That is to say, the section-retraction relation is preserved by logical connective α seen as functor $\alpha : \mathcal{P}^n \longrightarrow \mathcal{N}$. This automatically ensures that focalizing and inverting designs work inside a logical connective as well.

7.2 *Focalization for* MALL

Finally we sketch how to combine our treatment of focalization with full completeness of ludics *w.r.t.* **MALL** [9] to obtain focalization for **MALL**. The argument below is *not* intended to be a real proof, since the focalization theorem for **LL** is already proven many times and our argument is not easier (ours rather involves a delicate issue of exactness/linearity). Our purpose is just to illustrate how our results can be seen as a key "factor" of focalization for **MALL** under suitable factorization.

We consider the constant-only fragment of **MALL**, where formulas are generated by the grammar:

$$\mathsf{F} ::= \ \mathsf{0} \mid \top \mid \mathsf{F} \otimes \mathsf{F} \mid \mathsf{F} \oplus \mathsf{F} \mid \mathsf{F} \parr \mathsf{F} \mid \mathsf{F} \,\&\, \mathsf{F}$$

and rules are just standard. Inductively, we define the following interpretation function $(\_)^\bullet$ which sends a formula into a negative behaviour:

$$\mathsf{0}^\bullet := \uparrow\mathbf{0}, \qquad \top^\bullet := \mathsf{T}, \qquad (\mathsf{F} \otimes \mathsf{F})^\bullet := \uparrow(\mathsf{F}^\bullet \otimes \mathsf{F}^\bullet),$$
$$(\mathsf{F} \oplus \mathsf{F})^\bullet := \uparrow(\mathsf{F}^\bullet \oplus \mathsf{F}^\bullet),\ (\mathsf{F} \parr \mathsf{F})^\bullet := \ \downarrow\mathsf{F}^\bullet \parr \downarrow\mathsf{F}^\bullet,\ (\mathsf{F} \,\&\, \mathsf{F})^\bullet := \ \downarrow\mathsf{F}^\bullet \,\&\, \downarrow\mathsf{F}^\bullet.$$

Let π be a cut-free proof of **MALL** formula F. By soundness theorem, we get a design $D \in \mathsf{F}^\bullet$. Now, we can repeatedly apply focalizing designs $f_1, \ldots, f_n$ (in context) in order to get a design $(f_n \circ^* \ldots \circ^* f_1)^*(D)$ in which positive layers are maximally synthesized. For negative layers, we can apply sequences of inverting designs. We finally get a design D^f in a behaviour $\mathsf{F}_s^\bullet$ which is maximally synthesized. For an example of positive layer, the formula $(\mathsf{F} \oplus \mathsf{G}) \otimes \mathsf{H}$ is sent to $\uparrow(\uparrow(\mathsf{F}^\bullet \oplus \mathsf{G}^\bullet) \otimes \ \uparrow\mathsf{H}^\bullet)$ from which we can find (a maximal) synthetic connective $\uparrow(\overline{\gamma}\langle \mathsf{F}^\bullet, \mathsf{G}^\bullet, \mathsf{H}^\bullet\rangle)$. Now, we can apply the corresponding $u_n, \ldots, u_1$, where u_i are respectively the sections of f_i. We get a new design D^{fu} in $\mathsf{F}^\bullet$ built by **MALL** connectives with shifts and still *focalized*. It is clear that this procedure preserves ✠-freeness (while it is not immediate that it also preserves linearity/exactness [9]). Provided that it preserves exactness, full completeness of ludics yields a proof π' of F in **MALL**.

8 Conclusion and future works

We have attempted to analyze focalization in ludics. More specifically, we have detailed how to synthesize behaviours, and internally described focalization as focalizing functionals (designs) from unsynthesized to synthesized behaviours. We have also pointed out that every focalizing design is a retraction of an inverting design, which is related to invertibility of negative connectives. This in a way formalizes the common intuition that focalization of positives is dual to invertibility of negatives.

Our work naturally leads to several directions for extension and we wish to stress one of them which seems particularly promising. It is related with an analysis of usual computability and complexity theory by logical means. Indeed, our analysis of focalization in ludics was primarily motivated by by the concluding remarks of the third author's paper on computational ludics [17] where focalization on *data designs* (designs which represent usual first order data, such as natural numbers, lists, etc) was conjectured to correspond to the tape compression theorem of Turing machines. Some further work is required to make this correspondence formal and then enlarge our focalization results.

Another important thing to be done is to reformulate ludics from a more general, categorical perspective and to relate our work with other approaches, like polarized categories (for polarized **MALL**) [8] and categories arising from game semantics. In doing so, it would be interesting to see ludics as a *dialogue category*, a fundamental concept for linear logic and polarized systems identified by Melliès and Tabareau [14].

References

[1] Abramsky, S.: Sequentiality vs. concurrency in games and logic. Mathematical Structures in Computer Science **13**(4) (2003) 531–565.

[2] Andreoli, J.-M.: Logic Programming with Focusing Proofs in Linear Logic. Journal of Logic and Computation **2**(3) (1992) 297–347.

[3] Basaldella, M., Saurin, A., Terui, K.: On the meaning of focalization. In: Proceedings of Workshop on Games, Dialogues and Interaction (2009).

[4] Chaudhuri K.: Focusing Strategies in the Sequent Calculus of Synthetic Connectives. In: Proceedings of Logic for Programming, Artificial Intelligence and Reasoning (2008) 467–481.

[5] Curien, P.-L.: Introduction to linear logic and ludics, part II. Advances in Mathematics (China) **35**(1) (2006) 1–44.

[6] Curien, P.-L., Munch-Maccagnoni, G.: The duality of computation under focus. In: Proceedings of IFIP TCS (2010).

[7] Faggian, C., Piccolo, M.: Ludics is a model for the finitary linear pi-calculus. In: Proceedings of Typed Lambda Calculi and Applications (2007) 148–162.

[8] Hamano, M., Scott, P.J.: A Categorical Semantics for Polarized MALL. Annals of Pure and Applied Logic **145**(3) (2007) 276–313.

[9] Girard, J.-Y.: Locus solum: From the rules of logic to the logic of rules. Mathematical Structures in Computer Science **11**(3) (2001) 301–506.

[10] Laurent, O.: Étude de la polarization en logique. PhD thesis, Université Aix-Marseille II (2002).

[11] Laurent, O.: Polarized games. Annals of Pure and Applied Logic **130**(1-3) (2004) 79–123.

[12] Laurent, O.: A proof of the focalization property of Linear Logic. Unpublished note (2004).

[13] Melliès, P.-A.: Asynchronous Games 4: A Fully Complete Model of Propositional Linear Logic. In: Proceedings of Logic in Computer Science (2005) 386–395.

[14] Melliès, P.-A., Tabareau, N.: Resource modalities in tensor logic. Annals of Pure and Applied Logic **161**(5) (2010) 632–653.

[15] Miller, D., Saurin, A.: From proofs to focused proofs: a modular proof of focalization in Linear Logic. In: Proceedings of Computer Science Logic. (2007) 405–419.

[16] Munch-Maccagnoni G.: Focalisation and Classical Realisability In: Proceedings of Computer Science Logic (2009) 409–423.

[17] Terui, K.: Computational ludics. (2008) To appear in Theoretical Computer Science.

[18] Zeilberger, N.: The logical basis of evaluation order and pattern matching. PhD Thesis, Carnegie Mellon University (2009).