Inductive and Functional Types in Ludics*

Alice Pavaux

Université Paris 13, Sorbonne Paris Cité, LIPN, CNRS, UMR 7030

28th July 2017

Abstract

Ludics is a logical framework in which types/formulas are modelled by sets of terms with the same computational behaviour. This paper investigates the representation of inductive data types and functional types in ludics. We study their structure following a game semantics approach. Inductive types are interpreted as least fixed points, and we prove an internal completeness result giving an explicit construction for such fixed points. The interactive properties of the ludics interpretation of inductive and functional types are then studied. In particular, we identify which higher-order functions types fail to satisfy type safety, and we give a computational explanation.

1 Introduction

1.1 Context and Contributions

Context

Ludics was introduced by Girard [11] as a variant of game semantics with interactive types. Game Semantics has successfully provided fully abstract models for various logical systems and programming languages, among which PCF [12]. Although very close to Hyland–Ong (HO) games, ludics reverses the approach: in HO games one defines first the interpretation of a type (an arena) before giving the interpretation for the terms of that type (the strategies), while in ludics the interpretation of terms (the designs) is primitive and the types (the behaviours) are recovered dynamically as well-behaved sets of terms. This approach to types is similar to what exists in realisability [13] or geometry of interaction [10].

The motivation for such a framework was to reconstruct logic around the dynamics of proofs. Girard provides a ludics model for (a polarised version of) multiplicative-additive linear logic (MALL): a key role in his interpretation of logical connectives is played by the internal completeness results, which allow for a direct description of the behaviours’ content. As most behaviours are not the interpretation of MALL formulas, an interesting question, raised from the beginning of ludics, is whether these remaining behaviours can give a logical counterpart to computational phenomena. In particular, data and functions [17, 16], and also fixed points [2] have been studied in the setting of ludics. The present work follows this line of research.

Real life (functional) programs usually deal with data, functions over it, functions over functions, etc. Data types allow one to present information in a structured way. Some data types are defined inductively, for example:

Listing 1: Example of inductive types in OCaml

```ocaml
> type nat = Zero | Succ of nat ;;
> type 'a list = Nil | Cons of 'a * 'a list ;;
> type 'a tree = Empty | Node of 'a * ('a tree) list ;;
```

*Extended version of the paper accepted for publication in CSL 2017.
Upon this basis we can consider *functional types*, which are either first-order – from data to data – or higher-order – i.e., taking functions as arguments or returning functions as a result. This article aims at interpreting constructively the (potentially inductive) data types and the (potentially higher-order) functional types as behaviours of ludics, so as to study their structural properties. Inductive types are defined as (least) *fixed points*. As pointed out by Baelde, Doumane and Saurin [2], the fact that ludics puts the most constraints on the formation of terms instead of types, conversely to game semantics, makes it a more natural setting for the interpretation of fixed points than HO games [5].

**Contributions** The main contributions of this article are the following:

- We prove that internal completeness holds for infinite unions of behaviours satisfying particular conditions (Theorem 30), leading to an explicit construction of the least fixed points in ludics (Proposition 34).

- Inductive and functional types are interpreted as behaviours, and we prove that such behaviours are *regular* (Corollary 35 and Proposition 42). Regularity (that we discuss more in § 1.2) is a property that could be used to characterise the behaviours corresponding to \( \mu \text{MALL} \) formulas [1, 2] – i.e., MALL with fixed points.

- We show that a functional behaviour fails to satisfy *purity*, a property ensuring the safety of all possible executions (further explained in § 1.2), if and only if it is higher order and takes functions as argument (Proposition 43); this is typically the case of \((A \multimap B) \multimap C\). In § 5.2 we discuss the computational meaning of this result.

The present work is conducted in the term-calculus reformulation of ludics by Terui [17] restricted to the linear part – the idea is that programs call each argument at most once.

**Related Work** The starting point for our study of inductive types as fixed points in ludics is the work by Baelde, Doumane and Saurin [2]. In their article, they provide a ludics model for \( \mu \text{MALL} \), a variant of multiplicative-additive linear logic with least and greatest fixed points. The existence of fixed points in ludics is ensured by Knaster-Tarski theorem, but this approach does not provide an explicit way to construct the fixed points; we will consider Kleene fixed point theorem instead. Let us also mention the work of Melliès and Vouillon [14] which introduces a realisability model for recursive (i.e., inductive and coinductive) polymorphic types.

The representation of both data and functions in ludics has been studied previously. Terui [17] proposes to encode them as designs in order to express computability properties in ludics, but data and functions are not considered at the level of behaviours. Sironi [16] describes the behaviours corresponding to some data types: integers, lists, records, etc. as well as first-order function types; our approach generalises hers by considering generic data types and also higher order functions types.

**1.2 Background**

**Behaviours and Internal Completeness** A behaviour \( B \) is a set of designs which pass the same set of tests \( B^\perp \), where tests are also designs. \( B^\perp \) is called the *orthogonal* of \( B \), and behaviours are closed under bi-orthogonal: \( B^{\perp \perp} = B \). New behaviours can be formed upon others using various constructors. In this process, internal completeness, which can be seen as a built-in notion of observational equivalence, ensures that two agents reacting the same way to any test are actually equal. From a technical point of view, this means that it is not necessary to apply a \( \perp \perp \)-closure for the sets constructed to be behaviours.
Paths: Ludics as Game Semantics

This paper makes the most of the resemblance between ludics and HO game semantics. The connections between them have been investigated in many pieces of work [3, 7, 8] where designs are described as (innocent) strategies, i.e., in terms of the traces of their possible interactions. Following this idea, Fouqueré and Quatrini define paths [8], corresponding to legal plays in HO games, and they characterise a behaviour by its set of visitable paths. This is the approach we follow. The definitions of regularity and purity rely on paths, since they are properties of the possible interactions of a behaviour.

Regularity: Towards a Characterisation of $\mu$MALL?

Our proof that internal completeness holds for an infinite union of increasingly large behaviours (Theorem 30) relies in particular on the additional hypothesis of regularity for these behaviours. Intuitively, a behaviour $B$ is regular if every path in a design of $B$ is realised by interacting with a design of $B^\perp$, and vice versa. This property is not actually ad hoc: it was introduced by Fouqueré and Quatrini [9] to characterise the denotations of MALL formulas as being precisely the regular behaviours satisfying an additional finiteness condition. In this direction, our intuition is that – forgetting about finiteness – regularity captures the behaviours corresponding to formulas of $\mu$MALL. Although such a characterisation is not yet achieved, we provide a first step by showing that the data patterns, a subset of positive $\mu$MALL formulas, yield only regular behaviours (Proposition 33).

Purity: Type Safety

Ludics has a special feature for termination which is not present in game semantics: the daimon $\ordersymbol{\spadesuit}$. On a computational point of view, the daimon is commonly interpreted as an error, an exception raised at run-time causing the program to stop (see for example the notes of Curien [6]). Thinking of Ludics as a programming language, we would like to guarantee type safety, that is, ensure that “well typed programs cannot go wrong” [15]. This is the purpose of purity, a property of behaviours: in a pure behaviour, maximal interaction traces are $\ordersymbol{\spadesuit}$-free, in other words whenever the interaction stops with $\ordersymbol{\spadesuit}$ it is actually possible to “ask for more” and continue the computation. Introduced by Sironi [16] (and called principality in her work), this property is related to the notions of winning designs [11] and pure designs [17], but at the level of a behaviour. As expected, data types are pure (Corollary 40), but not always functional types are; we identify the precise cases where impurity arises (Proposition 43), and explain why some types are not safe.

1.3 Outline

In Section 2 we present ludics and we state internal completeness for the logical connectives constructions. In Section 3 we recall the notion of path, so as to define formally regularity and purity and prove their stability under the connectives. Section 4 studies inductive data types, which we interpret as behaviours; Kleene theorem and internal completeness for infinite union allows us to give an explicit and direct construction for the least fixed point, with no need for bi-orthogonal closure; we deduce that data types are regular and pure. Finally, in Section 5, we study functional types, showing in what case purity fails.

2 Computational Ludics

This section introduces the ludics background necessary for the rest of the paper, in the formalism of Terui [17]. The designs are the primary objects of ludics, corresponding to (polarised) proofs or programs in a Curry-Howard perspective. Cuts between designs can occur, and their reduction is called interaction. The behaviours, corresponding to the types or formulas of ludics, are then defined thanks to interaction. Compound behaviours can be formed with logical connectives constructions which satisfy internal completeness.
2.1 Designs and Interaction

Suppose given a set of variables \( V_0 \) and a set \( S \), called signature, equipped with an arity function \( ar : S \to \mathbb{N} \). Elements \( a, b, \ldots \in S \) are called names. A positive action is either \( \mathcal{X} \) (daimon), \( \Omega \) (divergence), or \( \overline{a} \) with \( a \in S \); a negative action is \( a(x_1, \ldots, x_n) \) where \( a \in S \), \( ar(a) = n \) and \( x_1, \ldots, x_n \in V_0 \) distinct. An action is proper if it is neither \( \mathcal{X} \) nor \( \Omega \).

Definition 1. Positive and negative designs\(^1\) are coinductively defined by:

\[
\begin{align*}
\mathcal{P} & ::= \mathcal{X} \mid \Omega \mid x\overline{\pi(n_1, \ldots, n_{ar(a)})} \mid n_0\overline{\pi(n_1, \ldots, n_{ar(a)})} \\
n & ::= \sum_{a \in S} a(x_1^a, \ldots, x_{ar(a)}^a) \mathcal{P}_a
\end{align*}
\]

Positive designs play the same role as applications in \( \lambda \)-calculus, and negative designs the role of abstractions, where each name \( a \in S \) binds \( ar(a) \) variables.

Designs are considered up to \( \alpha \)-equivalence. We will often write \( a(x^2) \) (resp. \( \overline{a}(\overline{n}^2) \)) instead of \( a(x_1, \ldots, x_n) \) (resp. \( \overline{a}(n_1, \ldots, n_n) \)). Negative designs can be written as partial sums, for example \( a(x, y), p + b() \cdot q \) instead of \( a(x, y) \cdot p + b() \cdot q + \sum_{c \in \mathcal{C}} c(x^2) \cdot \Omega \).

Given a design \( \mathcal{D} \), the definitions of the free variables of \( \mathcal{D} \), written \(fv(\mathcal{D})\), and the (capture-free) substitution of \( x \) by a negative design \( n \) in \( \mathcal{D} \), written \( \mathcal{D}[n/x] \), can easily be inferred. The design \( \mathcal{D} \) is closed if it is positive and it has no free variable. A subdesign of \( \mathcal{D} \) is a subterm of \( \mathcal{D} \). A cut in \( \mathcal{D} \) is a subdesign of \( \mathcal{D} \) of the form \( n_0\overline{\pi(n)} \), and a design is cut-free if it has no cut.

In the following, we distinguish a particular variable \( x_0 \), that cannot be bound. A positive design \( \mathcal{P} \) is atomic if \( fv(\mathcal{P}) \subseteq \{x_0\} \); a negative design \( n \) is atomic if \( fv(n) = \emptyset \).

A design is linear if for every subdesign of the form \( x\overline{\pi(n)} \) (resp. \( n_0\overline{\pi(n)} \)), the sets \( \{x\} \), \( fv(n_1) \), \( \ldots \), \( fv(n_{ar(a)}) \) (resp. the sets \( fv(n_0) \), \( fv(n_1) \), \( \ldots \), \( fv(n_{ar(a)}) \) are pairwise disjoint. This article focuses on linearity, so in the following when writing “design” we mean “linear design”.

Definition 2. The interaction corresponds to reduction steps applied on cuts:

\[
\sum_{a \in S} a(x_1^a, \ldots, x_{ar(a)}^a) \mathcal{P}_a \mid \overline{b}(n_1, \ldots, n_k) \leadsto \mathcal{P}_b[n_1/x_1^a, \ldots, n_k/x_k^a]
\]

We will later describe an interaction as a sequence of actions, a path (Definition 13).

Let \( \mathcal{P} \) be a design, and let \( \leadsto^* \) denote the reflexive transitive closure of \( \leadsto \); if there exists a design \( q \) which is neither a cut nor \( \Omega \) and such that \( \mathcal{P} \leadsto^* q \), we write \( \mathcal{P} \downarrow q \); otherwise we write \( \mathcal{P} \uparrow \). The normal form of a design, defined below, exists and is unique [17].

Definition 3. The normal form of a design \( \mathcal{D} \), noted \( (\mathcal{D}) \), is defined by:

\[
(\mathcal{P}) = \mathcal{X} \quad \text{if} \quad \mathcal{P} \downarrow \mathcal{X} \\
(\mathcal{P}) = \Omega \quad \text{if} \quad \mathcal{P} \uparrow \\
(\mathcal{P}) = \overline{\pi(n_1, \ldots, n_n)} \quad \text{if} \quad \mathcal{P} \downarrow x\overline{\pi(n_1, \ldots, n_n)}
\]

Note that the normal form of a closed design is either \( \mathcal{X} \) (convergence) or \( \Omega \) (divergence). Orthogonality expresses the convergence of the interaction between two atomic designs, and behaviours are sets of designs closed by bi-orthogonal.

Definition 4. Two atomic designs \( \mathcal{P} \) and \( n \) are orthogonal, noted \( \mathcal{P} \perp n \), if \( (\mathcal{P}[n/x_0]) = \mathcal{X} \).

Given an atomic design \( \mathcal{D} \), define \( \mathcal{D}^\perp = \{e \mid \mathcal{D} \perp e\} \); if \( E \) is a set of atomic designs of same polarity, define \( E^\perp = \{\mathcal{D} \mid \forall e \in E, \mathcal{D} \perp e\}\).

Definition 5. A set \( \mathcal{B} \) of atomic designs of same polarity is a behaviour\(^2\) if \( \mathcal{B}^\perp = \mathcal{B} \). A behaviour is either positive or negative depending on the polarity of its designs.

\(^1\)In the following, the symbols \( \mathcal{D}, e, \ldots \) refer to designs of any polarity, while \( p, q, \ldots \) and \( m, n, \ldots \) are specifically for positive and negative designs respectively.

\(^2\)Symbols \( A, B, \ldots \) will designate behaviours of any polarity, while \( M, N, \ldots \) and \( P, Q, \ldots \) will be for negative and positive behaviours respectively.
Behaviours could alternatively be defined as the orthogonal of a set $E$ of atomic designs of same polarity – $E$ corresponds to a set of tests or trials. Indeed, $E^ot$ is always a behaviour, and every behaviour $B$ is of this form by taking $E = B^ot$.

The incarnation of a behaviour $B$ contains the cut-free designs of $B$ whose actions are all visited during an interaction with a design in $B^ot$. Those correspond to the cut-free designs that are minimal for the stable ordering $\sqsubseteq$, where $\vartheta' \sqsubseteq \vartheta$ if $\vartheta$ can be obtained from $\vartheta'$ by substituting positive subdesigns for some occurrences of $\Omega$.

**Definition 6.** Let $B$ be a behaviour and $\vartheta \in B$ cut-free.

- The incarnation of $\vartheta$ in $B$, written $[\vartheta]_B$, is the smallest (for $\sqsubseteq$) cut-free design $\vartheta'$ such that $\vartheta' \sqsubseteq \vartheta$ and $\vartheta' \in B$. If $[\vartheta]_B = \vartheta$ we say that $\vartheta$ is incarnated in $B$.

- The incarnation $|B|$ of $B$ is the set of the (cut-free) incarnated designs of $B$.

### 2.2 Logical Connectives

Behaviour constructors – the **logical connectives** – can be applied so as to form compound behaviours. These connectives, coming from (polarised) linear logic, are used for interpreting formulas as behaviours, and will also indeed play the role of type constructors for the types of data and functions. In this subsection, after defining the connectives we consider, we state the internal completeness theorem for these connectives.

Let us introduce some notations. In the rest of this article, suppose the signature $S$ contains distinct unary names $\textsf{A}, \pi_1, \pi_2$ and a binary name $\varphi$, and write $\nabla = \textsf{A}, \iota_1 = \pi_1, \iota_2 = \pi_2$ and $\bullet = \varphi$. Given a behaviour $B$ and $x$ fresh, define $B^x = \{[\vartheta/x_0] \mid \vartheta \in B\}$; such a substitution operates a “delocation” with no repercussion on the behaviour’s inherent properties. Given a $k$-ary name $a \in S$, we write $\pi(N_1, \ldots, N_k)$ or even $\pi(N)$ for $\{x_0[\pi(N)] \mid n_i \in N_i\}$, and write $a(x^\pi).P$ for $\{a(x^\pi).p \mid p \in P\}$. For a negative design $n = \sum_{a \in S} a(x^\pi).p_a$ and a name $a \in S$, we denote by $n|a$ the design $a(x^\pi).p_a$ (that is $a(x^\pi).p_a + \sum_{b \neq a} b(x^\pi)\Omega$).

**Definition 7 (Logical connectives).**

\[
\begin{align*}
\downarrow N &= \nabla(N)^\bot & (\text{positive shift}) \\
\uparrow P &= (\textsf{A}(x).P^x)^\bot, \text{ with } x \text{ fresh} & (\text{negative shift}) \\
M \oplus N &= (\iota_1(M) \cup \iota_2(N))^\bot & (\text{plus}) \\
M \otimes N &= (M, N)^\bot & (\text{tensor}) \\
N \rightarrow P &= (N \otimes P^\bot)^\bot & (\text{linear map})
\end{align*}
\]

Our connectives $\downarrow, \uparrow, \oplus$ and $\otimes$ match exactly those defined by Terui [17], who also proves the following internal completeness theorem stating that connectives apply on behaviours in a constructive way – there is no need to close by bi-orthogonal. For each connective, we present two versions of internal completeness: one concerned with the full behaviour, the other with the behaviour’s incarnation.

**Theorem 8 (Internal completeness for connectives).**

\[
\begin{align*}
\downarrow N &= \nabla(N) \cup \{\textsf{X}\} & |\downarrow N| = \nabla(|N|) \cup \{\textsf{X}\} \\
\uparrow P &= \{n \mid n|\textsf{A} \in \textsf{A}(x).P^x\} & |\uparrow P| = \textsf{A}(x).|P^x| \\
M \oplus N &= \iota_1(M) \cup \iota_2(N) \cup \{\textsf{X}\} & |M \oplus N| = \iota_1(|M|) \cup \iota_2(|N|) \cup \{\textsf{X}\} \\
M \otimes N &= \bullet(M, N) \cup \{\textsf{X}\} & |M \otimes N| = \bullet(|M|, |N|) \cup \{\textsf{X}\}
\end{align*}
\]
3 Paths and Interactive Properties of Behaviours

Paths are sequences of actions recording the trace of a possible interaction. For a behaviour \( B \), we can consider the set of its visitable paths by gathering all the paths corresponding to an interaction between a design of \( B \) and a design of \( B^\perp \). This notion is needed for defining regularity and purity and proving that those two properties of behaviours are stable under (some) connectives constructions.

3.1 Paths

This subsection adapts the definitions of path and visitable path from [8] to the setting of computational ludics. In order to do so, we need first to recover location in actions so as to consider sequences of actions.

Location is a primitive idea in Girard’s ludics [11] in which the places of a design are identified with loci or addresses, but this concept is not visible in Terui’s presentation of designs-as-terms. We overcome this by introducing actions with more information on location, which we call located actions, and which are necessary to:

- represent cut-free designs as trees – actually, forests – in a satisfactory way,
- define views and paths.

Definition 9. A located action\(^3\) \( \kappa \) is one of: \( \kappa \) | \( x \) | \( a(x_1, \ldots, x_{\text{ar}(a)}) \) where in the last two cases (positive proper and negative proper respectively), \( a \in \mathcal{S} \) is the name of \( \kappa \), the variables \( x, x_1, \ldots, x_{\text{ar}(a)} \) are distinct, \( x \) is the address of \( \kappa \) and \( x_1, \ldots, x_{\text{ar}(a)} \) are the variables bound by \( \kappa \).

In the following, “action” will always refer to a located action. Similarly to notations for designs, \( x|\overline{a(x_1, \ldots, x_n)} \) stands for \( x|\overline{a(x_1, \ldots, x_n)} \) and \( a(x_1, \ldots, x_n) \) for \( a(x_1, \ldots, x_n) \).

Example 10. We show how cut-free designs can be represented as trees of located actions in this example. Let \( a^2, b^2, c^1, d^0 \in \mathcal{S} \), where exponents stand for arities. The following design is represented by the tree of Fig. 1.

\[
\delta = a(x_1, x_2). (x_2|\overline{a(x_3, x_4)}). x_2|\overline{b(z_1, z_2)} + c(y_1). (y_1|\overline{d()}). c(y_2). (x_1|\overline{d()})
\]

Such a representation is in general a forest: a negative design \( \sum_{a \in \mathcal{S}} a(x^3).p_a \) gives as many trees as there is \( a \in \mathcal{S} \) such that \( p_a \neq \Omega \). The distinguished variable \( x_0 \) is given as address to every negative root of a tree, and fresh variables are picked as addresses for negative actions bound by positive ones. This way, negative actions from the same subdesign, i.e., part of the same sum, are given the same address. A tree is indeed to be read bottom-up: a proper action \( \kappa \) is justified if

\(^3\)Located actions will often be denoted by symbol \( \kappa \), sometimes with its polarity: \( \kappa^+ \) or \( \kappa^- \).
its address is bound by an action of opposite polarity appearing below $\kappa$ in the tree; otherwise $\kappa$ is called initial. Except the root of a tree, which is always initial, every negative action is justified by the only positive action immediately below it. If $\kappa$ and $\kappa'$ are proper, $\kappa$ is hereditarily justified by $\kappa'$ if there exist actions $\kappa_1, \ldots, \kappa_n$ such that $\kappa = \kappa_1$, $\kappa' = \kappa_n$ and for all $i$ such that $1 \leq i < n$, $\kappa_i$ is justified by $\kappa_{i+1}$.

Before giving the definitions of view and path, let us give an intuition. On Fig. 1 are represented a view and a path of design $d$. Views are branches in the tree representing a cut-free design (reading bottom-up), while paths are particular “promenades” starting from the root of the tree; not all such promenades are paths, though. Views correspond to chronicles in original ludics [11].

For every positive proper action $\kappa^+ = x[\pi(\bar{y})]$ define $\kappa^+ = a_x(\bar{y})$, and similarly if $\kappa^- = a_x(\bar{y})$ define $\kappa^- = x[\pi(\bar{y})]$. Given a finite sequence of proper actions $s = \kappa_1 \ldots \kappa_n$, define $\bar{s} = \kappa_1 \ldots \kappa_n$.

Suppose now that if $s$ contains an occurrence of $\uplus$, it is necessarily in last position; the dual of $s$, written $\bar{s}$, is the sequence defined by:

- $\bar{s} = d\uplus$ if $s$ does not end with $\uplus$,
- $\bar{s} = \bar{s}' \uplus$ if $s = s' \uplus$.

Note that $\bar{\bar{s}} = s$. The notions of justified, hereditarily justified and initial actions also apply in sequences of actions.

**Definition 11.** An alternated justified sequence (or aj-sequence) $s$ is a finite sequence of actions such that:

- (Alternation) Polarities of actions alternate.
- (Daimon) If $\uplus$ appears, it is the last action of $s$.
- (Linearity) Each variable is the address of at most one action in $s$.

The (unique) justification of a justified action $\kappa$ in an aj-sequence is noted $\text{just}(\kappa)$, when there is no ambiguity on the sequence we consider.

**Definition 12.** A view $v$ is an aj-sequence such that each negative action which is not the first action of $v$ is justified by the immediate previous action. Given a cut-free design $d$, $v$ is a view of $d$ if it is a branch in the representation of $d$ as a tree (modulo $\alpha$-equivalence).

The way to extract the view of an aj-sequence is given inductively by:

- $[\epsilon] = \epsilon$ where $\epsilon$ is the empty sequence,
- $[sk^+] = [s][\kappa^+]$,
- $[sk^-] = [s_0][\kappa^-]$ where $s_0$ is the prefix of $s$ ending on $\text{just}(\kappa^-)$, or $s_0 = \epsilon$ if $\kappa^-$ initial.

The anti-view of an aj-sequence, noted $\uparrow s \downarrow$, is defined symmetrically by reversing the role played by polarities; equivalently $\uparrow s \downarrow = \bar{\bar{\downarrow s \uparrow}}$.

**Definition 13.** A path $s$ is a positive-ended aj-sequence satisfying:

- (P-visibility) For all prefix $s' \kappa^+$ of $s$, $\text{just}(\kappa^+) \in [s']^+$
- (O-visibility) For all prefix $s' \kappa^-$ of $s$, $\text{just}(\kappa^-) \in [s']^-$

Given a cut-free design $d$, a path $s$ is a path of $d$ if for all prefix $s'$ of $s$, $[s']^+$ is a view of $d$.7
Remark that the dual of a path is a path.

Paths are aimed at describing an interaction between designs. If \( \varnothing \) and \( \epsilon \) are cut-free atomic designs such that \( \varnothing \perp \epsilon \), there exists a unique path \( s \) of \( \varnothing \) such that \( \hat{s} \) is a path of \( \epsilon \). We write this path \( \langle \varnothing \leftarrow \epsilon \rangle \), and the good intuition is that it corresponds to the sequence of actions followed by the interaction between \( \varnothing \) and \( \epsilon \) on the side of \( \varnothing \). An alternative way defining orthogonality is then given by the following proposition.

**Proposition 14.** \( \varnothing \perp \epsilon \) if and only if there exists a path \( s \) of \( \varnothing \) such that \( \hat{s} \) is a path of \( \epsilon \).

At the level a behaviour \( B \), the set of visitable paths describes all the possible interactions between a design of \( B \) and a design of \( B^\perp \).

**Definition 15.** A path \( s \) is **visitable** in a behaviour \( B \) if there exist cut-free designs \( \varnothing \in B \) and \( \epsilon \in B^\perp \) such that \( s = \langle \varnothing \leftarrow \epsilon \rangle \). The set of visitable paths of \( B \) is written \( V_B \).

Note that for every behaviour \( B \), \( \tilde{V}_B = V_{B^\perp} \).

### 3.2 Regularity, Purity and Connectives

The meaning of regularity and purity has been discussed in the introduction. After giving the formal definitions, we prove that regularity is stable under all the connectives constructions. We also show that purity may fail with \( -\circ \), and only a weaker form called **quasi-purity** is always preserved.

**Definition 16.** \( B \) is **regular** if the following conditions are satisfied:

- for all \( \varnothing \in |B| \) and all path \( s \) of \( \varnothing \), \( s \in V_B \),
- for all \( \varnothing \in |B^\perp| \) and all path \( s \) of \( \varnothing \), \( s \in V_{B^\perp} \),
- The sets \( V_B \) and \( V_{B^\perp} \) are stable under shuffle.

where the operation of **shuffle** \((\shuffle)\) on paths corresponds to an interleaving of actions respecting alternation of polarities, and is defined below.

Let \( s \shuffle s' \) refer to the subsequence of \( s \) containing only the actions that occur in \( s' \). Let \( s \) and \( t \) be paths of same polarity, let \( S \) and \( T \) be sets of paths of same polarity. We define:

- \( s \shuffle t = \{ u \text{ path formed with actions from } s \text{ and } t \mid u|s = s \text{ and } u|t = t \} \) if \( s, t \) negative,
- \( s \shuffle t = \{ \kappa^+ u \text{ path } \mid u \in s' \shuffle t' \} \) if \( s = \kappa^+ s' \) and \( t = \kappa^+ t' \) positive with same first action,
- \( S \shuffle T = \{ u \text{ path } \mid \exists s \in S, \exists t \in T \text{ such that } s \shuffle t \text{ is defined and } u \in s \shuffle t \} \).

In fact, a behaviour \( B \) is regular if every path formed with actions of the incarnation of \( B \), even mixed up, is a visitable path of \( B \), and similarly for \( B^\perp \). Remark that regularity is a property of both a behaviour and its orthogonal since the definition is symmetrical: \( B \) is regular if and only if \( B^\perp \) is regular.

**Definition 17.** A behaviour \( B \) is **pure** if every \( \triangle \)-ended path \( s\triangle \in V_B \) is **extensible**, i.e., there exists a proper positive action \( \kappa^+ \) such that \( s\kappa^+ \in V_B \).

Purity ensures that when an interaction encounters \( \triangle \), this does not correspond to a real error but rather to a partial computation, as it is possible to continue this interaction. Note that daemons are necessarily present in all behaviours since the converse property is always true: if \( s\kappa^+ \in V_B \) then \( s\triangle \in V_B \).

**Proposition 18.** Regularity is stable under \( \downarrow, \uparrow, \oplus, \otimes \) and \( -\circ \).
Proposition 19. Purity is stable under $\downarrow$, $\uparrow$, $\oplus$ and $\otimes$.

Unfortunately, when $N$ and $P$ are pure, $N \rightarrow P$ is not necessarily pure, even under regularity assumption. However, a weaker form of purity holds for $N \rightarrow P$.

Definition 20. A behaviour $B$ is quasi-pure if all the $\vec{C}$-ended well-bracketed paths in $V_B$ are extensible.

We recall that a path $s$ is well-bracketed if, for every justified action $\kappa$ in $s$, when we write $s = s_0 \kappa' s_1 \kappa s_2$ where $\kappa'$ justifies $\kappa$, all the actions in $s_1$ are hereditarily justified by $\kappa'$.

Proposition 21. If $N$ and $P$ are quasi-pure and regular then $N \rightarrow P$ is quasi-pure.

4 Inductive Data Types

Some important contributions are presented in this section. We interpret inductive data types as positive behaviours, and we prove an internal completeness result allowing us to make explicit the structure of fixed points. Regularity and purity of data follows.

Abusively, we denote the positive behaviour $\{\vec{C}\}$ by $\vec{C}$ all along this section.

4.1 Inductive Data Types as Kleene Fixed Points

We define the data patterns via a type language and interpret them as behaviours, in particular $\mu$ is interpreted as a least fixed point. Data behaviours are the interpretation of steady data patterns.

Suppose given a countably infinite set $V$ of second-order variables: $X, Y, \cdots \in V$. Let $S' = S \setminus \{\alpha, \pi_1, \pi_2, \varphi\}$ and define the set of constants $\text{Const} = \{C_a \mid a \in S'\}$ which contains a behaviour $C_a = \{x_0[\overrightarrow{\alpha}(\overrightarrow{x})] \}^{1,1}$ (where $\overrightarrow{x} := \sum_{a \in S} a(x).\overrightarrow{\alpha}$) for each $a \in S'$, i.e., such that $a$ is not the name of a connective. Remark that $V_{C_a} = (\vec{C}, x_0[\overrightarrow{\alpha}(\overrightarrow{x})])$, thus $C_a$ is regular and pure.

Definition 22. The set $P$ of data patterns is generated by the inductive grammar:

$$A, B ::= X \in V \mid a \in S' \mid A \oplus B \mid A \otimes B \mid \mu X.A$$

The set of free variables of a data pattern $A \in P$ is denoted by $\text{FV}(A)$.

Example 23. Let $b, n, l, t \in S'$ and $X \in V$. The data types given as example in the introduction can be written in the language of data patterns as follows:

$$\text{Bool} = b \oplus^+ b \quad \text{Nat} = \mu X. (n \oplus^+ X) \quad \text{List}_A = \mu X. (l \oplus^+ (A \oplus^+ X))$$

$$\text{Tree}_A = \mu X. (t \oplus^+ (A \oplus^+ \text{List}_X)) = \mu X. (t \oplus^+ (A \oplus^+ \mu Y.(l \oplus^+ (X \oplus^+ Y))))$$

Let $B^+$ be the set of positive behaviours. Given a data pattern $A \in P$ and an environment $\sigma$, i.e., a function that maps free variables to positive behaviours, the interpretation of $A$ in the environment $\sigma$, written $[[A]]^\sigma$, is the positive behaviour defined by:

$$[[X]]^\sigma = \sigma(X) \quad [A \oplus^+ B]^\sigma = (\uparrow [A]^\sigma) \oplus (\uparrow [B]^\sigma)$$

$$[[a]]^\sigma = C_a \quad [A \otimes^+ B]^\sigma = (\uparrow [A]^\sigma) \otimes (\uparrow [B]^\sigma)$$

$$[[\mu X.A]]^\sigma = \text{lfp}(\phi^A_\sigma)$$

where lfp stands for the least fixed point, and the function $\phi^A_\sigma : B^+ \rightarrow B^+, P \mapsto [[A]]^\sigma : X \mapsto P$ is well defined and has a least fixed point by Knaster-Tarski fixed point theorem, as shown by Baelde, Doumane and Saurin [2]. Abusively we may write $\oplus^+$ and $\otimes^+$, instead of $(\uparrow \cdot) \oplus (\uparrow \cdot)$ and $(\uparrow \cdot) \otimes (\uparrow \cdot)$ respectively, for behaviours. We call an environment $\sigma$ regular (resp. pure) if its image contains
only regular (resp. pure) behaviours. The notation \( \sigma, X \mapsto \mathbf{P} \) stands for the environment \( \sigma \) where the image of \( X \) has been changed to \( \mathbf{P} \).

In order to understand the structure of fixed point behaviours that interpret the data patterns of the form \( \mu X.A \), we need a constructive approach, thus Kleene fixed point theorem is best suited than Knaster-Tarski. We now prove that we can apply this theorem.

Recall the following definitions and theorem. A partial order is a **complete partial order** (CPO) if each directed subset has a supremum, and there exists a smallest element, written \( \bot \). A function \( f : E \rightarrow F \) between two CPOs is **Scott-continuous** (or simply continuous) if for every directed subset \( D \subseteq E \) we have \( \bigvee_{x \in D} f(x) = f(\bigvee_{x \in D} x) \).

**Theorem 24** (Kleene fixed point theorem). Let \( L \) be a CPO and let \( f : L \rightarrow L \) be Scott-continuous. The function \( f \) has a least fixed point, defined by

\[
\text{lfp}(f) = \bigvee_{n \in \mathbb{N}} f^n(\bot)
\]

The set \( B^+ \) ordered by \( \subseteq \) is a CPO, with least element \( \emptyset \); indeed, given a subset \( \mathbb{P} \subseteq B^+ \), it is directed and we have \( \bigvee \mathbb{P} = (\bigcup \mathbb{P})^{\perp \perp} \). Hence next proposition proves that we can apply the theorem.

**Proposition 25.** Given a data pattern \( A \in \mathcal{P} \), a variable \( X \in \mathcal{V} \) and an environment \( \sigma : \text{FV}(A) \setminus \{X\} \rightarrow B^+ \), the function \( \phi^A_\sigma \) is Scott-continuous.

**Corollary 26.** For every \( A \in \mathcal{P} \), \( X \in \mathcal{V} \) and \( \sigma : \text{FV}(A) \setminus \{X\} \rightarrow B^+ \),

\[
[\mu X.A]^{\sigma} = \bigvee_{n \in \mathbb{N}} (\phi^A_\sigma)^n(\emptyset) = (\bigcup_{n \in \mathbb{N}} (\phi^A_\sigma)^n(\emptyset))^{\perp \perp}
\]

This result gives an explicit formulation for least fixed points. However, the \( \perp \perp \)-closure might add new designs which were not in the union, making it difficult to know the exact content of such a behaviour. The point of next subsection will be to give an internal completeness result proving that the closure is actually not necessary.

Let us finish this subsection by defining a restricted set of data patterns so as to exclude the degenerate ones. Consider for example \( \text{List}_{\mathcal{A}}^\prime = \mu X.(A \circ^+ X) \), a variant of \( \text{List}_{\mathcal{A}} \) (see Example 23) which misses the base case. It is degenerate in the sense that the base element, here the empty list, is interpreted as the design \( \emptyset \). This is problematic: an interaction going through a whole list will end with an error, making it impossible to explore a pair of lists for example. The pattern \( \text{Nat}^\prime = \mu X.X \) is even worse since \([\text{Nat}^\prime] = \emptyset \). The point of steady data patterns is to ensure the existence of a basis; this will be formalised in Lemma 37.

**Definition 27.** The set of **steady** data patterns is the smallest subset \( \mathcal{P}^s \subseteq \mathcal{P} \) such that:

- \( S^' \subseteq \mathcal{P}^s \)
- If \( A \in \mathcal{P}^s \) and \( B \) is such that \([B]^\sigma \) is pure if \( \sigma \) is pure, then \( A \circ^+ B \in \mathcal{P}^s \) and \( B \circ^+ A \in \mathcal{P}^s \)
- If \( A \in \mathcal{P}^s \) and \( B \in \mathcal{P}^s \) then \( A \circ^+ B \in \mathcal{P}^s \)
- If \( A \in \mathcal{P}^s \) then \( \mu X.A \in \mathcal{P}^s \)

The condition on \( B \) in the case of \( \circ^+ \) admits data patterns which are not steady, possibly with free variables, but ensuring the preservation of purity, i.e., type safety; the basis will come from side \( A \). We will prove (§ 4.3) that behaviours interpreting steady data patterns are pure, thus in particular a data pattern of the form \( \mu X.A \) is steady if the free variables of \( A \) all appear on the same side of a \( \circ^+ \) and under the scope of no other \( \mu \) (since purity is stable under \( \perp, \perp, \circ, \circ \)). We claim that steady data patterns can represent every type of finite data.

**Definition 28.** A **data behaviour** is the interpretation of a closed steady data pattern.
4.2 Internal Completeness for Infinite Union

Our main result is an internal completeness theorem, stating that an infinite union of simple regular behaviours with increasingly large incarnations is a behaviour: \( \bot \bot \)-closure is useless.

**Definition 29.** • A slice is a design in which all negative subdesigns are either \( \Omega^- \) or of the form \( a(\overline{Z})p_n \), i.e., at most unary branching. \( \epsilon \) is a slice of \( \delta \) if \( \epsilon \) is a slice and \( \epsilon \subseteq \delta \). A slice \( \epsilon \) of \( \delta \) is maximal if for any slice \( \epsilon' \) of \( \delta \) such that \( \epsilon \subseteq \epsilon' \), we have \( \epsilon = \epsilon' \).

• A behaviour \( B \) is simple if for every design \( \delta \in [B] \):
  1. \( \delta \) has a finite number of maximal slices, and
  2. every positive action of \( \delta \) is justified by the immediate previous negative action.

Condition (2) of simplicity ensures that, given \( \delta \in [B] \) and a slice \( \epsilon \subseteq \delta \), one can find a path of \( \epsilon \) containing all the positive proper actions of \( \epsilon \) until a given depth; thus by condition (1), there exists \( k \in \mathbb{N} \) depending only on \( \delta \) such that \( k \) paths can do the same in \( \delta \).

Now suppose \( \left( A_n \right)_{n \in \mathbb{N}} \) is an infinite sequence of simple regular behaviours such that for all \( n \in \mathbb{N} \), \( |A_n| \subseteq |A_{n+1}| \) (in particular we have \( A_n \subseteq A_{n+1} \)).

**Theorem 30.** The set \( \bigcup_{n \in \mathbb{N}} A_n \) is a behaviour.

A union of behaviours is not a behaviour in general. In particular, counterexamples are easily found if releasing either the inclusion of incarnations or the simplicity condition. Moreover, our proof for this theorem relies strongly on regularity. Under the same hypotheses we can prove \( V_{\bigcup_{n \in \mathbb{N}} A_n} = \bigcup_{n \in \mathbb{N}} V_{A_n} \) and \( |\bigcup_{n \in \mathbb{N}} A_n| = \bigcup_{n \in \mathbb{N}} |A_n| \), hence the following corollary.

**Corollary 31.** • \( \bigcup_{n \in \mathbb{N}} A_n \) is simple and regular;
  • if moreover all the \( A_n \) are pure then \( \bigcup_{n \in \mathbb{N}} A_n \) is pure.

4.3 Regularity and Purity of Data

The goal of this subsection is to show that the interpretation of data patterns of the form \( \mu X.A \) can be expressed as an infinite union of behaviours \( \left( A_n \right)_{n \in \mathbb{N}} \) satisfying the hypotheses of Theorem 30, in order to deduce regularity and purity. We will call an environment \( \sigma \) simple if its image contains only simple behaviours.

**Lemma 32.** For all \( A \in \mathcal{P} \), \( X \in \mathcal{V} \), \( \sigma : \text{FV}(A) \setminus \{X\} \rightarrow \mathcal{B}^+ \) simple and regular\(^4\), and \( n \in \mathbb{N} \) we have
\[
\left| \left( \phi^\sigma_A \right)^n(\overline{X}) \right| \subseteq \left| \left( \phi^\sigma_A \right)^{n+1}(\overline{X}) \right|
\]

**Proposition 33.** For all \( A \in \mathcal{P} \) and simple regular environment \( \sigma \), \( [A]^{\sigma} \) is simple regular.

**Proof.** By induction on data patterns. If \( A = X \) or \( A = a \) the conclusion is immediate. If \( A = A_1 \odot^+ A_2 \) or \( A = A_1 \odot A_2 \) then regularity comes from Proposition 18, and simplicity is easy since the structure of the designs in \( [A]^{\sigma} \) is given by internal completeness for the logical connectives (Theorem 8). So suppose \( A = \mu X.A_0 \). By induction hypothesis, for every simple regular behaviour \( P \in \mathcal{B}^+ \) we have \( \phi^\sigma_P(P) = \left[ [A_0]^{\sigma,X \mapsto P} \right]^{\sigma} \) simple regular. From this, it is straightforward to show by induction that for every \( n \in \mathbb{N} \), \( \left( \phi^\sigma_A \right)^n(\overline{X}) \) is simple regular. Moreover, for every \( n \in \mathbb{N} \) we have \( \left| \left( \phi^\sigma_A \right)^n(\overline{X}) \right| \subseteq \left| \left( \phi^\sigma_A \right)^{n+1}(\overline{X}) \right| \) by Lemma 32, thus by Corollary 26 and Theorem 30, \( \left[ \mu X.A_0 \right]^{\sigma} = \bigvee_{n \in \mathbb{N}} \left( \phi^\sigma_A \right)^n(\overline{X}) = \left( \bigcup_{n \in \mathbb{N}} \left( \phi^\sigma_A \right)^n(\overline{X}) \right)^{\perp \perp} = \bigcup_{n \in \mathbb{N}} \left( \phi^\sigma_A \right)^n(\overline{X}) \). Consequently, by Corollary 31, \( \left[ \mu X.A_0 \right]^{\sigma} \) is simple regular. \( \square \)

Remark that we have proved at the same time, using Theorem 30, that behaviours interpreting data patterns \( \mu X.A \) admit an explicit construction:

\(^4\)The hypothesis “simple and regular” has been added, compared to the CSL version of this article, for correction.
Proposition 34. If \( A \in \mathcal{P}, X \in \mathcal{V}, \) and \( \sigma : \text{FV}(A) \setminus X \rightarrow B^+ \) is simple regular,
\[
[\mu X . A]^{\sigma} = \bigcup_{n \in \mathbb{N}} (\phi_A^X)^n(\mathcal{X})
\]

Corollary 35. Data behaviours are regular.

We now move on to proving purity. The proof that the interpretation of a steady data pattern \( A \) is pure relies on the existence of a basis for \( A \) (Lemma 37). Let us first widen (to \( \mathcal{X} \)-free paths) and express in a different way (for \( \mathcal{X} \)-ended paths) the notion of extensible visitable path.

Definition 36. Let \( B \) be a behaviour.

- A \( \mathcal{X} \)-free path \( s \in V_B \) is **extensible** if there exists \( t \in V_B \) of which \( s \) is a strict prefix.
- A \( \mathcal{X} \)-ended path \( s \in V_B \) is **extensible** if there exists a positive action \( \kappa^+ \) and \( t \in V_B \) of which \( s\kappa^+ \) is a prefix.

Write \( V_B^{\max} \) for the set of maximal, i.e., non extensible, visitable paths of \( B \).

Lemma 37. Every steady data pattern \( A \in \mathcal{P}^s \) has a basis, i.e., a simple regular behaviour \( B \) such that for all simple regular environment \( \sigma \) we have

- \( B \subseteq [A]^{\sigma} \),
- for every path \( s \in V_B \), there exists \( t \in V_B^{\max} \) \( \mathcal{X} \)-free extending \( s \) (in particular \( B \) pure),
- \( V_B^{\max} \subseteq [A]^{\sigma} \).

Proof (idea). If \( A = a \), a basis is \( C_a \). If \( A = A_1 \oplus^+ A_2 \), and \( A_i \) is steady with basis \( B_i \), then \( \otimes_i^1 B_i := \iota_i(\uparrow B_i) \) is a basis for \( A \). If \( A = A_1 \oplus^+ A_2 \), a basis is \( B_1 \oplus^+ B_2 \) where \( B_1 \) and \( B_2 \) are basis of \( A_1 \) and \( A_2 \) respectively. If \( A = \mu X . A_0 \), its basis is the same as \( A_0 \).

Proposition 38. If \( A \in \mathcal{P}^s \) of basis \( B, X \in \mathcal{V}, \) and \( \sigma : \text{FV}(A) \setminus X \rightarrow B^+ \) simple regular,
\[
[\mu X . A]^{\sigma} = \bigcup_{n \in \mathbb{N}} (\phi_A^X)^n(\mathcal{B})
\]

Proof. Since \( B \) is a basis for \( A \) we have \( \mathcal{X} \subset B \subseteq [A]^{\sigma,X \rightarrow \mathcal{X}} = \phi_A^X(\mathcal{X}) \). The Scott-continuity of the function \( \phi_A^X \) implies that it is increasing, thus \( (\phi_A^X)^n(\mathcal{X}) \subseteq (\phi_A^X)^n(B) \subseteq (\phi_A^X)^{n+1}(\mathcal{X}) \) for all \( n \in \mathbb{N} \). Hence \( [A]^{\sigma} = \bigcup_{n \in \mathbb{N}} (\phi_A^X)^n(\mathcal{X}) = \bigcup_{n \in \mathbb{N}} (\phi_A^X)^n(B) \).

Proposition 39. For all \( A \in \mathcal{P}^s \) and simple regular pure environment \( \sigma \), \([A]^{\sigma}\) is pure.

Proof. By induction on \( A \). The base cases are immediate and the connective cases are solved using Proposition 19. Suppose now \( A = \mu X . A_0 \), where \( A_0 \) is steady with basis \( B_0 \). We have \([A]^{\sigma} = \bigcup_{\sigma \in \mathcal{N}} (\phi_{\sigma}^{A_0})^{n}(B_0)\) by Proposition 38, let us prove it satisfies the hypotheses needed to apply Corollary 31(2). By induction hypothesis and Proposition 33, for every simple, regular and pure behaviour \( P \in B^+ \) we have \( \phi_{\sigma}^{A_0}(P) = [A_0]^{\sigma,X \rightarrow P} \) simple, regular and pure, hence it is easy to show by induction that for every \( n \in \mathbb{N} \), \((\phi_{\sigma}^{A_0})^{n}(B_0)\) is as well. Moreover, for every \( n \in \mathbb{N} \) we prove that \([(\phi_{\sigma}^{A_0})^{n}(B_0)] \subseteq [(\phi_{\sigma}^{A_0})^{n+1}(B_0)] \) similarly to Lemma 32, replacing \( \mathcal{X} \) by the basis \( B_0 \). Finally, by Corollary 31, \([A]^{\sigma}\) is pure.

Corollary 40. Data behaviours are pure.
Remark. Although here the focus is on the interpretation of data patterns, we should say a word about the interpretation of (polarised) µMALL formulas, which are a bit more general. These formulas are generated by:

\[
P, Q ::= X_P \mid X_N \mid 1 \mid 0 \mid M \oplus N \mid M \otimes N \mid N \mid \mu X.P
\]

\[
M, N ::= P \perp
\]

where the usual involutive negation hides the negative connectives and constants, through the dualities \(1/\perp\), \(0/\top\), \(\oplus/\&\), \(\otimes/\`\), \(\`/\hat{\top}\), \(\mu/\nu\). The interpretation as ludics behaviours, given in [2], is as follows: \(1\) is interpreted as a constant behaviour \(C_a\), \(0\) is the daimon \(✠\), the positive connectives match their ludics counterparts, \(\mu\) is interpreted as the least fixed point of a function \(\phi^+\) similarly to data patterns, and the negation corresponds to the orthogonal. Since in ludics constants and \(✠\) are regular, and since regularity is preserved by the connectives (Proposition 18) and by orthogonality, the only thing we need in order to prove that all the behaviours interpreting µMALL formulas are regular is a generalisation of regularity stability under fixed points (for now we only have it in our particular case: Corollary 31 together with Proposition 34).

Note however that interpretations of µMALL formulas are not all pure. Indeed, as we will see in next section, orthogonality (introduced through the connective \(\rightarrow\)) does not preserve purity in general.

5 Functional Types

In this section we define functional behaviours which combine data behaviours with the connective \(\rightarrow\). A behaviour of the form \(N \rightarrow P\) is the set of designs such that, when interacting with a design of type \(N\), outputs a design of type \(P\); this is exactly the meaning of its definition \(N \rightarrow P := (N \otimes P)\perp\). We prove that some particular higher-order functional types – where functions are taken as arguments, typically \((A \rightarrow B) \rightarrow C\) – are exactly those who fail at being pure, and we interpret this result from a computational point of view.

5.1 Where Impurity Arises

We have proved that data behaviours are regular and pure. However, if we introduce functional behaviours with the connective \(\rightarrow\), purity does not hold in general. Proposition 42 indicates that a weaker property, quasi-purity, holds for functional types, and Proposition 43 identifies exactly the cases where purity fails.

Let us write \(D\) for the set of data behaviours.

Definition 41. A functional behaviour is a behaviour inductively generated by the grammar below, where \(P \rightarrow^+ Q\) stands for \(\downarrow((\uparrow P) \rightarrow Q)\).

\[
P, Q ::= P_0 \in D \mid P \oplus^+ Q \mid P \otimes^+ Q \mid P \rightarrow^+ Q
\]

From Propositions 18, 19 and 21 we easily deduce the following result.

Proposition 42. Functional behaviours are regular and quasi-pure.

For next proposition, consider contexts defined inductively as follows (where \(P\) is a functional behaviour):

\[
C ::= \[ \] \mid C \oplus^+ P \mid P \oplus^+ C \mid C \otimes^+ P \mid P \otimes^+ C \mid P \rightarrow^+ C
\]

Proposition 43. A functional behaviour \(P\) is impure if and only if there exist contexts \(C_1, C_2\) and functional behaviours \(Q_1, Q_2, R\) with \(R \notin \text{Const}\) such that

\[
P = C_1[ C_2[Q_1 \rightarrow^+ Q_2] \rightarrow^+ R ]
\]
5.2 Example and Discussion

Proposition 43 states that a functional behaviour which takes functions as argument is not pure: some of its visitable paths end with a daimon $\star$, and there is no possibility to extend them. In terms of proof-search, playing the daimon is like giving up; on a computational point of view, the daimon appearing at the end of an interaction expresses the sudden interruption of the computation. In order to understand why such an interruption can occur in the specific case of higher-order functions, consider the following example which illustrates the proposition.

Example 44. Let $Q_1, Q_2, 1$ be functional behaviours, with $1 \in \text{Const}$. Define $\text{Bool} = 1 \oplus 1$ and consider the behaviour $P = (Q_1 \mapsto Q_2) \mapsto \text{Bool}$: this is a type of functions which take a function as argument and output a boolean. Let $\alpha_1, \alpha_2, \beta$ be respectively the first positive action of the designs of $Q_1, Q_2, 1$. It is possible to exhibit a design $p \in P$ and a design $n \in P^\perp$ such that the visitable path $s = (p \leftarrow n)$ is $\star$-ended and maximal in $V_P$, in other words $s$ is a witness of the impurity of $P$. The path $s$ contains the actions $\alpha_1$ and $\alpha_2$ in such a way that it cannot be extended with $\beta$ without breaking the $P$-visibility condition, and there is no other available action in designs of $P$ to extend it. Reproducing the designs $p$ and $n$ and the path $s$ here would be of little interest since those objects are too large to be easily readable ($s$ visits the entire design $p$, which contains 11 actions). We however give an intuition in the style of game semantics: Fig. 2 represents $s$ as a legal play in a strategy of type $P = (Q_1 \mapsto Q_2) \mapsto \text{Bool}$ (note that only one “side” $\oplus 1 \uparrow 1$ of $\text{Bool}$ is represented, corresponding for example to $\text{True}$, because we cannot play in both sides). This analogy is informal, it should stand as an intuition rather than as a precise correspondence with ludics; for instance, and contrary to the way it is presented in game semantics, the questions are asked on the connectives, while the answers are given in the sub-types of $P$. On the right are given the actions in $s$ corresponding to the moves played. The important thing to remark is the following: if a move $b$ corresponding to action $\beta$ were played instead of $\star$ at the end of this play, it would break the $P$-visibility of the strategy, since this move would be justified by move $q_1$.

The computational interpretation of the $\star$-ended interaction between $p$ and $n$ is the following: a program $p$ of type $P$ launches a child process $p'$ to compute the argument of type $Q_1 \rightarrow Q_2$, but $p$ starts to give a result in $\text{Bool}$ before the execution of $p'$ terminates, leading to a situation where $p$ cannot compute the whole data in $\text{Bool}$. The interaction outputs $\star$, i.e., the answer given in

![ Figure 2: Representation of path $s$ from Example 44 in the style of a legal play](image)
**Bool** by $p$ is incomplete.

Moreover by Proposition 42 functional behaviours are quasi-pure, therefore the maximal $\star$-ended visitable paths are necessarily not well-bracketed. This is indeed the case of $s$: remark for example that the move $q_{\oplus 1}$ appears between $a_1$ and its justification $q_2$ in the sequence, but $q_{\oplus 1}$ is not hereditarily justified by $q_2$. In HO games, well-bracketedness is a well studied notion, and relaxing it introduces control operators in program. If we extend such an argument to ludics, this would mean that the appearance of $\star$ in the execution of higher-order functions can only happen in the case of programs with control operators such as *jumps*, i.e. programs which are not purely functional.

### 6 Conclusion

This article is a contribution to the exploration of the behaviours of linear ludics in a computational perspective. Our focus is on the behaviours representing data types and functional types. Inductive data types are interpreted using the logical connectives constructions and a least fixed point operation. Adopting a constructive approach, we provide an internal completeness result for fixed points, which unveils the structure of data behaviours. This leads us to proving that such behaviours are regular – the key notion for the characterisation of MALL in ludics – and pure – that is, type safe. But behaviours interpreting types of functions taking functions as argument are impure; for well-bracketed interactions, corresponding to the evaluation of purely functional programs, safety is however guaranteed.

**Further Work** Two directions for future research arise naturally:

- Extending our study to greatest fixed points $\nu X.A$, i.e., coinduction, is the next objective. Knaster–Tarski ensures that such greatest fixed point behaviours exist [2], but Kleene fixed point theorem does not apply here, hence we cannot find an explicit form for coinductive behaviours the same way we did for the inductive ones. However it is intuitively clear that, compared to least fixed points, greatest ones add the infinite “limit” designs in (the incarnation of) behaviours. For example, if $\text{Nat}_\omega = \nu X. (1 \oplus X)$ then we should have $\llbracket \text{Nat}_\omega \rrbracket = \llbracket \text{Nat} \rrbracket \cup \{\alpha_\omega\}$ where $\alpha_\omega = \text{succ}(\alpha_\omega) = x_0|\iota_2(\hat{x}.\alpha_\omega^x)$.

- Another direction would be to get a complete characterisation of $\mu$MALL in ludics, by proving that a behaviour is regular – and possibly satisfying a supplementary condition – if and only if it is the denotation of a $\mu$MALL formula.

**Acknowledgement** I thank Claudia Faggian, Christophe Fouqueré, Thomas Seiller and the anonymous referees for their wise and helpful comments.

### References

[1] David Baelde. Least and greatest fixed points in linear logic. *ACM Trans. Comput. Logic*, 13(1):2:1–2:44, January 2012. URL: http://doi.acm.org/10.1145/2071368.2071370. doi: 10.1145/2071368.2071370.

[2] David Baelde, Amina Doumane, and Alexis Saurin. Least and greatest fixed points in ludics. In Stephan Kreuzer, editor, Proceedings of the 24th Annual EACSL Conference on Computer Science Logic (CSL’15), pages 549–566, Berlin, Germany, September 2015. URL: http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/BS-csl15.pdf, doi:10.4230/LIPIcs.CSL.2015.549.

[3] Michele Basaldella and Claudia Faggian. Ludics with repetitions (exponentials, interactive types and completeness). *Logical Methods in Computer Science*, 7(2), 2011.
[4] Michele Basaldella and Kazushige Terui. On the meaning of logical completeness. *Logical Methods in Computer Science*, 6(4), 2010.

[5] Pierre Clairambault. Least and greatest fixpoints in game semantics. In *Foundations of Software Science and Computational Structures, 12th International Conference (FOSSACS 2009). Proceedings*, pages 16–31, 2009.

[6] Pierre-Louis Curien. Introduction to linear logic and ludics, part II. *CoRR*, abs/cs/0501039, 2005.

[7] Claudia Faggian and Martin Hyland. Designs, disputes and strategies. In *CSL*, pages 442–457, 2002.

[8] Christophe Fouqueré and Myriam Quatrini. Incarnation in ludics and maximal cliques of paths. *Logical Methods in Computer Sciences*, 9(4), 2013.

[9] Christophe Fouqueré and Myriam Quatrini. Study of behaviours via visitable paths. *CoRR*, abs/1403.3772v3, 2016. URL: https://arxiv.org/abs/1403.3772v3.

[10] Jean-Yves Girard. Geometry of interaction. I. Interpretation of system F. In *Proc. Logic Colloquium ’88*, pages 221–260, North-Holland, Amsterdam, 1989.

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

[12] Martin Hyland and Luke Ong. On full abstraction for PCF: I, II, and III. *Information and Computation*, 163(2):285–408, 2000.

[13] Jean-Louis Krivine. Realizability in classical logic. *Panoramas et synthèses*, 27:197–229, 2009. URL: https://hal.archives-ouvertes.fr/hal-00154500.

[14] Paul-André Melliès and Jerome Vouillon. Recursive polymorphic types and parametricity in an operational framework. In *20th IEEE Symposium on Logic in Computer Science (LICS 2005)*, 26-29 June 2005, Chicago, IL, USA, *Proceedings*, pages 82–91, 2005. URL: http://dx.doi.org/10.1109/LICS.2005.42, doi:10.1109/LICS.2005.42.

[15] Robin Milner. A theory of type polymorphism in programming. *Journal of Computer and System Sciences*, 17:348–375, 1978.

[16] Eugenia Sironi. *Types in Ludics*. PhD thesis, Aix-Marseille Université, January 2015.

[17] Kazushige Terui. Computational ludics. *Theor. Comput. Sci.*, 412(20):2048–2071, 2011.
In the appendix, we adopt Barendregt’s variable convention; that is, among objects in a given context, we will always assume that:

1. no variable appears both free and bound, and
2. bound variables have all distinct names.

This affects designs, multi-designs, representations of designs as trees, and paths.

## A Proof of Proposition 14

The purpose of this section is to lift the framework to *multi-designs*, in order to prove properties of the path recording the interaction between multi-designs (thus in particular, between designs). We show:

- the existence and uniqueness of the interaction path between two orthogonal multi-designs (Proposition 63),
- the equivalence between the existence of such a path and the orthogonality of two multi-designs (Proposition 65, a generalisation of Proposition 14),
- an associativity theorem for paths (Proposition 66).

These results are needed for next section. Their proofs require a lot of supplementary formalism, so the reader intuitively convinced may jump directly to next section.

### A.1 Multi-Designs

The notion of *multi-design* introduced below generalises the one of *anti-design* given by Terui [17], thus in particular it generalises designs. Interaction between two *compatible* multi-designs $\mathcal{D}$ and $\mathcal{E}$ corresponds to eliminating the cuts in another multi-design $\text{Cut}_{\mathcal{D}\mid\mathcal{E}}$. Several well-known notions of Ludics can be extended to this setting.

**Definition 45.**

- A negative multi-design is a set $\{ (x_1, n_1), \ldots, (x_n, n_n) \}$ where $x_1, \ldots, x_n$ are distinct variables and $n_1, \ldots, n_n$ are negative designs, such that for all $1 \leq i \leq n$, $\text{fv}(n_i) \cap \{ x_1, \ldots, x_n \} = \emptyset$, and for all $j \neq i$, $\text{fv}(n_j) \cap \text{fv}(n_i) = \emptyset$.
- A positive multi-design is a set $\{ p, (x_1, n_1), \ldots, (x_n, n_n) \}$ where $\{ (x_1, n_1), \ldots, (x_n, n_n) \}$ is a negative multi-design and $p$ is a positive design such that $\text{fv}(p) \cap \{ x_1, \ldots, x_n \} = \emptyset$, and for all $1 \leq i \leq n$, $\text{fv}(p) \cap \text{fv}(n_i) = \emptyset$.

We will use $\mathcal{D}, \mathcal{E}, \ldots$ to denote multi-designs of any polarity, $\mathfrak{N}, \mathfrak{N}, \ldots$ for negative ones and $\mathfrak{P}, \mathfrak{P}, \ldots$ for positive ones. A pair $(x, n)$ in a multi-design will be denoted by $n/x$ or $(n/x)$; hence a negative multi-design will be written $\{ n_1/x_1, \ldots, n_n/x_n \}$ (or even $n/x$), a positive one $\{ p, n_1/x_1, \ldots, n_n/x_n \}$, and we will write $(n/x) \in \mathcal{D}$ instead of $(x, n) \in \mathcal{D}$. This notation makes the parallel with substitution: if $\mathfrak{N} = \{ n_1/x_1, \ldots, n_n/x_n \}$ and $\mathfrak{d}$ is a design, then we will allow to write $\mathfrak{d}[\mathfrak{N}]$ for the substitution $\mathfrak{d}[n_1/x_1, \ldots, n_n/x_n]$. By abuse, we might also write $n \in \mathcal{D}$ when the variable associated to $n$ in the multi-design $\mathcal{D}$ does not matter; thus when writing “let $\mathfrak{d} \in \mathcal{D}$”, the design $\mathfrak{d}$ can be either positive or negative associated with a variable in $\mathcal{D}$.

A design can be viewed as a multi-design: a positive design $p$ corresponds to the positive multi-design $\{ p \}$, and a negative design $n$ to the negative multi-design $\{ n/x_0 \}$, where $x_0$ is the same distinguished variable we introduced for atomic designs. Notations $p$ and $n$ will be used instead of $\{ p \}$ and $\{ n/x_0 \}$ respectively.

Note that if $\mathcal{D}$ and $\mathcal{E}$ are multi-designs, $\mathcal{D} \cup \mathcal{E}$ is not always a multi-design.
**Definition 46.** Let $D$ be a multi-design. Its **normal form** is the cut-free multi-design defined by

\[
\langle D \rangle = \{([n]/x) \mid (n/x) \in D\} \cup \{[p] \mid p \in D\}
\]

**Definition 47.** Let $D$ be a multi-design.

- The **free variables** of $D$ are $fv(D) = \bigcup_{d \in D} fv(d)$
- The **negative places** of $D$ are $np(D) = \{x \mid \exists n (n/x) \in D\}$.

In Definition 45, the condition “for all $1 \leq i \leq n$, $fv(n_i) \cap \{x_1, \ldots, x_n\} = \emptyset$” (adding the similar condition for $p$ in the positive case) can thus be rephrased as “$fv(D) \cap np(D) = \emptyset$”. When two multi-designs $D$ and $E$ interact, this condition will ensure that a substitution specified in $D$ or in $E$ creates a cut between a design from $D$ and a design from $E$, and never between two designs on the same side. This is exactly the form of interaction we want in the following: an interaction with two distinct sides. But in order to talk about interaction between two multi-designs, we must first determine when two multi-designs are **compatible**, i.e., when we can define substitution between them in a unique way, without ambiguity, which is not the case in general.

**Definition 48.** Let $D$ and $E$ be multi-designs.

- $D$ and $E$ are **compatible** if they satisfy the following conditions:
  - $fv(D) \cap fv(E) = np(D) \cap np(E) = \emptyset$
  - either they are both negative and there exists $x \in np(D) \cup np(E)$ such that $x \notin fv(D) \cup fv(E)$, or they are of opposite polarities
- $D$ and $E$ are **closed-compatible** if they are of opposite polarities, compatible, and satisfying $fv(D) = np(E)$ and $fv(E) = np(D)$.

Intuitively, compatible means that we are able to define the multi-design of the interaction between $D$ and $E$, and closed-compatible means that this multi-design is a closed design.

**Definition 49.** Let $D$ and $E$ be compatible multi-designs. $\text{Cut}_{D|E}$ is a multi-design defined by induction on the number of designs in $E$ by:

\[
\text{Cut}_{D|\emptyset} = D \tag{1}
\]

\[
\text{Cut}_{D|\{p\}} = \text{Cut}_{(D \setminus S) \cup [p[S]]|E'} \tag{2}
\]

\[
\text{Cut}_{D|\{n/x\}} = \text{Cut}_{(D \setminus S) \cup [n[S]/x]|E'} \tag{3}
\]

\[
\text{Cut}_{D|\{y\}} = \text{Cut}_{(D \setminus S) \cup [n[S]/x]|E'} \tag{4}
\]

where $S = \{(m/y) \in D \mid y \in fv(p)\}$ in (2)

$S = \{(m/y) \in D \mid y \in fv(n)\}$ in (3) and (4).

The successive pairs of compatible (resp. closed-compatible) multi-designs stay compatible (resp. closed-compatible) after one step of the definition, thus this is well defined. Moreover, if $D$ and $E$ are closed-compatible then, according to the base case, $\text{Cut}_{D|E}$ will be a closed design. In particular, if $p$ and $n$ are atomic designs then $\text{Cut}_{p|n} = p[n/x_0]$.

In order to prove an **associativity** theorem for multi-designs, recall first the original theorem on designs:

**Theorem 50 (Associativity).** Let $d$ be a design and $n_1, \ldots, n_k$ be negative designs.

\[
([d[n_1/y_1, \ldots, n_k/y_k]]) = ([d][([n_1]/y_1, \ldots, ([n_k]/y_k))]).
\]

This result was first established by Girard [11]. The theorem, in the form given above, was proved by Basaldella and Terui [4]. Associativity naturally extends to multi-designs as follows:
Theorem 51 (Multi-associativity). Let $\mathcal{D}$ be a multi-design and $n_1, \ldots, n_k$ be negative designs.

$$[\mathcal{D}[n_1/y_1, \ldots, n_k/y_k]] = [[\mathcal{D}][[n_1]/y_1, \ldots, [n_k]/y_k]].$$

Proof. Immediate from the definition of the normal form of a multi-design (Definition 46) and simple associativity (Theorem 50). \qed

Corollary 52.

$$[\text{Cut}_\mathcal{D}[\epsilon]] = [[\text{Cut}_\mathcal{D}][[\epsilon]]].$$

Proof. By induction on $\mathcal{E}$:

- If $\mathcal{E} = \emptyset$ then $[\text{Cut}_\mathcal{D}][\emptyset] = [\mathcal{D}] = [[\text{Cut}_\mathcal{D}][[\emptyset]]] = [[\text{Cut}_\mathcal{D}][\emptyset]].$

- If $\mathcal{E} = \mathcal{E} \cup \{p\}$, let $S = \{m_1/y_1, \ldots, m_k/y_k\} = \{(m/y) \in \mathcal{D} \mid y \in \text{fv}(p)\}$. By definitions of the normal form of multi-designs (Definition 46) and of Cut (Definition 49), and using associativity (Theorem 51), we have:

  $$[\text{Cut}_\mathcal{D}[\epsilon]] = [[\text{Cut}_\mathcal{D}[S][\{p|S\}]]/\epsilon']] \quad \text{by Def. 49}$$

  $$= [[\text{Cut}_\mathcal{D}[S][\{p|S\}]]/\epsilon']] \quad \text{by induction hypothesis}$$

  $$= [[\text{Cut}_\mathcal{D}[S][\{p|S\}]]/\epsilon']] \quad \text{by Def. 46 and Thm. 51}$$

  $$= [[\text{Cut}_\mathcal{D}[S][\{p|S\}]]/\epsilon']] \quad \text{by Def. 46}$$

  $$= [[\text{Cut}_\mathcal{D}[S][\{p|S\}]]/\epsilon']] \quad \text{by Def. 46 and 49}$$

- If $\mathcal{E} = \mathcal{E} \cup \{n/x\}$ with $x \notin \text{fv}(\mathcal{D})$, similar as above with $S = \{(m/y) \in \mathcal{D} \mid y \in \text{fv}(n)\}$.

- If $\mathcal{E} = \mathcal{E} \cup \{n/x\}$ with $x \in \text{fv}(\mathcal{D})$, let $S = \{m_1/y_1, \ldots, m_k/y_k\} = \{(m/y) \in \mathcal{D} \mid y \in \text{fv}(n)\}$. We have:

  $$[\text{Cut}_\mathcal{D}[\epsilon]] = [[\text{Cut}_\mathcal{D}[S][n|S/x]]/\epsilon']] \quad \text{by Def. 49}$$

  $$= [[\text{Cut}_\mathcal{D}[S][n|S/x]]/\epsilon']] \quad \text{by induction hypothesis}$$

  $$= [[\text{Cut}_\mathcal{D}[S][n|S/x]]/\epsilon']] \quad \text{by Def. 49 and Thm. 51 twice}$$

  $$= [[\text{Cut}_\mathcal{D}[S][n|S/x]]/\epsilon']] \quad \text{by Thm. 51}$$

  $$= [[\text{Cut}_\mathcal{D}[S][n|S/x]]/\epsilon']] \quad \text{by induction hypothesis}$$

  $$= [[\text{Cut}_\mathcal{D}[S][n|S/x]]/\epsilon']] \quad \text{by Def. 46 and 49}$$

\qed

Lemma 53. $\text{Cut}_\mathcal{D}[\epsilon] = \text{Cut}_{\mathcal{E}[\mathcal{D}]}$.

Proof. By induction on the number $n$ of variables in $(\text{fv}(\mathcal{D}) \cap \text{np}(\mathcal{E})) \cup (\text{fv}(\mathcal{E}) \cap \text{np}(\mathcal{D}))$.

- If $n = 0$ then $\text{Cut}_{\mathcal{E}[\mathcal{D}]} = \text{Cut}_\mathcal{E}[\mathcal{D} \cup \mathcal{E}]$.

- Let $n > 0$ and suppose the property is satisfied for all $k < n$. Without loss of generality suppose there exists $x \in (\text{fv}(\mathcal{D}) \cap \text{np}(\mathcal{E}))$. Thus $\mathcal{E}$ is of the form $\mathcal{E} = \mathcal{E}' \cup \{n/x\}$. Let $S = \{(m/y) \in \mathcal{D} \mid y \in \text{fv}(n)\}$.

  - If $S = \emptyset$, let $\mathcal{D} \subseteq \mathcal{D}$ be the design such that $x \in \text{fv}(\mathcal{D})$, and let us write $\mathcal{D} = \mathcal{D}' \cup \{x\}$. If $\mathcal{D}$ is positive then:

    $$\text{Cut}_{\mathcal{D}[\epsilon]} = \text{Cut}_{\mathcal{D}' \cup \{x\}[\epsilon']} \quad \text{by one step 4 of Def. 49}$$

    $$= \text{Cut}_{\mathcal{E}'[\mathcal{D}' \cup \{x\}[\epsilon']] \quad \text{by induction hypothesis}$$

    $$= \text{Cut}_{\mathcal{E}'[\mathcal{D}' \cup \{x\}[\epsilon']} \quad \text{by one step 2 of Def. 49}$$
where $T' = \{(m/y) \in \mathcal{E} \mid y \in \text{fv}(\mathcal{D}[n/x])\}$. Let $T = \{(m/y) \in \mathcal{E} \mid y \in \text{fv}(\mathcal{D})\}$, we have $T = T' \cup \{n/x\}$, indeed: $\text{fv}(\mathcal{D}[n/x]) = (\text{fv}(\mathcal{D}) \setminus \{x\}) \cup \text{fv}(n)$, where $\text{fv}(n) \cap \text{np}(\mathcal{E}) = \emptyset$ by definition of a multi-design, thus also $\text{fv}(n) \cap \text{np}(\mathcal{E}') = \emptyset$. Therefore:

$$\text{Cut}_{\mathcal{E}\setminus\mathcal{D}[n/x]}(\mathcal{E}\setminus\mathcal{D}[n/x],T') = \text{Cut}_{\mathcal{E}\setminus\mathcal{D}[n/x]}(\mathcal{E}\setminus\mathcal{D}[n/x])$$

by one step 2 of Def. 49 backwards, hence the result. The reasoning is similar if $\mathcal{D}$ is negative and $\mathcal{D} = \mathcal{D}' \cup \{\mathcal{D}/y\}$, we just have to distinguish between the cases $y \in \text{fv}(\mathcal{E}')$ and $y \notin \text{fv}(\mathcal{E}')$.

- Otherwise, let $S' = \{(m/y) \in \mathcal{E} \mid y \in \text{fv}(\mathcal{E})\}$ and $S'' = \{(m/y) \in \mathcal{D} \mid y \in \text{fv}(\mathcal{E}')\}$; note that $S' \subseteq \mathcal{E}'$ and $S'' \subseteq (\mathcal{D} \setminus S)$. We have:

$$\text{Cut}_{\mathcal{E}\setminus\mathcal{D}[n/x]}(\mathcal{E}\setminus\mathcal{D}[n/x]) = \text{Cut}_{\mathcal{E}\setminus\mathcal{D}[n/x]}(\mathcal{E}\setminus\mathcal{D}[n/x])$$

by several steps 4 of Def. 49

$$= \text{Cut}_{\mathcal{E}\setminus\mathcal{D}[n/x]}(\mathcal{E}\setminus\mathcal{D}[n/x])$$

by induction hypothesis, since $S \neq \emptyset$

$$= \text{Cut}_{\mathcal{E}\setminus\mathcal{D}[n/x]}(\mathcal{E}\setminus\mathcal{D}[n/x])$$

by one step 4 of Def. 49

$$= \text{Cut}_{\mathcal{E}\setminus\mathcal{D}[n/x]}(\mathcal{E}\setminus\mathcal{D}[n/x])$$

by steps 4 of Def. 49 backwards

The last equality is obtained by moving successively, from left to right, all the designs from $S'$, and finally the design $n$.

\[\square\]

**Lemma 54.** Let $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{E}$ be multi-designs such that $\mathcal{D}_1 \cup \mathcal{D}_2$ is a multi-design, and $\mathcal{E}$ is compatible with $\mathcal{D}_1 \cup \mathcal{D}_2$. We have:

$$\text{Cut}_{\mathcal{D}_1 \cup \mathcal{D}_2}[\mathcal{E}] = \text{Cut}_{\mathcal{D}_1}[\text{Cut}_{\mathcal{D}_2}[\mathcal{E}]]$$

**Proof.** By induction on $\mathcal{D}_2$:

- If $\mathcal{D}_2 = \emptyset$ then $\text{Cut}_{\mathcal{E}\setminus\mathcal{D}_2} = \mathcal{E}$ hence the result.

- If $\mathcal{D}_2 = \mathcal{D}' \cup \{\mathcal{D}/y\}$ then $\text{Cut}_{\mathcal{E}\setminus\mathcal{D}_2} = \text{Cut}_{\mathcal{E}\setminus\mathcal{D}_2} = \text{Cut}_{\mathcal{E}\setminus\mathcal{D}_2}$ where $S = \{(m/y) \in \mathcal{E} \mid y \in \text{fv}(\mathcal{D})\}$. Thus by induction hypothesis:

$$\text{Cut}_{\mathcal{D}_1}[\text{Cut}_{\mathcal{E}\setminus\mathcal{D}_2}] = \text{Cut}_{\mathcal{D}_1}[\text{Cut}_{\mathcal{E}\setminus\mathcal{D}_2}]$$

by one step 2 of Def. 49

$$= \text{Cut}_{\mathcal{D}_1}[\text{Cut}_{\mathcal{E}\setminus\mathcal{D}_2}]$$

by Lemma 53

$$= \text{Cut}_{\mathcal{D}_1}[\text{Cut}_{\mathcal{E}\setminus\mathcal{D}_2}]$$

by one step 4 backwards of Def. 49

We now extend the notion of orthogonality to multi-designs.

**Definition 55.** Let $\mathcal{D}$ and $\mathcal{E}$ be closed-compatible multi-designs. $\mathcal{D}$ and $\mathcal{E}$ are **orthogonal**, written $\mathcal{D} \perp \mathcal{E}$, if $\text{Cut}_{\mathcal{D} \setminus \mathcal{E}} \downarrow \emptyset$.
A.2 Paths and Multi-Designs

Recall that we write $\epsilon$ for the empty sequence.

**Definition 56.** Let $\mathcal{D}$ be a multi-design.

- A **view** of $\mathcal{D}$ is a view of a design in $\mathcal{D}$.

- A **path** of $\mathcal{D}$ is a path $s$ of same polarity as $\mathcal{D}$ such that for all prefix $s'$ of $s$, $s'$ is a view of $\mathcal{D}$.

We are now interested in a particular form of closed interactions, where we can identify two sides of the multi-design: designs are separated into two groups such that there are no cuts between designs of the same group. This corresponds exactly to the interaction between two closed-compatible multi-designs.

**Definition 57.** Let $\mathcal{D}$ and $\mathcal{E}$ be closed-compatible multi-designs such that $\mathcal{D} \perp \mathcal{E}$. The **interaction path** of $\mathcal{D}$ with $\mathcal{E}$ is the unique path $s$ of $\mathcal{D}$ such that $s$ is a path of $\mathcal{E}$.

But nothing ensures the existence and uniqueness of such a path: this will be proved in the rest of this subsection. We will moreover show that, if $\mathcal{D} \perp \mathcal{E}$, this path corresponds to the interaction sequence defined below. For the purpose of giving an inductive definition of the interaction sequence, we define it not only for a pair of closed-compatible multi-designs but for a larger class of pairs of multi-designs.

**Definition 58.** Let $\mathcal{D}$ and $\mathcal{E}$ be multi-designs of opposite polarities, compatible, and satisfying $\text{fv}(\mathcal{D}) \subseteq \text{np}(\mathcal{E})$ and $\text{fv}(\mathcal{E}) \subseteq \text{np}(\mathcal{D})$. The **interaction sequence** of $\mathcal{D}$ with $\mathcal{E}$, written $(\mathcal{D} \leftarrow \mathcal{E})$, is the sequence of actions followed by interaction on the side of $\mathcal{D}$. More precisely, if we write $p$ for the only positive design of $\mathcal{D} \cup \mathcal{E}$, the interaction sequence is defined recursively by:

- If $p = \mathcal{K}$ then:
  - $(\mathcal{D} \leftarrow \mathcal{E}) = \mathcal{K}$ if $\mathcal{K} \in \mathcal{D}$
  - $(\mathcal{D} \leftarrow \mathcal{E}) = \epsilon$ if $\mathcal{K} \in \mathcal{E}$.

- If $p = \Omega$ then $(\mathcal{D} \leftarrow \mathcal{E}) = \epsilon$.

- If $p = x[p(\overrightarrow{\mathcal{M}})]$ then there exists $n$ such that $(n/x) \in \mathcal{E}$ if $p \in \mathcal{D}$, $(n/x) \in \mathcal{D}$ otherwise. Let us write $n = \sum_{b \in S} b(y^b).p_b$. We have $(\mathcal{D} \leftarrow \mathcal{E}) = \kappa(\mathcal{D}' \leftarrow \mathcal{E}')$ where:
  - if $p \in \mathcal{D}$, then $\kappa = x[p(\overrightarrow{\mathcal{M}})]$, $\mathcal{D}' = (\mathcal{D} \setminus \{p\}) \cup \{\overrightarrow{m/y^b}\}$ and $\mathcal{E}' = (\mathcal{E} \setminus \{n/x\}) \cup \{p_a\}$.
  - otherwise $\kappa = a_a(x^b)$, $\mathcal{D}' = (\mathcal{D} \setminus \{n/x\}) \cup \{p_a\}$ and $\mathcal{E}' = (\mathcal{E} \setminus \{p\}) \cup \{\overrightarrow{m/y^b}\}$.

Note that this applies in particular to two closed-compatible multi-designs. Remark also that this definition follows exactly the interaction between $\mathcal{D}$ and $\mathcal{E}$: indeed, in the inductive case of the definition, the multi-designs $\mathcal{D}'$ and $\mathcal{E}'$ are obtained from $\mathcal{D}$ and $\mathcal{E}$ similarly to the following lemma. In particular the interaction sequence is finite whenever the interaction between $\mathcal{D}$ and $\mathcal{E}$ is finite.

**Lemma 59.** Let $\mathcal{D}$ and $\mathcal{E}$ be closed-compatible multi-designs of opposite polarities. Suppose the only positive design $p \in \mathcal{D}$ is of the form $p = x[p(\overrightarrow{\mathcal{M}})]$, and suppose moreover there exists $n_0$ such that $(n_0/x) \in \mathcal{E}$, say $n_0 = \sum_{b \in S} b(x^b).p_b$. Then:

$$\text{Cut}_{\mathcal{D} \leftarrow \mathcal{E}} \sim \text{Cut}_{\mathcal{D}' \leftarrow \mathcal{E}'} \setminus \{(m/x^b) \mid x^b \notin \text{fv}(p_a)\}$$

where $\mathcal{D}' = (\mathcal{D} \setminus \{p\}) \cup \{n/x^b\}$ and $\mathcal{E}' = (\mathcal{E} \setminus \{n_0/x\}) \cup \{p_a\}$.  

21
Proof. First notice that as $\mathcal{D}$ and $\mathcal{E}$ are closed-compatible, $\text{Cut}_{\mathcal{D}|\mathcal{E}}$ is a design, and since this design has cuts we can indeed apply one step of reduction to it. Let $S' = \{(m/x_i) \ | \ x_i \notin \text{fv}(p_a)\}$. We have to prove $\text{Cut}_{\mathcal{D}|\mathcal{E}} \rightarrow \text{Cut}_{\mathcal{D}|\mathcal{E}'} \setminus S'$. The proof is done by induction on the number of designs in $\mathcal{E}$.

- If $\mathcal{E} = \{n_0/x\}$, then $\mathcal{E}' = \{p_a\}$. In this case let $S = \{(m/y) \in \mathcal{D} \ | \ y \in \text{fv}(n_0)\}$, and remark that, as $\mathcal{E}$ and $\mathcal{D}$ are closed-compatible, $S = \mathcal{D} \setminus \{p\}$. Thus:

$$
\text{Cut}_{\mathcal{D}|\mathcal{E}} = \text{Cut}_{\mathcal{D}\setminus S[n_0][x]/\emptyset} = p\langle n_0[S]/x \rangle
\rightarrow p_a[S][n/x^x]
= p_a[\mathcal{D}']
= \{p_a[\mathcal{D}' \cup S_0] \cup S_0 \setminus S' \} = \text{Cut}_{\mathcal{D}' \cup \{p_a[\mathcal{D}' \cup S]\} \setminus S' \setminus S'} = \text{Cut}_{\mathcal{D}' \setminus p_a \setminus S'} = \text{Cut}_{\mathcal{D}' \setminus \mathcal{E}' \setminus S'}
$$

by one step 4 of Def. 49

- Otherwise there exists $(n_1/z) \in \mathcal{E}$ such that $x \neq z$. Suppose $z \notin \text{fv}(\mathcal{D})$ (resp. $z \in \text{fv}(\mathcal{D})$). Define:

- $S = \{(m/y) \in \mathcal{D} \ | \ y \in \text{fv}(n_1)\}$, and remark that $S = \{(m/y) \in \mathcal{D}' \ | \ y \in \text{fv}(n_1)\}$.
- $\mathcal{D}'' = (\mathcal{D}' \setminus S) \cup \{(n_1[S]/z)\}$ (resp. $\mathcal{D}'' = (\mathcal{D}' \setminus S)[n_1[S]/z]$)
- $\mathcal{E}'' = \mathcal{E}' \setminus \{(n_1/z)\}$.

We have:

$$
\text{Cut}_{\mathcal{D}|\mathcal{E}} = \text{Cut}_{\mathcal{D}\setminus S \cup \{(n_1[S]/z)\}\setminus \mathcal{E} \setminus \{n_1/z\}} = \text{Cut}_{\mathcal{D}' \setminus \mathcal{E}'' \setminus S'}
$$

by one step 3 of Def. 49

 resp. $= \text{Cut}_{\mathcal{D}\setminus S \cup \{(n_1[S]/z)\}\setminus \mathcal{E} \setminus \{n_1/z\}} = \text{Cut}_{\mathcal{D}' \setminus \mathcal{E}'' \setminus S'}$

by one step 4 of Def. 49

 resp. $= \text{Cut}_{\mathcal{D}' \setminus \mathcal{E}'' \setminus S'}$

by induction hypothesis

resp. $= \text{Cut}_{\mathcal{D}' \setminus \mathcal{E}'' \setminus S'}$

by step 3 (resp. 4) of Def. 49 backwards

$\square$

Lemma 60. If $\mathcal{X} \in \{\text{Cut}_{\mathcal{D}|\mathcal{E}}\}$ (in particular if $\mathcal{D} \perp \mathcal{E}$) then $\langle \mathcal{D} \gets \mathcal{E} \rangle = \langle \mathcal{E} \gets \mathcal{D} \rangle$. Otherwise $\langle \mathcal{D} \gets \mathcal{E} \rangle = \langle \mathcal{E} \gets \mathcal{D} \rangle$.

Proof. It is clear from the definition of the interaction sequence that the proper actions in $\langle \mathcal{D} \gets \mathcal{E} \rangle$ are the opposite of those in $\langle \mathcal{E} \gets \mathcal{D} \rangle$. Concerning the daimon: since the interaction sequence follows the interaction between $\mathcal{D}$ and $\mathcal{E}$, $\mathcal{X}$ appears at the end of one of the sequences $\langle \mathcal{D} \gets \mathcal{E} \rangle$ or $\langle \mathcal{E} \gets \mathcal{D} \rangle$ if and only if $\mathcal{X} \in \{\text{Cut}_{\mathcal{D}|\mathcal{E}}\}$, and in this case $\langle \mathcal{D} \gets \mathcal{E} \rangle = \langle \mathcal{E} \gets \mathcal{D} \rangle$.

$\square$

Proposition 61. Every positive-ended prefix of $\langle \mathcal{D} \gets \mathcal{E} \rangle$ is a path of $\mathcal{D}$. In particular, if $\langle \mathcal{D} \gets \mathcal{E} \rangle$ is finite and positive-ended then it is a path of $\mathcal{D}$.

Proof. First remark that every (finite) prefix of $\langle \mathcal{D} \gets \mathcal{E} \rangle$ is an aj-sequence. Indeed, since $\mathcal{D}$ and $\mathcal{E}$ are well shaped multi-designs the definition of interaction sequence ensures that an action cannot appear before its justification, and all the conditions of the definition of an aj-sequence are satisfied: Alternation and Daimon are immediate from the definition of interaction sequence, while Linearity is indeed satisfied as variables are disjoint in $\mathcal{D}$ and $\mathcal{E}$ (by Barendregt’s convention).

By definition, for every prefix $s$ of $\langle \mathcal{D} \gets \mathcal{E} \rangle$, $\langle s \rangle$ is a view. We show that it is a view of $\mathcal{D}$ by induction on the length of $s$.
• If \( s = \epsilon \) then \( \gamma^s = \epsilon \) is indeed a view of \( \mathcal{D} \).

• If \( s = \nu \) then \( \langle \mathcal{D} \leftarrow \mathcal{E} \rangle = \nu \). From the definition of interaction sequence, we know that in this case \( \nu \in \mathcal{D} \), hence \( \gamma^\nu = \nu \) is a view of \( \mathcal{D} \).

• If \( s = \kappa s' \) where \( \kappa \) is proper, then \( \langle \mathcal{D} \leftarrow \mathcal{E} \rangle = \kappa \langle \mathcal{D}' \leftarrow \mathcal{E}' \rangle \) where \( \mathcal{D}' \) and \( \mathcal{E}' \) are as in Definition 58, and \( s' \) is a prefix of \( \langle \mathcal{D}' \leftarrow \mathcal{E}' \rangle \). By induction hypothesis, \( \gamma^{s'} \) is a view of \( \mathcal{D}' \).

Two possibilities:

- Either \( \kappa = \kappa^+ \) is positive. From the definition of the interaction sequence, this means \( \mathcal{p} := x[p(m)] \in \mathcal{D}, \kappa^+ = x[p(y)] \) and \( \mathcal{D}' = (\mathcal{D} \setminus \{p\}) \cup \{m/y\}. \) We have \( \gamma^s = \kappa^+ \gamma^{s'} \) and either \( \gamma^{\kappa^+ s'} = \kappa^+ \gamma^{s'} \) if the first negative action of \( \gamma^{s'} \) is justified by \( \kappa^+ \) (i.e., \( \exists i \) such that \( \gamma^{s'} \) is a view of \( m_i/y_i \)), or \( \gamma^{\kappa^+ s'} = \gamma^{s'} \) otherwise (i.e., \( \gamma^{s'} \) is a view of \( \mathcal{D} \setminus \{p\} \)). In the second case, there is nothing more to show; in the first one, by definition of the views of a design, \( \kappa^+ \gamma^{s'} \) is a view of \( \mathcal{D} \).

- Or \( \kappa = \kappa^- \) is negative. Hence there exists a design \( \mathcal{a} = \sum_{b \in \mathcal{S}} b(y)^p \) such that \( (n/x) \in \mathcal{D}, \kappa^- = a(x), \) and \( \mathcal{D}' = (\mathcal{D} \setminus \{n/x\}) \cup \{p_a\} \). We have \( \gamma^s = \kappa^- \gamma^{s'} \) and either \( \gamma^{\kappa^- s'} = \kappa^- \gamma^{s'} \) if the first action of \( \gamma^{s'} \) is positive (i.e., \( \gamma^{s'} \) is a view of \( \mathcal{p}_a \)), or \( \gamma^{\kappa^- s'} = \gamma^{s'} \) otherwise (i.e., \( \gamma^{s'} \) is a view of \( \mathcal{D}' \setminus \{p_a\} \)). In the second case, there is nothing to do; in the first one, note that \( \kappa^- \gamma^{s'} \) is a view of \( (n/x) \), hence the result.

We have proved that \( \gamma^{s'} \) is a view of \( \mathcal{D} \). This implies in particular that \( \langle \mathcal{D} \leftarrow \mathcal{E} \rangle \) satisfies \( P \)-visibility, indeed: given a prefix \( \kappa s' \) of \( \langle \mathcal{D} \leftarrow \mathcal{E} \rangle \), the action \( \kappa^+ \) is either initial or it is justified in \( s \) by the same action that justifies it in \( \mathcal{D} \); since \( \gamma^{s'} \) is a view of \( \mathcal{D} \), the justification of \( \kappa^+ \) is in it, thus \( P \)-visibility is satisfied. Similarly, we can prove that \( \gamma^{r} \) is a view of \( \mathcal{E} \) whenever \( t \) is a prefix of \( \langle \mathcal{E} \leftarrow \mathcal{D} \rangle \), therefore \( \langle \mathcal{E} \leftarrow \mathcal{D} \rangle \) also satisfies \( P \)-visibility; by Lemma 60 either \( \langle \mathcal{E} \leftarrow \mathcal{D} \rangle = \langle \mathcal{D} \leftarrow \mathcal{E} \rangle \) or \( \langle \mathcal{E} \leftarrow \mathcal{D} \rangle = \langle \mathcal{D} \leftarrow \mathcal{E} \rangle \), thus this implies that \( \langle \mathcal{D} \leftarrow \mathcal{E} \rangle \) satisfies \( O \)-visibility. Hence every positive-ended prefix of \( \langle \mathcal{D} \leftarrow \mathcal{E} \rangle \) is a path, and since the views of its prefixes are views of \( \mathcal{D} \), it is a path of \( \mathcal{D} \).

Remark. If \( sk_1^+ \) and \( sk_2^+ \) are views (resp. paths) of a multi-design \( \mathcal{D} \) then \( \ldots \). Indeed, if \( sk_1^+ \) and \( sk_2^+ \) are views of \( \mathcal{D} \), the result is immediate by definition of the views of a design; if they are paths of \( \mathcal{D} \), just remark that \( \gamma^{sk_1^+} = \gamma^{s} \gamma^k_1 \) and \( \gamma^{sk_2^+} = \gamma^{s} \gamma^k_2 \) are views of \( \mathcal{D} \), hence the conclusion.

Proposition 62. Suppose \( \mathcal{D} \models \mathcal{E}, s \) is a path of \( \mathcal{D} \) and \( \mathcal{z} \) is a path of \( \mathcal{E} \). The path \( s \) is a prefix of \( \langle \mathcal{D} \leftarrow \mathcal{E} \rangle \).

Proof. Suppose \( s \) is not a prefix of \( \langle \mathcal{D} \leftarrow \mathcal{E} \rangle \). Let \( t \) be the longest common prefix of \( s \) and \( \langle \mathcal{D} \leftarrow \mathcal{E} \rangle \) (possibly \( \epsilon \)). Without loss of generality, we can assume there exist actions of same polarity \( \kappa_1 \) and \( \kappa_2 \) such that \( \kappa_1 \neq \kappa_2 \). \( \tau k_1 \) is a prefix of \( s \) and \( \tau k_2 \) is a prefix of \( \langle \mathcal{D} \leftarrow \mathcal{E} \rangle \): indeed, if there are no such actions, it is because \( \langle \mathcal{D} \leftarrow \mathcal{E} \rangle \) is a strict prefix of \( s \); in this case, it suffices to consider \( \langle \mathcal{E} \leftarrow \mathcal{D} \rangle \) and \( \mathcal{z} \) instead.

- If \( \kappa_1 \) and \( \kappa_2 \) are positive, then \( \tau k_1 \) and \( \tau k_2 \) are paths of \( \mathcal{D} \), and by previous remark \( \kappa_1 = \kappa_2 \): contradiction.

- If \( \kappa_1 \) and \( \kappa_2 \) are negative, a contradiction arises similarly from the fact that \( \overline{\tau k_1} \) and \( \overline{\tau k_2} \) are paths of \( \mathcal{E} \) where \( \kappa_1 \) and \( \kappa_2 \) are positive.

Hence the result. □

The following result ensures that the interaction path is well defined.
Proposition 63. If $\mathcal{D} \perp \mathcal{E}$, there exists a unique path $s$ of $\mathcal{D}$ such that $\bar{s}$ is a path of $\mathcal{E}$.

Proof. Lemma 60 and Proposition 61 show that such a path exists, namely $(\mathcal{D} \leftarrow \mathcal{E})$. Unicity follows from Proposition 62.

Conversely, we prove that the existence of such a path implies the orthogonality of multi-designs (Proposition 65).

Proposition 64. Let $\mathcal{P}$ and $\mathcal{N}$ be closed-compatible multi-designs such that $\Omega \notin \mathcal{P}$ and such that their interaction is finite. Suppose that for every path $s\kappa^+$ of $\mathcal{P}$ such that $\kappa^+$ is proper and $\bar{s}$ is a path of $\mathcal{N}$, $s\kappa^+$ is a path of $\mathcal{N}$, and suppose also that the same condition is satisfied when reversing $\mathcal{P}$ and $\mathcal{N}$. Then $\mathcal{P} \perp \mathcal{N}$.

Proof. By induction on the number $n$ of steps of the interaction before divergence/convergence:

- If $n = 0$, then we must have $\mathcal{P} = \mathcal{N}$, since $\Omega \notin \mathcal{P}$. Hence the result.
- If $n > 0$ then $p \in \mathcal{P}$ is of the form $p = x[\pi(\bar{p})]$ and there exists $n_0 = \sum_{b \in S} b(x^0).p_b$ such that $(n_0/x) \in \mathcal{N}$. Let $\kappa^+ = x[\pi(\bar{x^0})]$ and remark that $\kappa^+$ is a path of $p$. By hypothesis, $\bar{\kappa} = a_x(x^0)$ is a path of $\mathcal{N}$, thus a path of $n_0$, and this implies $p_a \neq \Omega$. By Lemma 59, we have $\mathcal{Cut}_{p|\pi} \mathcal{Cut}_{p|\pi} \setminus \{\langle m/x^0 \rangle | x \notin \text{fv}(p_a)\}$ where $\mathcal{P}' = (\mathcal{P} \setminus \{p\}) \cup \{n/x^0\}$ and $\mathcal{N}' = (\mathcal{N} \setminus \{n_0/x\}) \cup \{p_a\}$. This corresponds to a cut-net between two closed-compatible multi-designs $\mathcal{P}'' \subseteq \mathcal{P}'$ (negative) and $\mathcal{N}'' \subseteq \mathcal{N}'$ (positive), where:
  - $\Omega \notin \mathcal{N}$'' because $p_a \neq \Omega$;
  - their interaction is finite and takes $n - 1$ steps;
  - the condition on paths stated in the proposition is satisfied for $\mathcal{P}''$ and $\mathcal{N}''$, because it is for $\mathcal{P}$ and $\mathcal{N}$: indeed, the paths of $\mathcal{P}''$ (resp. $\mathcal{N}''$) are the paths $t$ such that $\kappa^+ t$ is a path of $\mathcal{P}$ (resp. $\kappa^+ t$ is a path of $\mathcal{N}$), unless such a path $t$ contains a negative initial action whose address is not the address of a positive action on the other side, but this restriction is harmless with respect to our condition.

We apply the induction hypothesis to get $\mathcal{P}'' \perp \mathcal{N}''$. Finally $\mathcal{P} \perp \mathcal{N}$.

Proposition 65. Let $\mathcal{D}$ and $\mathcal{E}$ be closed-compatible multi-designs. $\mathcal{D} \perp \mathcal{E}$ if and only if there exists a path $s$ of $\mathcal{D}$ such that $\bar{s}$ is a path of $\mathcal{E}$.

Proof. $(\Rightarrow)$ If $\mathcal{D} \perp \mathcal{E}$ then result follows from Proposition 63.

$(\Leftarrow)$ We will prove that the hypothesis of Proposition 64 is satisfied. Let us show that every path of $\mathcal{D}$ (resp. of $\mathcal{E}$) of the form $t\kappa^+$ where $\kappa^+$ is proper and $\bar{t}$ is a path of $\mathcal{E}$ (resp. of $\mathcal{D}$) is a prefix of $s$ (resp. of $\bar{s}$).

- If $t$ is empty, $\kappa^+$ is necessarily the first action of the positive design in $\mathcal{D}$ (resp. in $\mathcal{E}$), hence the first action of $s$ (resp. of $\bar{s}$).
- If $t = t_0\kappa^-$, then $t_0\kappa^-$ is a path of $\mathcal{E}$ (resp. of $\mathcal{D}$) and $t_0$ is a path of $\mathcal{D}$ (resp. of $\mathcal{E}$). By induction hypothesis, $\bar{t} = t_0\kappa^-$ is a prefix of $\bar{s}$ (resp. of $s$), thus $t$ is a prefix of $s$ (resp. of $\bar{s}$). The path $s$ is of the form $s = t\kappa^+\bar{s}'$. But since $s$ and $t\kappa^+$ are both paths of $\mathcal{D}$ (resp. $\mathcal{E}$), they cannot differ on a positive action, hence $\kappa^+ = \kappa^+$. Thus $t\kappa^+$ is a prefix of $s$.


A.3 Associativity for Interaction Paths

If \( s \) is a path of a multi-design \( \mathcal{D} \), and \( \mathcal{E} \subseteq \mathcal{D} \), then we write \( s|\mathcal{E} \) for the longest subsequence of \( s \) that is a path of \( \mathcal{E} \). Note that this is well defined.

**Proposition 66** (Associativity for paths). Let \( \mathcal{D} \), \( \mathcal{E} \) and \( \mathcal{F} \) be cut-free multi-designs such that \( \mathcal{E} \cup \mathcal{F} \) is a multi-design, and suppose \( \mathcal{D} \downarrow (\mathcal{E} \cup \mathcal{F}) \). We have:

\[
(\mathcal{E} \leftarrow (\text{Cut}_{\mathcal{F} \mathcal{D}})) = (\mathcal{E} \cup \mathcal{F} \leftarrow \mathcal{D})|\mathcal{E}
\]

**Proof.** We will prove the result for a larger class of multi-designs. Instead of the assumption \( \mathcal{D} \downarrow (\mathcal{E} \cup \mathcal{F}) \), suppose that \( \mathcal{D} \) and \( \mathcal{E} \cup \mathcal{F} \) are:

- of opposite polarities
- compatible
- satisfying \( \text{fv}(\mathcal{D}) \subseteq \text{np}(\mathcal{E} \cup \mathcal{F}) \) and \( \text{fv}(\mathcal{E} \cup \mathcal{F}) \subseteq \text{np}(\mathcal{D}) \)
- and such that \( \mathcal{F} \in \{\text{Cut}_{\mathcal{E} \mathcal{D}}\} \) (in particular their interaction is finite).

First remark that \( \mathcal{F} \) and \( \mathcal{D} \) are compatible, hence it is possible to define \( \text{Cut}_{\mathcal{F} \mathcal{D}} \). Then since \( \mathcal{F} \in \{\text{Cut}_{\mathcal{E} \mathcal{D}}\} \), we have \( \mathcal{F} \in \{\text{Cut}_{\mathcal{E} \mathcal{D}}\} \), indeed:

\[
(\text{Cut}_{\mathcal{E} \mathcal{D}}) = (\text{Cut}_{\mathcal{E} \mathcal{D}}) \quad \text{by Lemmas 54 and 53}
\]

This also shows that \( \mathcal{E} \) and \( \{\text{Cut}_{\mathcal{F} \mathcal{D}}\} \) are compatible. As they are of opposite polarities and they satisfy the condition on variables, \( (\mathcal{E} \leftarrow (\text{Cut}_{\mathcal{F} \mathcal{D}})) \) is defined.

Let \( s = (\mathcal{E} \leftarrow (\text{Cut}_{\mathcal{F} \mathcal{D}})) \), and let us show the result (i.e., \( s|\mathcal{E} = (\mathcal{E} \leftarrow (\text{Cut}_{\mathcal{F} \mathcal{D}})) \)) by induction on the length of \( s \), which is finite because the interaction between \( \mathcal{D} \) and \( \mathcal{E} \cup \mathcal{F} \) is finite.

- If \( s = \epsilon \) then necessarily \( \mathcal{F} \in \mathcal{D} \) thus also \( \mathcal{E} \in \{\text{Cut}_{\mathcal{F} \mathcal{D}}\} \). Hence \( s|\mathcal{E} = \epsilon = (\mathcal{E} \leftarrow (\text{Cut}_{\mathcal{F} \mathcal{D}})) \).
- If \( s = \mathcal{F} \) then \( \mathcal{F} \in \mathcal{E} \cup \mathcal{F} \). If \( \mathcal{F} \in \mathcal{E} \) then \( (\mathcal{E} \leftarrow (\text{Cut}_{\mathcal{F} \mathcal{D}})) = \mathcal{F} = s|\mathcal{E} \). Otherwise \( \mathcal{F} \in \mathcal{F} \), thus \( \mathcal{F} \in \{\text{Cut}_{\mathcal{F} \mathcal{D}}\} \) and \( (\mathcal{E} \leftarrow (\text{Cut}_{\mathcal{F} \mathcal{D}})) = \epsilon = s|\mathcal{E} \).
- If \( s = \kappa^+ s' \) where \( \kappa^+ = x^m \) is a proper positive action, then \( \mathcal{E} \cup \mathcal{F} \) is a positive multi-design such that its only positive design is of the form \( p = x^m \). Thus \( \mathcal{D} \) is negative, and there exists \( n \) such that \( (n/x) \in \mathcal{D} \) of the form \( n = \sum_{a \in S} b(x^a)p_a \), where \( p_a \neq \Omega \) because the interaction converges. Let \( \mathcal{D}' = (\mathcal{D} \\{n/x\} \cup \{p_a\} \).

- Either \( p \in \mathcal{F} \) [reduction step].
  In this case, we have \( s|\mathcal{E} = s'|\mathcal{E} \), so let us show that \( s'|\mathcal{E} = \mathcal{E} \leftarrow (\text{Cut}_{\mathcal{F} \mathcal{D}}) \). By definition of the interaction sequence, we have \( s' = \mathcal{E} \mathcal{F} \mathcal{D} \) with \( \mathcal{F} = \mathcal{E} \mathcal{D} \) and \( \mathcal{D} = \mathcal{E} \mathcal{D} \). Thus by induction hypothesis \( s'|\mathcal{E} = \mathcal{E} \leftarrow (\text{Cut}_{\mathcal{F} \mathcal{D}}) \). But by Lemma 59, \( (\mathcal{E} \leftarrow (\text{Cut}_{\mathcal{F} \mathcal{D}})) = (\mathcal{E} \leftarrow (\text{Cut}_{\mathcal{F} \mathcal{D}})) \) because the negatives among \( (m/x^a) \) in \( (\text{Cut}_{\mathcal{F} \mathcal{D}}) \) will not interfere in the interaction with \( \mathcal{E} \), since the variables \( x^a \) do not appear in \( \mathcal{E} \). Hence the result.
- Or \( p \in \mathcal{E} \) [commutation step].
  In this case, we have \( s|\mathcal{E} = \kappa^+ s'|\mathcal{E} \), and by definition of the interaction sequence \( s' = \mathcal{E} \mathcal{F} \mathcal{D} \) where \( \mathcal{E} \mathcal{F} \mathcal{D} = \mathcal{E} \mathcal{D} \mathcal{F} \). Thus by induction hypothesis \( s'|\mathcal{E} = \mathcal{E} \leftarrow (\text{Cut}_{\mathcal{F} \mathcal{D}}) \). But we have

\[
(\mathcal{E} \leftarrow (\text{Cut}_{\mathcal{F} \mathcal{D}})) = (\mathcal{E} \leftarrow (\text{Cut}_{\mathcal{F} \mathcal{D}} \cup (n/x)\{p_a\})) = (\mathcal{E} \leftarrow (\text{Cut}_{\mathcal{F} \mathcal{D}} \cup (n/x)\{p_a\})) = \kappa^+ (\mathcal{E} \leftarrow (\text{Cut}_{\mathcal{F} \mathcal{D}}))
\]
Remark in particular that for all positive designs $\mathcal{D}$, we consider the observational ordering $\preceq$ over designs: $\mathcal{D}' \preceq \mathcal{D}$ if $\mathcal{D}$ can be obtained from $\mathcal{D}'$ by substituting:

- positive subdesigns for some occurrences of $\Omega$.
- $\Diamond$ for some positive subdesigns.

Remark in particular that for all positive designs $\mathcal{D}$, and if $\mathcal{D} \not\preceq \mathcal{D}'$, then $\mathcal{D}$ is positive with only positive design of the form $\mathcal{D} = \mathcal{D}' \cup \{p\}$ where $p$ is of the form $p = x \pi(m)$, and there exists a negative design $\mathcal{N}$ such that $(n/x) \in \mathcal{E} \cup \mathcal{F}$, with $n$ of the form $n = \sum_{b \in S} b(x^\rightarrow)$. $\mathcal{P}$ is such that $\mathcal{P} \not\preceq \mathcal{D}'$, and we prove:

- Either $n \in \mathcal{F}$ [reduction step].
  In this case, we have $\mathcal{P} \mathcal{E} = \mathcal{P}' | \mathcal{E}$, so let us show that $\mathcal{P}' | \mathcal{E} = (\mathcal{E} \leftarrow \{\mathcal{P}' \cup \{n/x\}\})$. By induction hypothesis $\mathcal{P}' | \mathcal{E} = (\mathcal{E} \leftarrow \{\mathcal{P}' \cup \{n/x\}\})$ where $\mathcal{P}' = (\mathcal{E} \leftarrow \{\mathcal{P}' \cup \{n/x\}\})$, hence the result.

- Or $n \in \mathcal{E}$ [commutation step].
  In this case, we have $\mathcal{P} | \mathcal{E} = \mathcal{P}' | \mathcal{E}$.
  By induction hypothesis $\mathcal{P} | \mathcal{E} = \mathcal{P}' | \mathcal{E}' = (\mathcal{E}' \leftarrow \{\mathcal{P}' \cup \{n/x\}\})$. But we have:

$$
\mathcal{E} \leftarrow \{\mathcal{P}' \cup \{n/x\}\} = \mathcal{E} \leftarrow \{\mathcal{P}' \cup \{n/x\}\}
$$

where $\mathcal{P}'$ is the only positive design of $\{\mathcal{P}' \cup \{n/x\}\}$, and for each $i \leq \alpha \tau(a)$, $m_i$ is the only negative design of $\{\mathcal{P}' \cup \{n/x\}\}$ on variable $x_i$. Therefore $\mathcal{E} \leftarrow \{\mathcal{P}' \cup \{n/x\}\} = \mathcal{E} \leftarrow \{\mathcal{P}' \cup \{n/x\}\}$, which concludes the proof.

\[\square\]

## B Proofs of Subsection 3.2

We now come back to (non “multi-“) designs, and we prove:

- the form of visitable paths for each connective (§ B.2), which is needed for next point;

- that (some) connectives preserve regularity (Propositions 84, 87, 88, corresponding to Proposition 18), purity (Proposition 19) and quasi-purity (Proposition 21).

### B.1 Preliminaries

#### B.1.1 Observational Ordering and Monotonicity

We consider the observational ordering $\preceq$ over designs: $\mathcal{D}' \preceq \mathcal{D}$ if $\mathcal{D}$ can be obtained from $\mathcal{D}'$ by substituting:

- positive subdesigns for some occurrences of $\Omega$.
- $\Diamond$ for some positive subdesigns.

Remark in particular that for all positive designs $\mathcal{P}$ and $\mathcal{P}'$, we have $\Omega \preceq \mathcal{P} \preceq \mathcal{P}'$, and if $\mathcal{P} \preceq \mathcal{P}'$ then $\mathcal{P} \preceq \mathcal{P}'$. We can now state the monotonicity theorem, an important result of ludics. A proof of the theorem formulated in this form is found in [17].

**Theorem 67 (Monotonicity).**

- If $\mathcal{D} \preceq \mathcal{E}$ and $\mathcal{M} \preceq \mathcal{N}$, then $\mathcal{D}[m/x] \preceq \mathcal{E}[n/x]$
- If $\mathcal{D} \preceq \mathcal{E}$ then $[\mathcal{D}] \preceq [\mathcal{E}]$
This means that the relation \( \preceq \) compares the likelihood of convergence: if \( \mathfrak{d} \perp \varepsilon \) and \( \mathfrak{d} \preceq \mathfrak{d}' \) then \( \mathfrak{d}' \perp \varepsilon \). In particular, if \( \mathcal{B} \) is a behaviour, if \( \mathfrak{d} \in \mathcal{B} \) and \( \mathfrak{d} \preceq \mathfrak{d}' \) then \( \mathfrak{d}' \in \mathcal{B} \).

Remark the following important fact: given a path \( s \) of some design \( \mathfrak{d} \), there is a unique design maximal for \( \preceq \) such that \( s \) is a path of it. Indeed, this design \( \pi_{\mathfrak{d}}^{\mathcal{B}} \) is obtained from \( \mathfrak{d} \) by replacing all positive subdesigns (possibly \( \Omega \)) whose first positive action is not in \( s \) by \( \triangleright \). Note that, actually, the design \( \pi_{\mathfrak{d}}^{\mathcal{B}} \) does not depend on \( \mathfrak{d} \) but only on the path \( s \).

**Example 68.** Consider design \( \mathfrak{d} \) and the path \( s \) below:

\[
\begin{align*}
\mathfrak{d} &= x | \pi(b(y).(y|\pi())|, c() \triangleright y|\pi()|, d(z).(z|\pi())| \\
\mathfrak{d}' &\sim \mathfrak{d} \\
s &= x | \pi(x_1, x_2) | b_{x_1}(y) | y|\pi()| | c_{x_2}() | d_{x_2}(y)
\end{align*}
\]

We have \( \pi_{\mathfrak{d}}^{\mathcal{B}} = x | \pi(b(y).(y|\pi())| + \sum_{f \neq b} f \triangleright x \triangleright \sum_{f \in S} f \triangleright \triangleright \).

**Proposition 69.** If \( s \in V_{\mathcal{B}} \) then \( \pi_{\mathfrak{d}}^{\mathcal{B}} \in \mathcal{B} \).

**Proof.** There exists \( \mathfrak{d} \in \mathcal{B} \) such that \( s \) is a path of \( \mathfrak{d} \), thus \( \mathfrak{d} \preceq \pi_{\mathfrak{d}}^{\mathcal{B}} \). The result then comes from monotonicity (Theorem 67).

**B.1.2 More on Paths**

Let \( \mathcal{B} \) be a behaviour.

**Lemma 70.** If \( \mathfrak{d} \in \mathcal{B} \) and \( s \in V_{\mathcal{B}} \) is a path of \( \mathfrak{d} \), then \( s \) is a path of \( \mathfrak{d} \).

**Proof.** Let \( \varepsilon \in \mathcal{B}^{-1} \) such that \( s = (\mathfrak{d} \leftarrow \varepsilon) \), and let \( t = (|\varepsilon| \leftarrow \varepsilon) \).

- Since \( |\varepsilon| \subseteq \mathfrak{d} \), the path \( s \) cannot be a strict prefix of \( t \), and \( s \) and \( t \) cannot differ on a positive action.

- If \( t \) is a strict prefix of \( s \) then it is positive-ended. So \( \tilde{s} \) and \( \tilde{t} \) are paths of \( \varepsilon \) differing on a positive action, which is impossible.

- If \( s \) and \( t \) differ on a negative action, say \( \kappa_1^- \) and \( \kappa_2^- \) are respective prefixes of \( s \) and \( t \) with \( \kappa_1^- \neq \kappa_2^- \), then \( \kappa_1^- \) and \( \kappa_2^- \) are paths of \( \varepsilon \) differing on a positive action, which is impossible.

Thus we must have \( s = t \), hence the result.

**Lemma 71.** Let \( s \in V_{\mathcal{B}} \). For every positive-ended (resp. negative-ended) prefix \( s' \) of \( s \), we have \( s' \in V_{\mathcal{B}} \) (resp. \( s' \triangleright \in V_{\mathcal{B}} \)).
Proof. Let \( s = (δ ← ε) \) where \( δ \in B \) and \( ε \in B^⊥ \), and let \( s' \) be a prefix of \( s \).

- If \( s' \) is negative-ended, let \( κ^+ \) be such that \( s'κ^+ \) is a prefix of \( s \). The action \( κ^+ \) comes from \( δ \). Consider design \( δ' \) obtained from \( δ \) by replacing the positive subdesign of \( δ \) starting on action \( κ^+ \) with \( \overline{δ} \). Since \( δ \preceq δ' \), by monotonicity \( δ' \in B \). Moreover \( s'\overline{δ} = (δ' ← ε) \), hence the result.

- If \( s' \) is positive-ended then either \( s' = s \) and there is nothing to prove or \( s' \) is a strict prefix of \( s \), so assume we are in the second case. \( s' \) is \( \overline{δ} \)-free, hence \( s' \) is a negative-ended prefix of \( s \in V_B^+ \). Using the argument above, it comes \( s' = \overline{s'} \in V_B^+ \), thus \( s' \in V_B \).

\( \square \)

**Lemma 72.** Let \( s \in V_B \). For every prefix \( s'κ^- \) of \( s \) and every \( δ \in B \) such that \( s' \) is a path of \( δ \), \( s'κ^- \) is a prefix of a path of \( δ \).

**Proof.** There exist \( δ_0 \in B \) and \( ε_0 \in B^⊥ \) such that \( s = (δ_0 ← ε_0) \). Let \( s'κ^- \) be a prefix of \( s \), and let \( δ \in B \) such that \( s' \) is a path of \( δ \). Since \( s' \) is a prefix of a path of \( ε_0 \), \( s' \) is a prefix of \( δ ← ε_0 \).

We cannot have \( s' = (δ ← ε_0) \), otherwise \( s' = \overline{s}\overline{δ} \) and \( s'κ^- \) would be paths of \( ε_0 \) differing on a positive action, which is impossible. Thus there exists \( κ^- \) such that \( s'κ^- \) is a prefix of \( (δ ← ε_0) \), which is a path of \( δ \), and necessarily \( κ^- = d^- \). Finally \( s'κ^- \) is a prefix of a path of \( δ \).

\( \square \)

### B.1.3 An Alternative Definition of Regularity

Define the **anti-shuffle** (\( \overline{\sqcup} \)) as the dual operation of shuffle, that is:

- \( s\overline{t} = \overline{s} \sqcup t \) if \( s \) and \( t \) are paths of same polarity;
- \( S\overline{T} = \overline{S} \sqcup \overline{T} \) if \( S \) and \( T \) are sets of paths of same polarity.

**Definition 73.** A **trivial view** is an aj-sequence such that each proper action except the first one is justified by the immediate previous action. In other words, it is a view such that its dual is a view as well.

- The **trivial view** of an aj-sequence is defined inductively by:

  \[
  \langle \epsilon \rangle = \epsilon \quad \text{empty sequence}
  \]

  \[
  \langle s\overline{δ} \rangle = \langle s \rangle \overline{δ} \quad \text{if } s \overline{δ} \text{ initial}
  \]

  \[
  \langle sk \rangle = \langle s \rangle k \quad \text{if } s \overline{δ} \text{ justified, where } s_0 \text{ prefix of } s \text{ ending on just}(k)
  \]

We also write \( \langle κ \rangle^t \) (or even \( \langle κ \rangle \)) instead of \( \langle s'κ \rangle \) when \( s'κ \) is a prefix of \( s \).

- **Trivial views of a design** \( δ \) are the trivial views of its paths (or of its views). In particular, \( ε \) is a trivial view of negative designs only.

- **Trivial views of designs in \( |B| \)** are called **trivial views of \( B \)**.

**Lemma 74.** 1. Every view is in the anti-shuffle of trivial views.

2. Every path is in the shuffle of views.

**Proof.**

1. Let \( v \) be a view, the result is shown by induction on \( v \):

   - If \( v = ε \) or \( v = κ \), it is itself a trivial view, hence the result.
Suppose \( \nu = \nu' \kappa \) with \( \nu' \neq \epsilon \) and \( \nu' \in \downarrow_{1|\uparrow_1} \ldots \downarrow_{n|\uparrow_n} \) where the \( \downarrow_i \) are trivial views.

- If \( \kappa \) is negative, as \( \nu \) is a view, the action \( \kappa \) is justified by the last action of \( \nu' \), say \( \kappa^- \). Hence \( \kappa^+ \) is the last action of some trivial view \( \downarrow_m \). Hence \( \nu \in \downarrow_{1|\uparrow_1} \ldots \downarrow_{m-1|\uparrow_{m-1}} \downarrow_m(\kappa^-) \downarrow_{m+1|\uparrow_{m+1}} \ldots \downarrow_n \).

- If \( \kappa \) is positive, either it is initial and \( \nu \in \downarrow_{1|\uparrow_1} \ldots \downarrow_n \kappa \) with \( \kappa \) a trivial view, or it is justified by \( \nu' \) with \( \nu' \in \downarrow \). In the last case, there exists a unique \( \uparrow_i \) such that \( \kappa^- \) appears in \( \downarrow_i \), so let \( \kappa^- \) be the prefix of \( \downarrow_i \) ending with \( \kappa^- \). We have that \( \nu \in \downarrow_{1|\uparrow_1} \ldots \downarrow_n \kappa^- \) where \( \kappa^- \) is a trivial view.

2. Similar reasoning as above, but replacing \( \downarrow \) by \( \uparrow \), “trivial view” by “view”, “view” by “path”, and exchanging the role of the polarities of actions.

\[ \blacksquare \]

**Remark.** Following previous result, note that every view (resp. path) of a design \( \mathfrak{d} \) is in the anti-shuffle of trivial views (resp. in the shuffle of views) of \( \mathfrak{d} \).

**Proposition 75.** \( \mathcal{B} \) is regular if and only if the following conditions hold:

- the positive-ended trivial views of \( \mathcal{B} \) are visitable in \( \mathcal{B} \),
- \( V_\mathcal{B} \) and \( V_\mathcal{B}^- \) are stable under \( \uplus \) (i.e., \( V_\mathcal{B} \) is stable under \( \uplus \) and \( \uplus \downarrow \)).

**Proof.** Let \( \mathcal{B} \) be a behaviour.

(\( \Rightarrow \)) Suppose \( \mathcal{B} \) is regular, and let \( \downarrow \) be a positive-ended trivial view of \( \mathcal{B} \). There exists a view \( \nu \) of a design \( \mathfrak{d} \in \mathcal{B} \) such that \( \downarrow \) is a subsequence of \( \nu \), and such that \( \nu \) ends with the same action as \( \downarrow \). Since \( \nu \) is a view of \( \mathfrak{d} \), \( \nu \) is in particular a path of \( \mathfrak{d} \), hence by regularity \( \nu \in V_\mathcal{B} \). There exists \( \epsilon \in \mathcal{B}^\downarrow \) such that \( \nu = (\mathfrak{d} \leftarrow \epsilon) \), and by Lemma 70 we can even take \( \epsilon \in \mathcal{B}_\uplus \). Since \( \nu \) is a path of \( \epsilon \), \( \nu \) is a view of \( \epsilon \). But notice that \( \nu \) is a view of \( \epsilon \) by definition of a view and of a trivial view. We deduce that \( \downarrow \) is a view (and in particular a path) of \( \epsilon \), hence \( \downarrow \in V_\mathcal{B}^- \). Now, \( \downarrow \in V_\mathcal{B}^- \).

(\( \Leftarrow \)) Assume the two conditions of the statement. Let \( s \) be a path of some design of \( \mathcal{B} \). By Lemma 74, we know that there exist views \( \nu_1, \ldots, \nu_n \) such that \( s = \nu_1 \uplus \cdots \uplus \nu_n \), and for each \( \nu_i \) there exist trivial views \( \downarrow_{i,1}, \ldots, \downarrow_{i,m_i} \) such that \( \nu_i \in \downarrow_{i,1|\uparrow_{i,1}} \ldots \downarrow_{i,m_i|\uparrow_{i,m_i}} \). By hypothesis each \( \downarrow_{i,j} \) is visitable in \( \mathcal{B} \), hence as \( V_\mathcal{B} \) is stable by anti-shuffle, \( \nu_i \in V_\mathcal{B} \). Thus as \( V_\mathcal{B} \) is stable by shuffle, \( s \in V_\mathcal{B} \). Similarly the paths of designs of \( \mathcal{B}^\uplus \) are visitable in \( \mathcal{B}^\uplus \). Hence the result. \[ \blacksquare \]

**B.2 Form of the Visitable Paths**

From internal completeness, we can make explicit the form of the visitable paths for behaviours constructed by logical connectives; such results are necessary for proving the stability of regularity and purity (§ B.3 and B.4 respectively).

We will use the notations given at the beginning of Subsection 2.2, and also the following. Given an action \( \kappa \) and a set of sequences \( V \), we write \( \kappa V \) for \( \{ \kappa s \mid s \in V \} \). Let us note \( \kappa_v = x_0 \downarrow (x) \), \( \kappa_{\downarrow} = x_0 \downarrow (x, y) \) and \( \kappa_{\uparrow} = x_0 \uplus (x, y) \) for \( i \in \{1, 2\} \).

In this section are proved the following results:

- \( V_N = \kappa_v N_N \cup \{ \kappa \} \) and \( V_P = \kappa_v P_v \cup \{ \epsilon \} \) (Proposition 78),
- \( V_{M \otimes N} = \kappa_{\downarrow} M_{\downarrow} \cup \kappa_{\uparrow} N_{\uparrow} \cup \{ \kappa \} \) (Proposition 79),
- \( V_{M \otimes N} = \kappa_{\downarrow} (M_{\downarrow} \uplus N_{\uparrow}) \cup \{ \kappa \} \) if \( M \) and \( N \) are regular (Proposition 82),
- the general form of the visitable paths of \( M \otimes N \), not as simple (Proposition 80),
- finally, the case of \( \neg \) easily deduced from \( \otimes \) (Corollaries 81 and 83).
B.2.1 Shifts

Lemma 76.

1. \((\triangleleft)(N^\perp)^\perp \subseteq \nabla(N) \cup \{\Xi\}\)
2. \(\triangleleft(N^\perp)^\perp \subseteq \nabla(N)^\perp\)

Proof. Let \(E = \nabla(N)\), and let \(F = \triangleleft(N^\perp)^\perp\). To show the lemma, we must show \(F^\perp \subseteq E \cup \{\Xi\}\) and \(F \subseteq E^\perp\).

1. Let \(q \in F^\perp\). If \(q \neq \Xi\), \(q\) is necessarily of the form \(\nabla(n)\) where \(n\) is a negative atomic design. For every design \(p \in N^\perp\), we have \(\triangleleft(x).p^x \in F\) and \(q[\triangleleft(x).p^x/x_0] \leadsto p[n/x_0]\), thus \([q[\triangleleft(x).p^x/x_0]] = (p[n/x_0]) = \Xi\) since \(q \perp \triangleleft \triangleleft(x).p^x\). We deduce \(n \in N\), hence \(q \in E\).

2. Let \(m = \triangleleft(x).p^x \in F\). For every design \(n \in N\), we have \(\nabla(n)[m/x_0] \leadsto p[n/x_0]\), thus \(\langle \nabla(n)[m/x_0] \rangle = (p[n/x_0]) = \Xi\) since \(p \in N^\perp\) and \(n \in N\). Hence \(m \in E^\perp\).

\(\square\)

Lemma 77. \(\dagger \Pi = (\dagger \Pi^\perp)^\perp\).

Proof. If we take \(N = \Pi^\perp\), Lemma 76 gives us:

1. \((\triangleleft(x).P^x)^\perp \subseteq \nabla(\Pi^\perp) \cup \{\Xi\}\) and
2. \(\triangleleft(x).P^x \subseteq \nabla(\Pi^\perp)^\perp\).

Let \(E = \nabla(\Pi^\perp)\), and let \(F = \triangleleft(x).P^x\). By definition \(\dagger \Pi = F^\perp\). From (2) we deduce \(F^\perp \subseteq E^\perp\), and from (1) \(E^\perp = (E \cup \{\Xi\})^\perp \subseteq F^\perp\). Hence \(\dagger \Pi = F^\perp = E^\perp = (\dagger \Pi^\perp)^\perp\).

\(\square\)

Proposition 78.

1. \(V_{\Pi N} = \kappa_\Pi V_{\Pi}^\perp \cup \{\Xi\}\)
2. \(V_{\Pi}^\perp = \kappa_\Pi V_{\Pi}^\perp \cup \{\epsilon\}\)

Proof.

1. (\(\subseteq\)) Let \(q \in \dagger \Pi^\perp\) and \(m \in (\dagger \Pi^\perp)^\perp\), let us show that \(\langle q \leftarrow m \rangle \in \kappa_\Pi V_{\Pi}^\perp \cup \{\Xi\}\). By Lemma 77, \(m \in \dagger \Pi^\perp\). If \(q = \Xi\) then \(\langle q \leftarrow m \rangle = \Xi\). Otherwise, by Theorem 8, \(q = \nabla(n)\) with \(n \in N\). We have \(\langle q \leftarrow m \rangle = \langle q \leftarrow |m| \rangle\) by Lemma 70, where \(|m| \in \triangleleft(x).|\langle N^\perp \rangle^\perp|\) by Theorem 8, hence \(|m|\) is of the form \(|m| = \triangleleft(x).p^x\) with \(p \in N\). By definition \(\langle q \leftarrow |m| \rangle = \kappa_\Pi \langle n^x \leftarrow p^x \rangle\), where \((n^x \leftarrow p^x) \in V_{\Pi}^\perp\).

(\(\supseteq\)) Indeed \(\Xi \in V_{\Pi N}\). Now let \(s \in \kappa_\Pi V_{\Pi}^\perp\). There exist \(n \in N\) and \(p \in N^\perp\) such that \(s = \kappa_\Pi \langle n^x \leftarrow p^x \rangle\). Note that \(\nabla(n) \in \nabla(N)\) and \(\triangleleft(x).p^x \in \triangleleft(x).|\langle N^\perp \rangle^\perp|\). By Lemma 77, \(\dagger \Pi^\perp = (\dagger \Pi^\perp)^\perp\), hence \(\nabla(n) \perp \triangleleft(x).p^x\). Moreover \(\langle \nabla(n) \leftarrow \triangleleft(x).p^x \rangle = \kappa_\Pi \langle n^x \leftarrow p^x \rangle = s\), therefore \(s \in V_{\Pi N}\).

2. By Lemma 77 and previous item, and remarking that \(V_B = \widetilde{V_B^\perp}\) for every behaviour \(B\), we have: \(V_{\Pi}^\perp = \widetilde{V_{\Pi}^\perp} = \widetilde{V_{\Pi}^\perp} = (\kappa_\Pi V_{\Pi}^\perp \cup \{\Xi\}) = \kappa_\Pi V_{\Pi}^\perp \cup \{\epsilon\} = \kappa_\Pi V_{\Pi}^\perp \cup \{\epsilon\}\).

\(\square\)
B.2.2 Plus

**Proposition 79.** \( V_{M \oplus N} = \kappa_1 V^x_M \cup \kappa_2 V^y_N \cup \{ \overline{\cdot} \} \)

*Proof.* Let \( M \oplus N = (\ell_1(M) \cup \{ \overline{\cdot} \}) \cup (\ell_2(N) \cup \{ \overline{\cdot} \}) \) be the union of behaviours \( \oplus_1 M \) and \( \oplus_2 N \), which correspond respectively to \( \downarrow M \) and \( \downarrow N \) with a different name for the first action. Moreover, \((M \oplus N)^\perp = \{ n' | n' \pi_1 \in \pi_1(x).M^\perp \} \) and \( n' \pi_2 \in \pi_2(x).N^\perp \) = \((\kappa_1 M^\perp) \cap (\kappa_2 N^\perp), \) where the behaviours \( \kappa_1 M^\perp \) and \( \kappa_2 N^\perp \) correspond to \( \tilde{\downarrow} M^\perp \) and \( \tilde{\downarrow} N^\perp \) with different names; note also that for every \( \overline{v} \in | \kappa_1 M^\perp | \) (resp. \( | \kappa_2 N^\perp | \) there exists \( \overline{v}' \in (M \oplus N)^\perp \) such that \( \overline{v} \subseteq \overline{v}' \), in other words such that \( \overline{v} = [\overline{v}' | \kappa_1 M^\perp] \) (resp. \( [\overline{v}' | \kappa_2 N^\perp] \). Therefore the proof can be conducted similarly to the one of Proposition 78(1). \( \square \)

B.2.3 Tensor and Linear Map

The following proposition is a joint work with Fouqueré and Quatrini; in [9], they prove a similar result in the framework of original Ludics.

**Proposition 80.** \( s \in V_{M \oplus N} \) if and only if the two conditions below are satisfied:

1. \( s \in \kappa_s(V^x_M \cup V^y_N) \cup \{ \overline{\cdot} \} \).

2. for all \( t \in V^x_M \cup V^y_N \), for all \( \kappa^- \) such that \( \kappa^- \overline{\cdot} = \text{a path of } s \), \( t \kappa^- \overline{\cdot} \in V^x_M \cup V^y_N \).

*Proof.* \((\Rightarrow)\) Let \( s \in V_{M \oplus N} \). If \( s = \overline{\cdot} \) then both conditions are trivial, so suppose \( s \neq \overline{\cdot} \). By internal completeness (Theorem \( 8 \)), there exist \( m \in M \), \( n \in N \) and \( x_0 \in (M \oplus N)^\perp \) such that \( s = \langle m \otimes n | x_0 \rangle \). Thus \( x_0 \) must be of the form \( n_0 = \sum_{a \in S} a(z^{a_0}).p_a \) with \( p_a \neq \Omega \) (remember that \( \bullet = [\overline{\cdot}] \), and we have \( s = \kappa_s s' \) where \( s' = \langle \{ m^x, n^y \} \leftarrow \{ p_a, n_0/x_0 \} \rangle = \langle \{ m^x, n^y \} \leftarrow \{ p_a \} s' \rangle \). Let us prove both properties:

1. By Proposition 66, \( s' \rangle m^x = \langle \{ p_a[n/y] \} \rangle \), where \( m^x \in M^\perp \). Moreover, \( \langle \{ p_a[n/y] \} \rangle \in M^\perp \), indeed: for any \( m^x \in M \), we have \( \langle \{ p_a[n/y] \} \rangle[m^x/x] = \langle \{ p_a[n/y, m^x/x] \} \rangle = \langle \{ m^x \otimes n \} n_0/x_0 \rangle = \overline{\cdot} \) using associativity and one reduction step backwards. Thus \( s' \rangle m^x \in V^x_M \).

   Likewise, \( s' \rangle n^y = \langle \{ p_a[m/x] \} \rangle \), so \( s' \rangle n^y \in V^y_N \).

   Therefore \( s' \rangle \in (V^x_M \cup V^y_N) \).

2. Now let \( t_1 \in V^x_M \), \( t_2 \in V^y_N \). Suppose \( t \in (t_1 \cup t_2) \) and \( \kappa^- \) is a negative action such that \( \kappa^- \overline{\cdot} = \text{a path of } s \). Without loss of generality, suppose moreover that the action \( \kappa^- \) comes from \( m^x \), and let us show that \( t_1 \kappa^- \overline{\cdot} \in V^x_M \).

   Let \( t' = \{ \tau_t \pi_1^{\perp x}/x, \tau_t \pi_2^{\perp y}/y \} \leftarrow \{ \tau_t \pi_2^{\perp y}/y \} \). We will show that \( t_1 \kappa^- t' \) is a prefix of \( t_1 \pi_1^{\perp x} \) and that \( t_1 \pi_1^{\perp x} \in V^x_M \), leading to the conclusion by Lemma 71. Note the following facts:

   (a) \( \pi_2 s = \pi_2 s' + \sum a(z^{a_0}).a \pi_2 \), and thus \( s' \neq \overline{\cdot} \) (otherwise a path of the form \( \kappa_s t \kappa^- \) cannot be path of \( s \)).

   (b) \( t \) is a path of the multi-design \( \{ \pi_1 t_1 \pi_2^{\perp x}/x, \pi_2 t_2 \pi_2^{\perp y}/y \} \), and \( t \) is a prefix of a path of \( s' \), since \( \kappa^- \kappa^- \pi_1^{\perp x} \) is a path of \( s' \), hence \( t \) is a prefix of \( s' \) by Proposition 62.

   (c) Since \( t \) is a \( \overline{\cdot} \)-free positive-ended prefix of \( t' \), we have that \( t \kappa^- \) is a strict prefix of \( \kappa^- t' \).

   Thus there exists a positive action \( \kappa_0^+ \) such that \( t \kappa^- t \kappa_0^+ \) is a prefix of \( \kappa^- t' \). The paths \( t \kappa^- t \kappa_0^+ \) and \( \kappa_0^+ t \kappa_0^+ \) are both paths of \( s' \), hence necessarily \( \kappa_0^+ = \kappa^- \). We deduce that \( t \kappa^- \) is a prefix of \( t' \).

   (d) The sequence \( t' \pi_1^{\perp x} \) therefore starts with \( (t \kappa^-) t_1 \pi_1^{\perp x} \).

31
(e) We have \((t\kappa^-)\mid t_1 \kappa^- = (t\mid t_1 \kappa^-)\kappa^-\) because, since \(\kappa^-\) comes from \(m^x\), it is hereditarily justified by an initial negative action of address \(x\), and thus \(\kappa^-\) appears in design \(\pi t_1 \kappa^-\).
We deduce \((t\kappa^-)\mid t_1 \kappa^- = (t\mid t_1 \kappa^-)\kappa^- = t_1 \kappa^-\).

(f) Moreover, by Proposition 66 \(t\mid t_1 \kappa^- = (t\mid t_1 \kappa^-)\kappa^-\). Hence (by d, e, f) the sequence \(t_1 \kappa^-\) is a prefix of \(t\mid t_1 \kappa^-\).
Since \(t_1 \kappa^- \in M^x\) (by Proposition 69) and \((t\mid t_1 \kappa^-)\kappa^- \in M^{x\perp}\) (by associativity, similar reasoning as item 1), we deduce \(t\mid t_1 \kappa^- \in V^n_M\). Finally \(t_1 \kappa^- \in V^n_M\) by Lemma 71.

(\(=\)) Let \(s \in \kappa_*(V^n_M \cup V^y_N) \cup \{\emptyset\}\) such that the second constraint is also satisfied. If \(s = \emptyset\) then \(s \in V^n_M \cup V^y_N\) is immediate, so suppose \(s = \kappa s'\) where \(s' \in (V^n_M \cup V^y_N)\). Consider the design \(\pi \kappa_{\pi x} s\), and note that \(\pi \kappa_{\pi x} s = \varphi(x, y), s' + \sum_{a \neq \varphi} a(z^\varphi)\emptyset\). We will show by contradiction that \(\pi \kappa_{\pi x} s \in (M \otimes N)^-\), leading to the conclusion.

Let \(m \in M\) and \(n \in N\) such that \(m \otimes n \not\equiv \pi \kappa_{\pi x} s\). By Proposition 64 and given the form of design \(\pi \kappa_{\pi x} s\), the interaction with \(m \otimes n\) is finite and the cause of divergence is necessarily the existence of a path \(t\) and an action \(\kappa^-\) such that:

1. \(t\) is a path of \(m \otimes n\),
2. \(t\kappa^-\) is a path of \(\pi \kappa_{\pi x} s\),
3. \(t\kappa^-\) is not a path of \(m \otimes n\).

Hence \(t\) is of the form \(t = \kappa t\). Choose \(m\) and \(n\) such that \(t\) is of minimal length with respect to all such pairs of designs non orthogonal to \(s\). Let \(t_1 = t\mid m^x\) and \(t_2 = t\mid n^y\), we have \(t \in \kappa_*(t_1 \cup t_2)\). Consider the design \(\pi \kappa_{\pi x} t\), and note that \(t = \varphi(x, y), t' + \sum_{a \neq \varphi} a(z^\varphi)\emptyset\).

We prove the following:

- \(\pi \kappa_{\pi x} t \in (M \otimes N)^-\): By contradiction. Let \(m' \in M\) and \(n' \in N\) such that \(m' \otimes n' \not\equiv t\).

   Again using Proposition 64, divergence occurs necessarily because there exists a path \(v\) and a negative action \(\kappa^-\) such that:

   1. \(v\) is a path of \(m' \otimes n'\),
   2. \(v\kappa^-\) is a path of \(\pi \kappa_{\pi x} t\),
   3. \(v\kappa^-\) is not a path of \(m' \otimes n'\).

   Since the views of \(v\kappa^-\) are views of \(t\), \(v\kappa^-\) is a path of \(\pi \kappa_{\pi x} t\). Thus \(m' \otimes n' \not\equiv \pi \kappa_{\pi x} t\). Moreover \(v\) is strictly shorter than \(t\), indeed: \(v\) and \(t\) are \(\emptyset\)-free, and since \(v\kappa^-\) is a path of \(t\) any action of \(v\kappa^-\) is an action of \(t\). This contradicts the fact that \(t\) is of minimum length. We deduce \(t \in (M \otimes N)^-\).

- \(\pi \kappa_{\pi x} t \in (V^n_M \cup V^y_N)^-\): We show \(t_1 \in V^n_M\), the proof of \(t_2 \in V^y_N\) being similar. Since \(t\) is a path of \(m \otimes n\) and \(t\) a path of \(\pi \kappa_{\pi x} t\), we have \(t = (m \otimes n \leftarrow \pi \kappa_{\pi x} t) = \kappa_*(\{m^x, n^y\} \leftarrow \pi \kappa_{\pi x} t)\), hence \(t' = \{m^x, n^y\} \leftarrow \pi \kappa_{\pi x} t\). Thus by Proposition 66 \(t_1 = t'\mid m^x = \{m^x \leftarrow (t'\mid n^y)\}\) and \(m^x\) because of the equality \((t'\mid n^y)\} \leftarrow \pi \kappa_{\pi x} t\) using associativity, one reduction step backwards, and the fact that \(t' \in (M \otimes N)^-\). It follows that \(t_1 \in V^n_M\).
Proposition 84. B.3 Proof of Proposition 18: Regularity and Connectives

If $s \in \mathcal{N \rightarrow P}$, thus we must show $\kappa \in \mathbb{M} \cup \{x\}$. Hence $\mathcal{N \rightarrow P}$ is regular then $\mathbb{M} \cup \{x\}$. 2. for all $t \in \mathbb{M} \cup \{x\}$, for all $\kappa^\rightarrow$ such that $\kappa \in \mathbb{M} \cup \{x\}$ is a path of $\mathcal{N \rightarrow P}$, $\kappa \in \mathbb{M} \cup \{x\}$.

B.2.4 Tensor and Linear Map, Regular Case

Corollary 83. If $\mathcal{N \rightarrow P}$ and $\mathcal{P} \rightarrow \mathbb{M}$ are regular then $\mathbb{M} \cup \{x\}$ is regular then $\mathbb{M} \cup \{x\}$.

B.3 Proof of Proposition 18: Regularity and Connectives

Corollary 81. $s \in \mathcal{N \rightarrow P}$ if and only if the two conditions below are satisfied:

1. $\exists \in \mathcal{N \rightarrow P}$, $\mathcal{P} \rightarrow \mathbb{M}$ and $\mathcal{M} \rightarrow \mathcal{N}$.

2. for all $t \in \mathbb{M} \cup \{x\}$, for all $\kappa^\rightarrow$ such that $\kappa \in \mathbb{M} \cup \{x\}$ is a path of $\mathcal{N \rightarrow P}$, $\kappa \in \mathbb{M} \cup \{x\}$.

Proposition 82. Suppose $\mathcal{N \rightarrow P}$ and $\mathcal{P} \rightarrow \mathbb{M}$ are regular. Following Proposition 80, it suffices to show that any path $s \in \kappa \cup \{x\}$ satisfies the following condition: for all $t \in \mathbb{M} \cup \{x\}$, for all negative action $\kappa^\rightarrow$ such that $\kappa \in \mathcal{N \rightarrow P}$, $\mathcal{P} \rightarrow \mathbb{M}$ and $\mathcal{M} \rightarrow \mathcal{N}$.

If $s = \mathcal{N \rightarrow P}$, there is nothing to prove, so suppose $s = \kappa \cup \{x\}$ where $\kappa^\rightarrow$ is a trivial view of $\mathcal{N \rightarrow P}$, $\mathcal{P} \rightarrow \mathbb{M}$ and $\mathcal{M} \rightarrow \mathcal{N}$. Let $t \in \mathcal{N \rightarrow P}$ and $\kappa^\rightarrow$ be such that $\kappa \in \mathcal{N \rightarrow P}$, $\mathcal{P} \rightarrow \mathbb{M}$ and $\mathcal{M} \rightarrow \mathcal{N}$. Let $s_1, s_2, t_1, t_2 \in \mathcal{N \rightarrow P}$ such that $s' \in \mathbb{M} \cup \{x\}$ and $\kappa^\rightarrow$ is an action in $s_1$, thus we must show $t_1 \kappa^\rightarrow \in \mathcal{N \rightarrow P}$. Notice that $t_1 \kappa^\rightarrow = \langle t \kappa^\rightarrow \rangle = \langle \kappa^\rightarrow \rangle \kappa^\rightarrow = \langle \kappa^\rightarrow \rangle = \langle \kappa^\rightarrow \rangle = \langle \kappa^\rightarrow \rangle = \langle \kappa^\rightarrow \rangle$. (the second equality follows from the fact that $t \kappa^\rightarrow$ is a path of $\mathcal{N \rightarrow P}$). Since $s_1 \in \mathbb{M} \cup \{x\}$, the sequence $\langle \kappa^\rightarrow \rangle = \langle t_1 \kappa^\rightarrow \rangle = \langle \kappa^\rightarrow \rangle$ is a trivial view of $\mathbb{M} \cup \{x\}$. Let $s_1 \kappa^\rightarrow$ be the prefix of $s_1$ ending with $\kappa^\rightarrow$. By Lemma 71 $s_1 \kappa^\rightarrow \in \mathbb{M} \cup \{x\}$, so $t_1 \kappa^\rightarrow \in \mathbb{M} \cup \{x\}$.

Corollary 82. Suppose $\mathcal{M} \rightarrow \mathcal{N}$ and $\mathcal{N} \rightarrow \mathcal{P}$ are regular then $\mathcal{M} \cup \{x\}$ is regular then $\mathcal{N} \rightarrow \mathcal{P}$.

Proposition 84.

1. If $\mathcal{N} \rightarrow \mathcal{P}$ is regular then $\mathcal{N} \rightarrow \mathcal{P}$ is regular.

2. If $\mathcal{P} \rightarrow \mathbb{M}$ is regular then $\mathcal{P} \rightarrow \mathbb{M}$ is regular.

Proof.

1. Following Proposition 75:

   • By internal completeness, the trivial views of $\mathcal{N}$ are of the form $\kappa \in \mathbb{V}$ where $\in \mathbb{V}$ is a trivial view of $\mathcal{N}$. Since $\mathbb{V}$ is regular $\in \mathbb{V}$. Hence by Proposition 78, $\kappa \in \mathbb{V}$.

   • Since $\mathbb{V}$ is stable by shuffle, so is $\mathbb{V} = \kappa \mathbb{V}$ where $\kappa$ is a positive action.
Proof.

Without loss of generality, we can assume that respectively. Indeed, if this is not true, appears in \(\kappa\), thus \(u\) = \(\kappa\), contradicting the fact that \(\kappa\) appears before \(\alpha\). If it is not the case, let \(\kappa\) be negative quasi-paths. If \(\kappa\) is regular, therefore so is \((\uparrow P^\bot)^\bot\).

By Lemma 77, this means that \(\uparrow P^\bot\) is regular.

\(\square\)

**Proposition 85.** If \(M\) and \(N\) are regular then \(M \oplus N\) is regular.

Proof. Similar to Proposition 84 (1), by the same remark as in proof of Proposition 79.

In order to show that \(\otimes\) preserves regularity, consider first the following definitions and lemma. We call **quasi-path** a positive-ended \(P\)-visible \(a\)-sequence. The **shuffle** \(s \uplus t\) of two negative quasi-paths \(s\) and \(t\) is the set of paths \(u\) formed with actions from \(s\) and \(t\) such that \(u|s = s\) and \(u|t = t\).

**Lemma 86.** Let \(s\) and \(t\) be negative quasi-paths. If \(s \uplus t \neq \emptyset\) then \(s\) and \(t\) are paths.

Proof. We prove the result by contradiction. Let us suppose that there exists a triple \((s, t, u)\) such that \(s\) and \(t\) are two negative quasi-paths, \(u \in s \uplus t\) is a path, and at least one of \(s\) or \(t\) does not satisfy O-visibility, say \(s\): there exists a negative action \(\kappa\) and a prefix \(s_0\kappa\) of \(s\) such that the action \(\kappa\) is justified in \(s_0\) but just(\(\kappa\)) does not appear in \(u|s_0\).

We choose the triple \((s, t, u)\) such that the length of \(u\) is minimal with respect to all such triples. Without loss of generality, we can assume that \(u\) and \(s\) are of the form \(u = u_0\kappa\) and \(s = s_0\kappa\) respectively. Indeed, if this is not true, \(u\) has a strict prefix of the form \(u_0\kappa\); in this case we can replace \((s, t, u)\) by the triple \((s_0\kappa, u_0|t, u_0\kappa\)) which satisfies all the constraints, and where the length of \(u_0\kappa\) is less or equal to the length of \(u\).

Let \(\kappa^+ = \text{just}(\kappa^-)\). \(u\) is necessarily of the form \(u = u_0\alpha^- u_0\alpha^+\kappa\) where \(\alpha^-\) justifies \(\alpha^+\) and \(\kappa^+\) appears in \(u_0\), indeed:

- \(\kappa^+\) does not appear immediately before \(\kappa^-\) in \(u\), otherwise it would also be the case in \(s\), contradicting the fact that \(\kappa^-\) is not O-visible in \(s\).
- The action \(\alpha^+\) which is immediately before \(\kappa^-\) in \(u\) is justified by an action \(\alpha^-\), and \(\kappa^+\) appears before \(\alpha^-\) in \(u\), otherwise \(\kappa^+\) would not appear in \(u|t\) and that would contradict O-visibility of \(u\).

Let us show by contradiction something that will be useful for the rest of this proof: in the path \(u\), all the actions of \(u_2\) (which cannot be initial) are justified in \(\alpha^- u_2\). If it is not the case, let \(u_1\alpha^- u_2^\beta\) be longest prefix of \(u\) such that \(\beta\) is an action of \(u_2\) justified in \(u_1\), and let \(\beta'\) be the following action (necessarily in \(\alpha^+u^+_2\)), thus \(\beta'\) is justified in \(\alpha^+ u_2\). If \(\beta'\) is positive (resp. negative) then \(\beta\) is negative (resp. positive), thus \(u_1\alpha^- u_2^\beta = u_1^\gamma\) (resp. \(u_1\alpha^- u_2^\beta\) = \(u_1^\gamma\)) where \(u_1^\gamma\) is the prefix of \(u_1\) ending on just(\(\beta\)). But then \(u_1\alpha^- u_2^\beta\) (resp. \(u_1\alpha^- u_2^\beta\)) does not contain just(\(\beta'\)): this contradicts the fact that \(u\) is a path, since O-visibility (resp. O-visibility) is not satisfied.

Now define \(u' = u_1\kappa\) and \(s' = u'|s\) and \(t' = u'|t\), and remark that:

- \(u'\) is a path, indeed, O-visibility for \(\kappa^-\) is still satisfied since \(\iota, u_1\alpha^- u_2\alpha^+\kappa^- = \iota, u_1\alpha^- \alpha^+\kappa^-\) and \(\iota, u_1\kappa\) are both contain \(\kappa^+\) in \(u|t\).
- \(s'\) and \(t'\) are quasi-paths, since \(s' = s_1\kappa\) where \(s_1\) is a prefix of \(s\) containing \(\kappa^+ = \text{just}(\kappa^-)\), and \(t' = u'|t = u_1|t\) is a prefix of \(t\).
- \(u' \in s' \uplus t'\).
**Proof.** Following Proposition 75, we will prove that the positive-ended trivial views of $M \otimes N$ are regular, positive-ended trivial views of $M \otimes N$, and that $V_{M \otimes N}$ and $V_{M \otimes N}^\perp$ are regular. 

Every trivial view of $M \otimes N$ is of the form $\kappa s$. It follows from internal completeness (incarnated form) that $\kappa s$ is a trivial view of $M \otimes N$ if and only if $s$ is a trivial view of $M$. As $M$ (resp. $N$) is regular, positive-ended trivial views of $M^\perp$ (resp. $N^\perp$) are in $V_M^\perp$ (resp. $V_N^\perp$). Thus by Proposition 82, positive-ended trivial views of $M \otimes N$ are in $V_{M \otimes N}^\perp$.

From Proposition 82, and from the fact that $\omega$ is associative and commutative, we also have that $V_{M \otimes N}$ is regular by anti-shuffle.

Let us prove that $V_{M \otimes N}$ is stable by anti-shuffle. Let $t, u \in V_{M \otimes N}$ and let $s \in \pi \alpha u$, we show that $s \in V_{M \otimes N}$ by induction on the length of $s$. Notice first that, from Proposition 82, there exist paths $t, u \in V_{M \otimes N}$ such that $t \in \kappa s(t_1 \cup t_2)$ and $\kappa = \kappa u(t_1 \cup u_2)$. In the case $s$ of length 1, either $s = \otimes \kappa$ or $s = \kappa u$, thus the result is immediate. Thus suppose $s = s' \kappa^+$ and by induction hypothesis $s' \in V_{M \otimes N}$. Hence, it follows from Proposition 82 that there exist paths $s_1 \in V_{M}$ and $s_2 \in V_{\otimes N}$ such that $s' = \kappa s_1 (s_1 \cup s_2)$. Without loss of generality, we can suppose that $\kappa^+$ is an action of $t_1$, hence of $t$. We study the different cases, proving each time either that $s \in V_{M \otimes N}$ or that the case is impossible.

- Either $\kappa^+ = \otimes \kappa$. In that case, $s_1 \kappa^+ \otimes \kappa$ is a negative quasi-path. As $s$ is a path and $s \in \kappa s(s_1 \kappa^+ \otimes \kappa)$, by Lemma 86, we have moreover that $s_1 \kappa^+ \otimes \kappa$ is a path. Notice that $\kappa s(s_1 \kappa^+) = \langle \kappa^+ \rangle_t = \langle \kappa^+ \rangle_t$. Hence $\kappa s(s_1 \kappa^+)$ is a trivial view of $M^\perp$. Let $\langle \kappa^+ \rangle_t = \langle \kappa^+ \rangle_t$. By Lemma 74, $s_1$ is a shuffle of anti-shuffles of trivial views of $M^\perp$, one of which is the trivial view $\kappa^+$. Then remark that $s_1 \kappa^+ \otimes \kappa$ is also a shuffle of anti-shuffles of trivial views of $M^\perp$, replacing $\langle \kappa^+ \rangle_t$ by $\langle \kappa^+ \rangle_t$ (note that $\langle \kappa^+ \rangle_t$ is indeed a trivial view of $M^\perp$ since $\langle \kappa^+ \rangle_t = \langle t_1 \kappa^+ \rangle_t \otimes \kappa$ where $t_1 \kappa^+$ is the prefix of $t_1$ ending with $\kappa^+$, and $t_1 \kappa^+ \in V_M$ by Lemma 71). It follows from Proposition 75 that $s_1 \kappa^+ \otimes \kappa \in V_{M \otimes N}$. Finally, as $s \in \kappa s(s_1 \kappa^+ \otimes \kappa)$ and by Proposition 82, we have $s \in V_{M \otimes N}$.

- Or $\kappa^+$ is a proper action of $t_1$, hence of $t$. Remark that $\langle s', \kappa^+ \rangle_t = \kappa s_1 \kappa^+ = \kappa s_1 \kappa^+ \otimes \kappa$, thus just $\langle \kappa^+ \rangle_t$ appears in $\langle s', \kappa^+ \rangle_t$ hence $s_1 \kappa^+ \otimes \kappa$ is a (negative) quasi-path. As $s$ is a path and as $s \in \kappa s(s_1 \kappa^+ \otimes \kappa)$, by Lemma 86 $s_1 \kappa^+ \otimes \kappa$ is a path. We already know from previous item that $s_1 \kappa^+ \otimes \kappa \in V_{M \otimes N}$. Notice that $\kappa s(s_1 \kappa^+ \otimes \kappa) = \langle \kappa^+ \rangle_t = \langle \kappa^+ \rangle_t$. Hence $\kappa s(s_1 \kappa^+ \otimes \kappa)$ is a trivial view of $M^\perp$. Let $\kappa \kappa^+ = \kappa s_1 \kappa^+ \otimes \kappa$. By Lemma 74, $s_1 \kappa^+ \otimes \kappa$ is a shuffle of anti-shuffles of trivial views of $M^\perp$, one of which is the trivial view $\kappa \kappa^+$. Remark that $s_1 \kappa^+ \otimes \kappa$ is also a shuffle of anti-shuffles of trivial views of $M^\perp$, replacing $\kappa \kappa^+$ by $\kappa \kappa^+$. By Proposition 75, $s_1 \kappa^+ \otimes \kappa \in V_{M \otimes N}$. Finally, as $s \in \kappa s(s_1 \kappa^+ \otimes \kappa)$ and by Proposition 82, we have $s \in V_{M \otimes N}$.

- Or $\kappa^+$ is a proper action of $u_1$, hence of $u$. The reasoning is similar to previous item, using $u$ and $u_1$ instead of $t$ and $t_1$ respectively.

- Or $\kappa^+$ is a proper action of $t_2$, hence of $t$. This is impossible, being given the structure of $s$: the action $\kappa_0^+$ following the negative action $\kappa^-$ in $t$ is necessarily in $t_1$ (due to the structure of a shuffle), hence the action following $\kappa^-$ in $s$ is necessarily either $\kappa_0^+$ (hence in $t_1$) or in $u$. 

Hence the triple $(s', t', u')$ satisfies all the conditions. This contradicts the minimality of $u$. 

| Proposition 87. If $M$ and $N$ are regular, then $M \otimes N$ is regular.|

| Proof. Following Proposition 75, we will prove that the positive-ended trivial views of $M \otimes N$ are visitable in $M \otimes N$, and that $V_{M \otimes N}$ and $V_{M \otimes N}^\perp$ are stable by shuffle. |
• Or $\kappa^+$ is a proper action of $u_2$, hence of $u$: this case also leads to a contradiction. We know from previous item that a positive action of $t_2$ cannot immediately follow a negative action of $t_1$ in $s$. Similarly, a positive action of $u_2$ (resp. $t_1$, $u_1$) cannot immediately follow a negative action of $u_1$ (resp. $t_2$, $u_2$) in $s$. Suppose that there exists a positive action $\kappa_0^+$ of $u_2$ (or resp. $t_2$, $u_1$, $t_1$) which follows immediately a negative action $\kappa_0^-$ of $t_1$ (or resp. $u_1$, $t_2$, $u_2$). Let $s_0\kappa_0^+ \kappa_0^-$ be the shortest prefix of $s$ satisfying such a property, say $\kappa_0^+$ is an action of $u_2$ and $\kappa_0^-$ is an action of $t_1$. Then the view $s_0\kappa_0^-$ is necessarily only made of $\kappa_0$ and of actions from $t_1$ or $u_1$, thus it does not contain $\text{just}(\kappa_0^+)$ (where $\kappa_0^+$ cannot be initial because $N$ is negative), i.e., $s$ does not satisfy $P$-visibility: contradiction.

\[\square\]

corollary 88. If $N$ and $P$ are regular, then $N \rightarrow P$ is regular.

B.4 Proofs of Propositions 19 and 21: Purity and Connectives

Proof (Proposition 19). We must prove:

• If $N$ is pure then $\downarrow N$ is pure.

• If $P$ is pure then $\uparrow P$ is pure.

• If $M$ and $N$ are pure then $M \oplus N$ is pure.

• If $M$ and $N$ are pure then $M \otimes N$ is pure.

For the shifts and plus, the result is immediate given the form of visitable paths of $\downarrow N$, $\uparrow P$ and $M \oplus N$ (Propositions 78 and 79). Let us prove the result for the tensor.

Let $s = s' \nabla \epsilon \in V_{M \otimes N}$. According to Proposition 80, either $s = \nabla \epsilon$ or there exist $s_1 \in V_M^-$ and $s_2 \in V_N^+$ such that $s = \kappa_\epsilon (s_1 \cup s_2)$. If $s = \nabla \epsilon$ then it is extensible with $\kappa_\epsilon$, so suppose $s = \kappa_\epsilon (s_1 \cup s_2)$. Without loss of generality, suppose $s_1 = s' \nabla \epsilon$. Since $M$ is pure, $s_1$ is extensible: there exists a proper positive action $\kappa^+$ such that $s'_1 \kappa^+ \in V_M^+$. Then, note that $s'_1 \kappa^+$ is a path: indeed, since $s'_1 \kappa^+$ is a path, the justification of $\kappa^+$ appears in $s'_1 \kappa^+ = s^\gamma$. Moreover $s'_1 \kappa^+ \in \kappa_\epsilon (V_M^+ \cup V_N^+)$, let us show that $s'_1 \kappa^+ \in V_{M \otimes N}$. Let $t \in V_M^+ \cup V_N^-$ and $t^\gamma$ a negative action such that $\kappa_\epsilon t^\gamma$ is a path of $s'_1 \kappa^+$. By Proposition 80 it suffices to show that $t^\gamma \nabla \epsilon \in V_M^+ \cup V_N^-$. But $s'_1 \kappa^+ \nabla \epsilon = t^\gamma \nabla \epsilon = \sigma_{t^\gamma} = \sigma_{s'_1 \kappa^+}$, therefore $\kappa_\epsilon t^\gamma$ is a path of $\sigma_{s'_1 \kappa^+}$. Since $s \in V_{M \otimes N}$, by Proposition 80 we get $t^\gamma \nabla \epsilon \in V_M^+ \cup V_N^-$. Finally $s'_1 \kappa^+ \in V_{M \otimes N}$, hence $s$ is extensible.

Proof (Proposition 21). Since $N$ and $P$ are regular, $V_{(N \rightarrow P)^+} = \kappa_\epsilon (V_N^+ \cup \widehat{V_P^+}) \cup \{\nabla \epsilon\}$ by Corollary 83. Let $s \in V_{(N \rightarrow P)^+}$, and suppose $s$ is $\nabla \epsilon$-ended, i.e., $s$ is $\nabla \epsilon$-free. We must show that either $\widehat{s}$ is extensible or $\widehat{s}$ is not well-bracketed. The path $s$ of the form $s = \kappa_\epsilon s'$ and there exist $\nabla \epsilon$-free paths $t \in V_N^-$ and $u \in V_P^+$ such that $s' = t \cup u$. We are in one of the following situations:

• Either $\widehat{u} \in V_P^+$ is not well-bracketed, hence neither is $\widehat{s}$.

• Otherwise, since $P$ is quasi-pure, $\widehat{u} = \nabla \epsilon$ is extensible, i.e., there exists a proper positive action $\kappa^+$ such that $\beta_\kappa^+ \in V_P^+$. If $\beta_\kappa^+$ is a path, then $\beta_\kappa^+ \in \nabla \epsilon$, $\widehat{s}$ is extensible: indeed, $\beta_\kappa^+ = s \nabla \epsilon \in \kappa_\epsilon (t \cup u \nabla \epsilon \nabla \epsilon)$, thus $s \nabla \epsilon \nabla \epsilon \nabla \epsilon \epsilon \epsilon \in \kappa_\epsilon(V_N^+ \cup \widehat{V_P^+})$. In the case $\beta_\kappa^+$ is not a path, this means that $\kappa^+$ is justified by an action $\kappa^-$ that does not appear in $\beta^\gamma$, thus we have something of the form:
If $\kappa^-$ comes from $\overline{s}$, and thus also $\kappa^+$, then $\overline{s}$ is not well-bracketed, indeed: since $\kappa^-$ is hereditarily justified by $\kappa^+$ and by no action from $\overline{t}$, we have:

$$\overline{s} = \kappa^- \ldots \kappa^+ \ldots \kappa^- \ldots$$

So suppose now that $\kappa^-$ comes from $\overline{u}$, thus also $\kappa^+$. We know that $\overline{\pi}$ contains $\kappa^- = \text{just}(\kappa^+_u)$, thus in particular $\overline{\pi}$ does not contain $\kappa^-$; on the contrary, we have seen that $\overline{\tau}$ contains $\kappa^-$. By definition of the view of a sequence, this necessarily means that, in $\overline{s}$, between the action $\kappa^-$ and the end of the sequence, the following happens: $\overline{\tau}$ comes across an action $\alpha^-_t$ from $\overline{t}$, justified by an action $\alpha^+_t$ also from $\overline{t}$, making the view miss at least one action $\alpha_u$ from $\overline{u}$ appearing in $\overline{\pi}$, as depicted below.

$$\overline{s} = \kappa^- \ldots \alpha^+_t \ldots \alpha^-_u \ldots$$

Since $\alpha_u$ is hereditarily justified by $\kappa_u$ and by no action from $\overline{t}$, the path $\overline{s}$ is not well-bracketed: the justifications of $\alpha_u$ and of $\alpha^-_t$ intersect.

To sum up, we have proved that in the case when $\overline{u} = \overline{w}$ is extensible, either $\overline{s}$ is extensible too or it is not well-bracketed.

Hence $\mathbf{N} \hookrightarrow \mathbf{P}$ is quasi-pure. \hfill \Box

## C Proofs of Section 4

In this section we prove:

- that the functions $\phi_\sigma^A$ are Scott-continuous (Proposition 25),
- internal completeness for particular infinite unions of behaviours (Theorem 30),
- two lemmas of Subsection 4.3 (Lemmas 32 and 37).
C.1 Proof of Proposition 25

Lemma 89. Let $E, F$ be sets of atomic negative designs and $G$ be a set of atomic positive designs.

1. $\downarrow(E^\perp) = \nabla(E)^\perp$
2. $\uparrow(G^\perp) = \{n \mid n \not\in \Delta(x \cdot G^\perp)\}^\perp$
3. $(E^\perp) \oplus (F^\perp) = (\iota_1(E) \cup \iota_2(F))^\perp$
4. $(E^\perp) \otimes (F^\perp) = \bullet(E, F)^\perp$

**Proof.** We prove (1) and (2), the other cases are very similar to (1).

1. $\nabla(E)^\perp = \{n \mid n \not\in \Delta(x \cdot (E^\perp))^\perp = (\uparrow(E)^\perp)^\perp = \downarrow(E)^\perp$
2. $\{n \mid n \not\in \Delta(x \cdot G^\perp)\}^\perp = \{\nabla(m) \mid m \in G^\perp\}^\perp = (\downarrow(G^\perp))^\perp = (\uparrow(G^\perp))^\perp$

using the definition of the orthogonal, internal completeness, and Lemma 77.

**Proof (Proposition 25).** By induction on $A$, we prove that for every $X$ and every $\sigma$ the function $\phi_A^\sigma$ is continuous. Note that $\phi_A^\sigma$ is continuous if and only if for every directed subset $P \subseteq B^+$ we have $\bigvee_{P \subseteq F} ([A]^\sigma \cdot X \rightarrow P) = [A]^\sigma \cdot X \rightarrow V_P$. The cases $A = Y \in V$ and $A = a \in S$ are trivial, and the case $A = A_1 \oplus A_2$ is very similar to the tensor, hence we only treat the two remaining cases. Let $P \subseteq B^+$ be directed.

- Suppose $A = A_1 \otimes A_2$, then $[A]^\sigma \cdot X \rightarrow P = [A_1]^\sigma \cdot X \rightarrow P \otimes [A_2]^\sigma \cdot X \rightarrow P$, with both functions $\phi_{A_1}^\sigma : P \rightarrow [A_1]^\sigma \cdot X \rightarrow P$ continuous by induction hypothesis. For any positive behaviour $P$, let us write $\phi_{AP}$ instead of $\sigma, X \rightarrow P$. We have

$$\bigvee_{P \subseteq F} [A]^\sigma = \bigcup_{P \subseteq F} ([A]^\sigma)^\perp = \bigcup_{P \subseteq F} (\bigcup_{P \subseteq F} [A_1]^\sigma \otimes [A_2]^\sigma)^\perp$$

Let us show that

$$\bigcup_{P \subseteq F} ([A_1]^\sigma \otimes [A_2]^\sigma) = \bigcup_{P \subseteq F} ([A_1]^\sigma)^\perp \cup [A_2]^\sigma)^\perp$$

By internal completeness we have $[A_1]^\sigma \otimes [A_2]^\sigma = \bullet(\bigcup_{P \subseteq F} [A_1]^\sigma, \bigcup_{P \subseteq F} [A_2]^\sigma)$ for every $P \subseteq F$. The inclusion ($\subseteq$) of $(\ast)$ is then immediate, so let us prove $(\ast)$. First, indeed, $\otimes$ belongs to the left side. Let $P', P'' \subseteq F$, let $m \in [A_1]^\sigma$ and $n \in [A_2]^\sigma$, and let us show that $m \otimes n \in [A_1]^\sigma \otimes [A_2]^\sigma$ where $P = P' \vee P''$ (note that $P \subseteq F$ since $P$ is directed). By induction hypothesis, $\phi_{A_1}^\sigma$ is continuous, thus in particular increasing; since $P' \subseteq P$, it follows that $[A_1]^\sigma \subset \phi_{A_1}^\sigma(P') \subset [A_1]^\sigma$. Similarly, $[A_2]^\sigma \subset \phi_{A_2}^\sigma(P'') \subset [A_2]^\sigma$. We get $m \otimes n \in \bullet([A_1]^\sigma, [A_2]^\sigma)$, which proves $(\ast)$. Using internal completeness, Lemma 89 and induction hypothesis, we deduce

$$(\bigcup_{P \subseteq F} ([A_1]^\sigma \otimes [A_2]^\sigma))^\perp = \bigcup_{P \subseteq F} (\bigcup_{P' \subseteq F} [A_1]^\sigma)^\perp$$

Consequently $\phi_A^\sigma$ is continuous.
• If \( A = \mu Y.A_0 \), define \( f_0 : Q \rightarrow [A_0]^{\sigma,Y \rightarrow \forall P,Y \rightarrow Q} \) and, for every \( P \in B^+ \), \( f_P : Q \rightarrow [A_0]^{\sigma,X \rightarrow \forall P,Y \rightarrow Q} \). Those functions are continuous by induction hypothesis, thus using Kleene fixed point theorem we have \( \text{lfp}(f_0) = \bigvee_{n \in \mathbb{N}} f_0^n(\mathbf{X}) \) and \( \text{lfp}(f_P) = \bigvee_{n \in \mathbb{N}} f_P^n(\mathbf{X}) \). Therefore \( \bigvee_{P \in P}(\bigvee_{n \in \mathbb{N}} f_0^n(\mathbf{X})) = \bigvee_{P \in P}(\bigvee_{n \in \mathbb{N}} f_P^n(\mathbf{X})) \).

For every \( Q \in B^+ \) the function \( g_Q : P \rightarrow f_P(Q) \) is continuous by induction hypothesis, hence \( f_0(Q) = \bigvee_{P \in P} f_P(Q) \). From this, we prove easily by induction on \( m \) that for every \( Q \in B^+ \) we have \( f_0^m(Q) = \bigvee_{P \in P} f_P^m(Q) \). Thus \( \bigvee_{Q \in P}(\bigvee_{n \in \mathbb{N}} f_0^n(\mathbf{X})) = \bigvee_{n \in \mathbb{N}} f_0^n(\mathbf{X}) \).

C.2 Proof of Theorem 30

Before proving Theorem 30 we need some lemmas. Suppose \( (A_n)_{n \in \mathbb{N}} \) is an infinite sequence of regular behaviours such that for all \( n \in \mathbb{N} \), \( |A_n| \subseteq |A_{n+1}| \); the simplicity hypothesis is not needed for now. Let us note \( A = \bigcup_{n \in \mathbb{N}} A_n \). Notice that the definition of visitable paths can harmlessly be extended to any set \( E \) of designs of same polarity, even if it is not a behaviour; the same applies to the definition of incarnation, provided that \( E \) satisfies the following: if \( \mathbf{d}, \mathbf{e}_1, \mathbf{e}_2 \in E \) are cut-free designs such that \( \mathbf{e}_1 \subseteq \mathbf{d} \) and \( \mathbf{e}_2 \subseteq \mathbf{d} \) then there exists \( \mathbf{e} \in E \) cut-free such that \( \mathbf{e} \subseteq \mathbf{e}_1 \) and \( \mathbf{e} \subseteq \mathbf{e}_2 \).

In particular, as a union of behaviours, \( A \) satisfies this condition.

Lemma 90. 1. \( \forall n \in \mathbb{N} \), \( V_{\{A_n\}} \subseteq V_{\{A_{n+1}\}} \).

2. \( V_{\bigcup_{n \in \mathbb{N}} A_n} = \bigcup_{n \in \mathbb{N}} V_{A_n} \).

3. \( \bigcup_{n \in \mathbb{N}} A_n = \bigcup_{n \in \mathbb{N}} |A_n| \).

Proof.

1. Fix \( n \) and let \( s \in V_{A_n} \). There exist \( \mathbf{d} \in |A_n| \) such that \( s \) is a path of \( \mathbf{d} \). Since \( |A_n| \subseteq |A_{n+1}| \) we have \( \mathbf{d} \in |A_{n+1}| \), thus by regularity of \( A_{n+1} \), \( s \in V_{A_{n+1}} \).

2. (\( \subseteq \)) Let \( s \in V_{A_n} \). There exist \( n \in \mathbb{N} \) and \( \mathbf{d} \in |A_n| \) such that \( s \) is a path of \( \mathbf{d} \). By regularity of \( A_n \) we have \( s \in V_{A_n} \).

(\( \supseteq \)) Let \( m \in \mathbb{N} \) and \( s \in V_{A_m} \). For all \( n \geq m \), \( V_{A_n} \subseteq V_{A_m} \) by previous item, thus \( s \in V_{A_m} \).

Hence if we take \( \mathbf{e} = \bigvee_{n \geq m} \mathbf{e}_n \), we have \( \mathbf{e} \in \mathbf{A}_n \) for all \( n \geq m \) by monotonicity. We deduce \( \mathbf{e} \in \bigcap_{n \geq m} A_n = (\bigcup_{n \geq m} A_n)^+ = (\bigcup_{n \in \mathbb{N}} A_n)^+ = \mathbf{A}^+ \). Let \( \mathbf{d} \in A_m \) such that \( s \) is a path of \( \mathbf{d} \); we have \( \mathbf{d} \in A^+ \) and \( \mathbf{e} \in \mathbf{A}^+ \), thus \( (\mathbf{d} \subseteq \mathbf{e}) \) \( s \in V_{A_n} \).

3. (\( \subseteq \)) Let \( \mathbf{d} \) be cut-free and minimal for \( \subseteq \) in \( A \). There exists \( m \in \mathbb{N} \) such that \( \mathbf{d} \in A_m \). Thus \( \mathbf{d} \) is minimal for \( \subseteq \) in \( A_m \), otherwise it would not be minimal in \( A \), hence the result.

(\( \supseteq \)) Let \( m \in \mathbb{N} \), and let \( \mathbf{d} \in |A_m| \). By hypothesis, \( \mathbf{d} \in |A_n| \) for all \( n \geq m \). Suppose \( \mathbf{d} \) is not in \( |A| \), so there exists \( \mathbf{d'} \in A \) such that \( \mathbf{d'} \subseteq \mathbf{d} \) and \( \mathbf{d'} \neq \mathbf{d} \). In this case, there exists \( n \geq m \) such that \( \mathbf{d'} \in A_n \), but this contradicts the fact that \( \mathbf{d} \in |A_n| \).

Lemma 91. \( V_{\bigcup_{n \in \mathbb{N}} A_n} = \bigvee_{(U_{n \in \mathbb{N}} A_n)^1} = \bigvee_{(U_{n \in \mathbb{N}} A_n)^1+1} \).

Proof. In this proof we use the alternative definition of regularity (Proposition 75). We prove \( V_{A} = \bigvee_{A} \), and the result will follow from the fact that for any behaviour \( B \) (in particular if \( B = A^{\bot\bot} \)) we have \( \bigvee_{B^1} = V_B \). First note that the inclusion \( V_A \subseteq \bigvee_{A} \) is immediate.

Let \( s \in V_{A_n} \) and let us show that \( \mathbf{s} \in \bigvee_{A_n} \). Let \( \mathbf{e} \in \mathbf{A}^+ \) such that \( s \) is a path of \( \mathbf{e} \). By Lemma 74 and the remark following it, \( s \) is in the shuffle of anti-shuffles of trivial views \( \mathbf{t}_1, \ldots, \mathbf{t}_k \) of \( \mathbf{A}^+ \). For every \( i \leq k \), suppose \( t_i = (\kappa_i) \); necessarily, there exists a design \( \mathbf{d}_i \in A \) such that \( \kappa_i \) occurs in \( (\mathbf{e} \leftarrow \mathbf{d}_i) \), i.e., such that \( t_i \) is a subsequence of \( (\mathbf{e} \leftarrow \mathbf{d}_i) \), otherwise \( \mathbf{e} \) would not be in the incarnation of \( \mathbf{A}^+ \) (it would not be minimal). Let \( n \) be big enough such that \( \mathbf{d}_1, \ldots, \mathbf{d}_k \in A_n \), and note that
in particular, if \( \epsilon \in A_n^{-} \). For all \( i, \tilde{u} \), \( \tilde{u} \) is a trivial view of \( |\mathcal{D}|_{A_n} \), thus it is a trivial view of \( \mathcal{D}_n \). By regularity of \( \mathcal{D}_n \) we have \( \tilde{u} \subseteq V_{\mathcal{D}_{n}} \). Since \( \tilde{u} \) is in the anti-shuffle of shuffles of \( \tilde{u}_1, \ldots, \tilde{u}_k \), we have \( \tilde{u} \in V_{\mathcal{D}_{n}} \) using regularity again. Therefore \( \tilde{u} \in V_{\mathcal{D}} \) by Lemma 90.

**Lemma 92.** \( (\bigcup_{n \in \mathbb{N}} A_n)_{+} \) and \( (\bigcup_{n \in \mathbb{N}} A_n)_{+\perp} \) are regular.

**Proof.** Let us show \( \mathcal{A}^{-} \) is regular using the equivalent definition (Proposition 75).

- Let \( \epsilon \) be a trivial view of \( \mathcal{A}^{-} \). By a similar argument as in the proof above, there exists \( n \in \mathbb{N} \) such that \( \epsilon \in V_{\mathcal{D}_{n}} \), thus \( \epsilon \in V_{\mathcal{D}_{n}} \subseteq V_{\mathcal{D}} \). By Lemma 91 \( \epsilon \in V_{\mathcal{D}_{n}} \).

- Let \( s, t \subseteq V_{\mathcal{D}_{n}} \). By Lemma 91, \( s, t \subseteq V_{\mathcal{D}} \). By Lemma 90(2), there exists \( n \in \mathbb{N} \) such that \( s, t \subseteq V_{\mathcal{D}_{n}} \), thus by regularity of \( \mathcal{D}_{n} \) we have \( \exists \pi \pi', s \pi' t \subseteq V_{\mathcal{D}_{n}} \), in other words \( s \pi' t \subseteq V_{\mathcal{D}_{n}} \). By Lemma 91 we deduce \( s \pi' t \subseteq V_{\mathcal{D}_{n}} \), hence \( V_{\mathcal{D}_{n}} \) is stable under shuffle and anti-shuffle.

Finally \( \mathcal{A}^{-} \) is regular. We deduce that \( \mathcal{A}^{-\perp} \) is regular since regularity is stable under orthogonality.

Let us introduce some more notions for next proof. An **\( \infty \)-path** (resp. **\( \infty \)-view**) is a finite or infinite sequence of actions satisfying all the conditions of the definition of path (resp. view) but the requirement of finiteness. In particular, a finite **\( \infty \)-path** (resp. **\( \infty \)-view**) is a path (resp. a view). An **\( \infty \)-path** (resp. **\( \infty \)-view**) of a design \( \mathcal{D} \) is such that any of its positive-ended prefix is a path (resp. a view) of \( \mathcal{D} \). We call **infinite chattering** a closed interaction which diverges because the computation never ends; note that infinite chattering occurs in the interaction between two atomic designs \( \mathcal{P} \) and \( \mathcal{N} \) if and only if there exists an infinite **\( \infty \)-path** \( \mathcal{S} \) of \( \mathcal{P} \) such that \( \tilde{S} \) is an **\( \infty \)-path** of \( \mathcal{N} \) (where, when \( \mathcal{S} \) is infinite, \( \tilde{S} \) is obtained from \( \mathcal{S} \) by simply reversing the polarities of all the actions). Given an infinite **\( \infty \)-path** \( \mathcal{S} \), the design \( \pi \mathcal{S} \pi^{-} \) is constructed similarly to the case when \( \mathcal{S} \) is finite (see § B.1.1).

For the proof of the theorem, suppose now that the behaviours \( (\mathcal{A}_{n}), n \in \mathbb{N} \) are simple. Remark that the second condition of simplicity implies in particular that the dual of a path in a design of a simple behaviour is a view.

**Proof (Theorem 30).** We must show that \( \mathcal{A}^{-\perp} \subseteq \mathcal{A} \) since the other inclusion is trivial. Remark the following: given designs \( \mathcal{D} \) and \( \mathcal{D}' \), if \( \mathcal{D} \subseteq \mathcal{A} \) and \( \mathcal{D} \subseteq \mathcal{D}' \) then \( \mathcal{D}' \subseteq \mathcal{A} \). Indeed, if \( \mathcal{D} \subseteq \mathcal{A} \) then there exists \( n \in \mathbb{N} \) such that \( \mathcal{D} \subseteq A_n \); if moreover \( \mathcal{D} \subseteq \mathcal{D}' \) then in particular \( \mathcal{D} \subseteq \mathcal{D}' \), and by monotonicity \( \mathcal{D}' \subseteq A_n \), hence \( \mathcal{D}' \subseteq \mathcal{A} \). Thus it is sufficient to show that \( |\mathcal{A}^{-\perp}| \subseteq \mathcal{A} \) since for every \( \mathcal{D}' \subseteq \mathcal{A}^{-\perp} \) we have \( |\mathcal{D}'| \subseteq |\mathcal{A}^{-\perp}| \) and \( |\mathcal{D}'| \subseteq |\mathcal{A}| \).

So let \( \mathcal{D} \subseteq |\mathcal{A}^{-\perp}| \) and suppose \( \mathcal{D} \notin \mathcal{A} \). First note the following: by Lemmas 91 and 92, every path \( \mathcal{S} \) of \( \mathcal{D} \) is in \( V_{\mathcal{A}^{-\perp}} = V_{\mathcal{A}} \), thus there exists \( \mathcal{D}' \subseteq |\mathcal{A}| \) containing \( \mathcal{S} \). We explore separately the possible cases, and show how they all lead to a contradiction.

If \( \mathcal{D} \) has an infinite number of maximal slices then:

- Either there exists a negative subdesign \( n = \sum_{a \in \mathcal{S}} a(x^a) \cdot \mathcal{P}_a \) of \( \mathcal{D} \) for which there is an infinity of names \( a \in \mathcal{A} \) such that \( \mathcal{P}_a \notin \Omega \). In this case, let \( \nu \) be the view of \( \mathcal{D} \) such that for every action \( \kappa^{-} \) among the first ones of \( n, \nu \) is the prefix of a view of \( \mathcal{D} \). All such sequences \( \nu \kappa^{-} \) being prefixes of paths of \( \mathcal{D} \), we deduce by regularity of \( \mathcal{A}^{-\perp} \) and using Lemma 71 that \( \nu \kappa^{-} \mathcal{D} \subseteq V_{\mathcal{A}^{-\perp}} \). Let \( \mathcal{D}' \subseteq |\mathcal{A}| \) be such that \( \nu \) is a view of \( \mathcal{D}' \). Since \( \mathcal{D}' \) is also in \( \mathcal{A}^{-\perp} \), we deduce by Lemma 72 that for every action \( \kappa^{-} \) among the first ones of \( n, \nu \) is the prefix of a view of \( \mathcal{D}' \). Thus \( \mathcal{D}' \) has an infinite number of slices: contradiction.

- Or we can find an infinite **\( \infty \)-view** \( \nu = (\kappa_{-})^{n} \kappa_{1}^{n} \kappa_{1}^{n} \kappa_{2}^{n} \kappa_{3}^{n} \kappa_{4}^{n} \ldots \) of \( \mathcal{D} \) (the first action \( \kappa_{0}^{-} \) being optional depending on the polarity of \( \mathcal{D} \)) satisfying the following: there is an infinity of \( i \in \mathbb{N} \) such that \( \kappa_{i}^{-} \) is one of the first actions of a negative subdesign \( \sum_{a \in \mathcal{S}} a(x^a) \cdot \mathcal{P}_a \) of \( \mathcal{D} \) with at least two names \( a \in \mathcal{A} \) such that \( \mathcal{P}_a \notin \Omega \). Let \( \nu \) be the prefix of \( \nu \) ending on
There is no design \( \delta' \in |A| \) containing \( v \), indeed: in this case, for all \( i \) and all negative action \( \kappa^- \) such that \( v_i \kappa^- \) is a prefix of a view of \( \delta \), \( v_i \kappa^- \) would be a prefix of a view of \( \delta' \) by Lemma 72, thus \( \delta' \) would have an infinite number of slices, which is impossible since the \( A_n \) are simple. Thus consider \( \epsilon = \exists v' \in \mathcal{V}^-_n \): since all the \( v_i \) are views of designs in \( |A| = \bigcup_{n \in \mathbb{N}} |A_n| \) and since the \( A_n \) are simple, the sequences \( \tau_i \) are views, thus \( \exists v \) is an \( \infty \)-view. Therefore an interaction between a design \( \delta' \in A \) and \( \epsilon \) necessarily eventually converges by reaching a daimon of \( \epsilon \), indeed: infinite chattering is impossible since we cannot follow \( v \) forever, and interaction cannot fail after following a finite portion of \( v \) since those finite portions \( v_i \) are in \( \mathcal{V}_A \). Hence \( \epsilon \in A^+ \). But \( \delta \not\subseteq \epsilon \), because of infinite chattering following \( v \). Contradiction.

If \( \delta \) has a finite number of maximal slices \( c_1, \ldots, c_k \), then for every \( i \leq k \) there exist an \( \infty \)-path \( s_i \) that visits all the proper actions of \( c_i \). Indeed, any (either infinite or positive-ended) sequence \( s \) of proper actions in a slice \( c \subseteq \delta \), without repetition, such that polarities alternate and the views of prefixes of \( s \) are views of \( \epsilon \), is an \( \infty \)-path:

- (Linearity) is ensured by the fact that we are in only one slice,
- (O-visibility) is satisfied since positive actions of \( \delta \), thus also of \( \epsilon \), are justified by the immediate previous negative action (a condition true in \( |A| \), thus also satisfied in \( \delta \) because all its views are views of designs in \( |A| \))
- (P-visibility) is natively satisfied by the fact that \( s \) is a promenade in the tree representing a design.

For example, \( s \) can travel in the slice \( c \) as a breadth-first search on couples of nodes \((\kappa^-, \kappa^+)\) such that \( \kappa^+ \) is just above \( \kappa^- \) in the tree, and \( \kappa^+ \) is proper. Then 2 cases:

- Either for all \( i \), there exists \( n_i \in \mathbb{N} \) and \( \delta_i \in A_{n_i} \) such that \( s_i \) is an \( \infty \)-path of \( \delta_i \). Without loss of generality we can even suppose that \( c_i \subseteq \delta_i \); if it is not the case, replace some positive subdesigns (possibly \( \Omega \)) of \( \delta_i \) by \( \mathcal{W} \) until you obtain \( \delta'_i \) such that \( c_i \subseteq \delta'_i \), and note that indeed \( \delta'_i \in A_{n_i} \) since \( \delta_i \subseteq \delta'_i \). Let \( N = \max_{1 \leq i \leq k(n_i)} \). Since \( \delta \not\subseteq A \), thus in particular \( \delta \not\subseteq A_N \), there exists \( \epsilon \in A_N^+ \) such that \( \delta \not\subseteq \epsilon \). The reason of divergence cannot be infinite chattering, otherwise there would exist an infinite \( \infty \)-path \( t \) in \( \delta \) such that \( t \) is in \( \epsilon \), and \( t \) is necessarily in a single slice of \( \delta \) (say \( c_i \)) to ensure its linearity; but in this case we would also have \( \delta_i \not\subseteq \epsilon \) where \( \delta_i \in A_{\neg}, \) impossible. Similarly, for all (finite) path \( s \) of \( \delta \), there exists \( i \) such that \( s \) is a path of \( c_i \), thus of \( \delta_i \in A_N \); this ensures that interaction between \( \delta \) and \( \epsilon \) cannot diverge after a finite number of steps either, leading to a contradiction.

- Or there is an \( i \) such that the (necessarily infinite) \( \infty \)-path \( s_i \) is in no design of \( A \). In this case, let \( \epsilon = \exists \exists \mathcal{N} \cdot \mathcal{Z} \in \mathcal{W}^+ \) (where \( \mathcal{Z} \) is a view since the \( A_n \) are simple), and with a similar argument as previously we have \( \epsilon \in A^+ \) but \( \delta \not\subseteq \epsilon \) by infinite chattering, contradiction.

\[ \square \]

C.3 Proofs of Subsection 4.3

Proof (Lemma 32). By induction on \( A \), we prove that for all \( X \in \mathcal{V} \) and \( \kappa : \text{FV}(A) \setminus \{X\} \rightarrow B^+ \) simple and regular, the induction hypothesis consisting in all the following statements holds:

1. for all simple regular behaviours \( P, P' \in B^+ \), if \( |P| \subseteq |P'| \) then \( |\phi^A_\kappa(P)| \subseteq |\phi^A_\kappa(P')| \);
2. for all \( n \in \mathbb{N} \), \( |(\phi^A_\kappa)^n(\mathcal{Z})| \subseteq |(\phi^A_\kappa)^{n+1}(\mathcal{Z})| \);
3. for all simple regular behaviour \( P \in B^+ \), \( \phi^A_\kappa(P) \) is simple and regular;
4. \( [\mu \times A]^\sigma = \bigcup_{n \in \mathbb{N}} (\phi^A_\kappa)^n(\mathcal{Z}) \).
Let us write $\sigma_P$ for $\sigma, X \mapsto P$. Note that the base cases are immediate. If $A = A_1 \oplus A_2$ or $A = A_1 \otimes A_2$ then:

1. Follows from the incarnated form of internal completeness (in Theorem 8).
2. Easy by induction on $n$, using previous item.
3. Regularity of $\phi_{\sigma}^A(P)$ comes from Proposition 18, and simplicity is easy since the structure of the designs in $[A]^{\sigma^*}$ is given by internal completeness.
4. By Corollary 26 we have $[\mu X.A]^\sigma = (\bigcup_{n \in \mathbb{N}} (\phi_{\sigma}^A)^n(\mathcal{X}))^{\perp_{\perp}}$, and by Theorem 30 we have $(\bigcup_{n \in \mathbb{N}} (\phi_{\sigma}^A)^n(\mathcal{X}))^{\perp_{\perp}} = \bigcup_{n \in \mathbb{N}} (\phi_{\sigma}^A)^n(\mathcal{X})$ since items (2) and (3) guarantee that the hypotheses of the theorem are satisfied.
5. By previous item and Lemma 90(3).

If $A = \mu Y.A_0$ then:

1. Suppose $|P| \subseteq |P'|$, where $P$ and $P'$ are simple regular. We have $|\phi_{\sigma}^A(P)| = |[\mu Y.A_0]^\sigma^*| = \bigcup_{n \in \mathbb{N}} (\phi_{\sigma}^{A_0})^n(\mathcal{X})$ by induction hypothesis (5), and similarly for $P'$. By induction on $n$, we prove that $|(\phi_{\sigma}^{A_0})^n(\mathcal{X})| \subseteq |(\phi_{\sigma}^{A_0})^n(\mathcal{X})|$.

It is immediate for $n = 0$, and the inductive case is:

$$|(\phi_{\sigma}^{A_0})^{n+1}(\mathcal{X})| = |(\phi_{\sigma}^{A_0})^n((\phi_{\sigma}^{A_0})^n(\mathcal{X}))|$$

$$\subseteq |(\phi_{\sigma}^{A_0})^n((\phi_{\sigma}^{A_0})^n(\mathcal{X}))|$$

(by induction hypotheses (1), (3) and (5))

$$= |\phi_{\sigma}^{A_0}(\phi_{\sigma}^{A_0})^n(\mathcal{X})(P)|$$

(by induction hypothesis (1) and (3))

$$\subseteq |\phi_{\sigma}^{A_0}(\phi_{\sigma}^{A_0})^n(\mathcal{X})(P')|$$

$$= |(\phi_{\sigma}^{A_0})^{n+1}(\mathcal{X})|$$

3. By induction hypotheses (2), (3) and (4) respectively, we have

- for all $n \in \mathbb{N}$, $|(\phi_{\sigma}^{A_0})^n(\mathcal{X})| \subseteq |(\phi_{\sigma}^{A_0})^{n+1}(\mathcal{X})|$, 
- for all $n \in \mathbb{N}$, $(\phi_{\sigma}^{A_0})^n(\mathcal{X})$ is simple regular,
- $[\mu Y.A_0]^\sigma = \bigcup_{n \in \mathbb{N}} (\phi_{\sigma}^{A_0})^n(\mathcal{X})$.

Consequently, by Corollary 31, $[\mu Y.A_0]^\sigma$ is simple regular.

2. 4. 5. Similar to the cases $A = A_1 \oplus A_2$ and $A = A_1 \otimes A_2$.

\[ \square \]

**Proof (Lemma 37).** By induction on $A$:

- If $A = a$ then it has basis $[a] = \mathcal{C}_a$.

- If $A = A_1 \oplus A_2$, without loss of generality suppose $A_1$ is steady, with basis $\mathcal{B}_1$. Take $\otimes_1 \mathcal{B}_1$, as a basis for $A$, where the connective $\otimes_1$ is defined like $\otimes$ with a different name of action: $\otimes_1 \mathcal{N} := \epsilon_1(\mathcal{N})^{\perp_{\perp}}$ and by internal completeness $\otimes_1 \mathcal{N} := \epsilon_1(\mathcal{N})$.  

42
\* If $A = A_1 \otimes^+ A_2$ then both $A_1$ and $A_2$ are steady, of respective base $B_1$ and $B_2$. The behaviour $B = B_1 \otimes^+ B_2$ is a basis for $A$, indeed: since $B_1$ and $B_2$ are regular, Proposition 82 gives $V_{B_1} \otimes^+ B_2 = \kappa_\sigma(V_{B_1}^\sigma \cup V_{B_2}^\sigma) \cup \{0\}$ where, by Proposition 78, $V_{B_1} = \kappa_\sigma V_{B_1}^\sigma \cup \{\epsilon\}$ for $i \in \{1, 2\}$; from this, and using internal completeness, we deduce that $B$ satisfies all the conditions.

\* Suppose $A = \mu X. A_0$, where $A_0$ is steady and has a basis $B_0$, let us show that $B_0$ is also a basis for $A$.

- By Proposition 34, $[A]^+ = \bigcup_{n \in \mathbb{N}} (\sigma_\sigma^\sigma)^n(\varnothing)$, and since $B_0$ is a basis for $A_0$ we have $B_0 \subseteq [A]^+$. Since $B_0$ is a basis for $A_0$, we have $P \subseteq \{\epsilon\}$.

- By induction hypothesis, we immediately have that for every path $s \in V_{B_0}$, there exists $t \in V_{B_0}^\text{max} \varnothing$-free extending $s$.

- By Lemma 90(2) $V_A^\varnothing = \{\emptyset\} \cup \bigcup_{n \in \mathbb{N}} \bigcup_{V_{A_0}}^\sigma \bigcup_{V_{A_0}}^\sigma$ where $\sigma_n = \sigma, t \rightarrow (\sigma_n^\sigma)^n(\varnothing)$ has a simple regular image. By induction hypothesis, for all $n \in \mathbb{N}$ $V_{B_0}^\text{max} \subseteq V_{B_0}^\text{max} \subseteq V_{A_0}^\sigma$.

\[\square\]

D Proof of Proposition 43

In this section, we prove Proposition 43, which requires first several lemmas. Let us denote the set of functional behaviours by $\mathcal{F}$, and recall that $\mathcal{D}$ stands for the set of data behaviours.

Lemma 93. Let $P \in D$, and let $Q$ be a pure regular behaviour. The behaviour $P \rightarrow^+ Q$ is pure.

Proof. By Proposition 19 it suffices to show that $(\uparrow P) \rightarrow Q$ is pure. Remark first that, by construction of data behaviours, the following assertion is satisfied in views (thus also in paths) of $(\uparrow P)$: every proper positive action is justified by the negative action preceding it.

By regularity and Corollary 83, we have $V_{(\uparrow P) \rightarrow Q} = \kappa_\sigma(V_{(\uparrow P) \rightarrow Q} \cup \{\epsilon\})$. Let $s \varnothing \in V_{(\uparrow P) \rightarrow Q}$, and we will show that it is extensible. There exist $t_1 \in V_{(\uparrow P)}$ and $t_2 \in V_Q$ such that $s \varnothing = s \varnothing t_1 \cup t_2$. In particular, $t_1$ is $\varnothing$-free and $t_2$ is $\varnothing$-ended, since $t_2 = t^\varnothing$. Since $Q$ is pure, there exists $\kappa^+$ such that $t^\varnothing \kappa^+ \in V_Q$. Let us show that $s^\varnothing \kappa^+$ is a path, i.e., that $\kappa^+$ is justified then just($\kappa^+$) appears in $\nu s$, $\nu s$, by induction on the length of $t_1$:

- If $t_1 = \epsilon$ then $s^\varnothing \kappa^+ = t^\varnothing \kappa^+$ hence it is a path.

- Suppose $t_1 = t^\varnothing \kappa^+_p \kappa^+_\bar{p}$. Since $t_1$ is $\varnothing$-free, $\kappa^+_p$ is proper. Thus $s$ is of the form $s = s_1 \kappa^+_p \kappa^+_\bar{p} s_2$, and by induction hypothesis $s_1 s_2 \kappa^+$ is a path, i.e., just($\kappa^+$) appears in $\nu s_1 s_2 \kappa^+$.

- Either $\nu s = \nu s_1 s_2$ and indeed just($\kappa^+$) also appears in $\nu s$.

- Or $\nu s$ is of the form $\nu s = \nu s_1 \kappa^+_p \kappa^+_\bar{p} s_2$ since, by the remark at the beginning of this proof, $\kappa^+_p$ is justified by $\kappa^+_\bar{p}$. This means in particular that $s_2$ start with the same positive action as $s_2$, thus we have $s_1 s_2 = t^\varnothing \kappa^+_p \kappa^+_\bar{p} s_2$. Since just($\kappa^+$) appears in $\nu s_1 s_2$ and it is an action of $s_1$, it appears in $\nu s_1$ thus also in $\nu s$.



Therefore $s^\varnothing \kappa^+$ is a path. Since $s^\varnothing \kappa^+ \in \kappa_\sigma(V_{(\uparrow P) \rightarrow Q})$ and the behaviours are regular, $s^\varnothing \kappa^+ \in V_{\rightarrow^+ Q}$, thus $s \varnothing$ is extensible. As this is true for every $\varnothing$-ended path in $V_{(\uparrow P) \rightarrow Q}$, the behaviour $(\uparrow P) \rightarrow Q$ is pure, and so is $P \rightarrow^+ Q$.

\[\square\]

Lemma 94. If $P \in F$ and $Q \in Const$ then $P \rightarrow^+ Q$ is pure.
Proof. We prove that $\langle 1P \rangle \rightarrow Q$ is pure, and the conclusion will follow from Proposition 19. Let $\kappa^+ = x_0[\pi(\overline{f})]$ where $Q = C_\kappa$, and let $s^\kappa \in V_{(1P)\rightarrow Q}$. As in the proof of Lemma 93, there exist $t_1 \in V_{1P}$ and $t_2 \in V_Q$ such that $s^\kappa t_1 = \overline{t} \in \kappa_\ast(t_1 \sqcup t_2)$ with $t_2$ $\Xi$-ended. But $V_Q = \{\Xi, \kappa^+, \}$. Hence $s^\kappa t_1$, and this path is extensible with action $\kappa^+$, indeed: $s\kappa^+$ is a path because $s\kappa^+$ is justified by $\kappa_\ast$, which is the only initial action of $s\kappa^+$ thus appearing in $t \vdash$; moreover $s\kappa^+ \in \kappa_\ast(t_1 \sqcup \kappa^+)$ where $\kappa^+ \in V_Q$, therefore $s\kappa^+ \in V_{(1P)\rightarrow Q}$.

Lemma 95. Let $P, Q \in F$. If there is $s \in V_Q$ $\Xi$-free (resp. $\Xi$-ended) and maximal, then there is $t \in V_{P\rightarrow Q}$ $\Xi$-free (resp. $\Xi$-ended) and maximal.

Proof. Suppose there exists $s \in V_Q$ $\Xi$-free (resp. $\Xi$-ended) and maximal. Since $P$ is positive and different from $\Xi$, there exists $s' \in V_{1P}$ $\Xi$-free and non-empty. Let $t' = \kappa_\ast s'\delta$, and remark that $t' = \kappa_\ast s'\delta s$. This is a path (O- and P-visibility are satisfied), it belongs to $V_{(1P)\rightarrow Q}$, it is $\Xi$-free (resp. $\Xi$-ended). Suppose it is extensible, and consider both the “$\Xi$-free” and the “$\Xi$-ended” cases:

- if $s$ and $t'$ are $\Xi$-free, then there exists a negative action $\kappa^-$ such that $t'\kappa^-\Xi \in V_{(1P)\rightarrow Q}$. But $t'\kappa^-\Xi = \kappa_\ast s'\delta s\kappa^+\Xi$, and since it belongs to $V_{(1P)\rightarrow Q} = \kappa_\ast(V_{1P} \sqcup V_{Q'} \cup \{\epsilon\})$, we necessarily have $s\kappa^-\Xi \in V_Q$ – indeed: the sequence $\overline{s}\kappa^-$ has two adjacent negative actions. This contradicts the maximality of $s$ in $V_Q$.
- if $s$ and $t'$ are $\Xi$-ended, there exists a positive action $\kappa^+$ that extends $t'$ and a contradiction arises with a similar reasoning.

Hence $t'$ is maximal in $V_{(1P)\rightarrow Q}$. Finally, $t = \kappa_\ast t'$ fulfills the requirements.

Lemma 96. For every behaviour $P \in F$, there exists $s \in V_P$ maximal and $\Xi$-free.

Proof. By induction on $P$. If $P \in D$ then take $s \in V_B$ maximal, where $B$ is a base of $P$. Use Lemma 95 in the case of $\neg^+$, and the result is easy for $\oplus^+$ and $\oplus^+$.

Lemma 97. Let $P \in F$ and let $C$ be a context. If $C[P]$ pure then $P$ pure.

Proof. We prove the contrapositive by induction on $C$. Suppose $P$ is impure.

- If $C = [ ]$ then $C[P] = P$, thus $C[P]$ is impure.
- If $C = C' \oplus^+ Q$ or $Q \oplus^+ C'$ and by induction hypothesis $C'[P]$ is impure, i.e., there exists a maximal path $s^\kappa \in V_{C'[P]}$, then one of $\kappa_\ast s^\kappa \kappa_\ast s^\kappa$ or $\kappa_\ast s^\kappa$ is maximal in $V_{C'[P]}$, hence the result.
- If $C = C' \otimes^+ Q$ or $Q \otimes^+ C'$ and by induction hypothesis there exists a maximal path $s^\kappa \in V_{C'[P]}$, then by Lemma 96, there exists a $\Xi$-free maximal path $t \in V_Q$. Consider the path $u = \kappa_\ast s^\kappa t \kappa_\ast s^\kappa$, where:
  - $\kappa_\ast$ justifies the first action of $t$,
  - $\kappa_\ast$ justifies the first one of $s$, and
  - $\kappa_\ast$ justifies $\kappa_\ast$ and $\kappa_\ast$, one on each (1st or 2nd) position, depending on the form of $C$.
We have $u \in V_{C'[P]}$, and $u$ is $\Xi$-ended and maximal, hence the result.
- If $C = Q \rightarrow^+ C'$ and by induction hypothesis $C'[P]$ is impure, then Lemma 95 (in its “$\Xi$-ended” version) concludes the proof.

Proof (Proposition 43). $(\Rightarrow)$ Suppose $P$ impure. By induction on behaviour $P$: 
• $P \in \mathcal{D}$ impossible by Corollary 40.

• If $P = P_1 \oplus^+ P_2$ (resp. $P = P_1 \ominus^+ P_2$) then one of $P_1$ or $P_2$ is impure by Proposition 19, say $P_1$. By induction hypothesis, $P_1$ is of the form $P_1 = C_1[Q_1 \rightarrow^+ Q_2] \rightarrow^+ R]$. Let $C_1 = C_1' \oplus^+ P_2$ (resp. $C_1 = C_1' \ominus^+ P_2$) and $C_2 = C_2'$, in order to get the result for $P$.

• If $P = P_1 \rightarrow^+ P_2$, then $P_2 \notin \text{Const}$ by Lemma 94, and:
  
  - If $P_2$ impure, then by induction hypothesis $P_2$ is of the form $P_2 = C_2'[Q_1 \rightarrow^+ Q_2] \rightarrow^+ R]$, and it suffices to take $C_1 = P_1 \rightarrow C_1'$ and $C_2 = C_2'$ to get the result for $P$.
  
  - If $P_2$ is pure, since it is also regular the conclusion follows from Lemma 93.

(\Rightarrow) Let $C_1, C_2$ be contexts, $Q_1, Q_2, R \in P$ with $R \notin \text{Const}$. Let $P = C_1[Q_1 \rightarrow^+ Q_2] \rightarrow^+ R]$ and $Q = C_2[Q_1 \rightarrow^+ Q_2]$. We prove that $P$ is impure.

First suppose that $P = C_2[Q_1 \rightarrow^+ Q_2] \rightarrow^+ R$, and in this case we show the result by induction on the depth of context $C_2$. The exact induction hypothesis will be: there exists a maximal $\mathfrak{X}$-ended path in $V_P$ of the form $\kappa_1 \mathfrak{s}\mathfrak{X}$ where $\mathfrak{X} \in \kappa_1[Q_R \cup \hat{V}_R]$.

• If $C_2 = \emptyset$, then $Q = Q_1 \rightarrow^+ Q_2 = \downarrow(Q_1 \rightarrow Q_2)$ and $P = Q \rightarrow^+ R = \downarrow(Q \rightarrow R)$. In order to differentiate actions $\kappa_1, \kappa_2, \kappa_3$ used to construct $Q$ from those to construct $P$, we will use superscripts. Let $\kappa_1 t_1 \in V_{Q_1}$ be $\mathfrak{X}$-free (and non-empty). Let $t_2 \in V_{Q_2}$ be a maximal $\mathfrak{X}$-free path: its existence is ensured by Lemma 96, and it has one proper positive initial action $\kappa_2^0$. Now let $t = \kappa_1 \kappa_1^0 \kappa_2 t_1 t_2 = \kappa_1 \kappa_1^0 \kappa_2 t_1 t_2$. Similarly to the path constructed in proof of Lemma 95, we have that $t$ is $\mathfrak{X}$-free, it is in $V_{(Q_1)\rightarrow Q_2}$, and it is maximal. Thus $\kappa_1^0 t \in V_Q$. Since $R \notin \text{Const}$, there exists a path of the form $\kappa^+ \kappa^\cdot \mathfrak{X} \in V_R$, and thus necessarily $\kappa^+$ justifies $\kappa^-$. Define the sequence:

$$\mathfrak{s}\mathfrak{X} = \kappa^P \kappa^P \kappa^Q \kappa^Q \kappa^Q \kappa^Q \kappa^P \kappa^Q \kappa^- t_1 \hat{t}_2 \mathfrak{X}$$

and notice the following facts:

1. $\mathfrak{s}\mathfrak{X}$ is a path: it is a linear aj-sequence. Since $\kappa^-$ is justified by $\kappa^+$, $O$- and $P$-visibility are easy to check.

2. $\mathfrak{s}\mathfrak{X} \in V_{Q \rightarrow R}$: indeed, we have $\hat{\mathfrak{s}\mathfrak{X}} \in \kappa^P(Q \cup \kappa^Q t) \cup \kappa^\cdot \mathfrak{X}$ where $\kappa^P \kappa^Q t \in V_{Q}$ and $\kappa^\cdot \mathfrak{X} \in V_R$.

3. $\mathfrak{s}\mathfrak{X}$ is maximal: Let us show that $\mathfrak{s}\mathfrak{X}$ is not extensible. First, it is not possible to extend it with an action from $Q^+$, because this would contradict the maximality of $t$ in $V_Q$. Suppose it is extensible with an action $\kappa^+$ from $R$, that is $\kappa^+ t \in V_{Q \rightarrow R}$ and $\kappa^+ \mathfrak{X} \in \kappa^P(Q \cup \kappa^\cdot \mathfrak{X})$ where $\kappa^+ \kappa^- \mathfrak{X} \in V_R$. The action $\kappa^+$ (that cannot be initial) is necessarily justified by $\kappa^-$. But $\mathfrak{X}$ contains necessarily the first negative action of $\hat{\mathfrak{X}}$, which is the only initial action in $\mathfrak{X}$, and this action is justified by $\kappa^Q$ in $s$. Therefore $\mathfrak{X}$ does not contain any action from $s$ between $\kappa^Q$ and $\hat{\mathfrak{X}}$, in particular it does not contain $\kappa^-$ just by $\kappa^+$. Thus $\kappa^+ t$ is not $P$-visible: contradiction. Hence $\mathfrak{s}\mathfrak{X}$ maximal.

Finally $\kappa^P \mathfrak{s}\mathfrak{X} \in V_P$ is not extensible, and of the required form.

• If $C_2 = Q_0 \rightarrow^+ C$, then $Q$ is of the form $Q = Q_0 \rightarrow^+ Q'$, thus previous reasoning applies.

• If $C_2 = C \ominus^+ Q_0$ or $Q_0 \oplus^+ C$, the induction hypothesis gives us the existence of a maximal path in $V_{C(Q_1 \rightarrow^+ Q_2) \rightarrow^+ R}$ of the form $\kappa^P \kappa^P \kappa^P \kappa^P \mathfrak{s}\mathfrak{X}$ where $\kappa^P \mathfrak{S} \in (\kappa^P t') \cup \hat{u}$ with $t' \in V_{C(Q_1 \rightarrow^+ Q_2)}$ and $u \in V_R$. Let $t_0 \in V_{Q_0}$ be $\mathfrak{X}$-free and maximal, using Lemma 96. Consider the following sequence:

$$\mathfrak{s}\mathfrak{X} = \kappa^P \kappa^P \kappa^P \kappa^P \kappa^P \kappa^P \kappa^P \kappa^P \kappa^P \kappa^- u \mathfrak{X}$$

where:
- $\kappa_0^0$ justifies the first action of $t_0$.
- $\kappa_1^1$ justifies the first action of $t'$ thus the first action of $t'$.
- $\kappa_0^Q$ justifies $\kappa_0^0$ and $\kappa_1^1$.
- $\kappa_1^P$ now justifies $\kappa_1^Q$.
- $\kappa_1^P$ justifies the same actions as before.

Notice that:

1. $s\overline{\kappa}$ is a path: O- and P-visibility are satisfied.
2. $s\overline{\kappa} \in V_{\mathcal{Q} \rightarrow \mathcal{R}}$: We have $\kappa_0^Q \kappa_0^0 \kappa_1^1 \kappa_1^P \in \kappa_0^Q (\kappa_0^0 V_{\mathcal{Q}_0} \cup \kappa_1^1 V_{\mathcal{C}[\mathcal{Q}_1 \rightarrow \mathcal{Q}_2]}) = V_{\mathcal{Q}}$, hence $s\overline{\kappa} \in \kappa_1^P (V_{\mathcal{Q}} \cup V_{\mathcal{R}})$.
3. $s\overline{\kappa}$ is maximal: Indeed, it cannot be extended neither by an action of $\mathcal{Q}_0^\perp$ (contradicts the maximality of $t_0$) nor by an action of $\mathcal{C}[\mathcal{Q}_1 \rightarrow \mathcal{Q}_2]^\perp$ or $\mathcal{R}$ (contradicts the maximality of $s'$).

Finally $\kappa_1^P s\overline{\kappa} \in V_P$ is a path satisfying the constraints.

- If $\mathcal{C}_2 = \mathcal{C} \oplus^+ \mathcal{Q}_0$ or $\mathcal{Q}_0 \oplus^+ \mathcal{C}$, by induction hypothesis, there exists a path of the form $\kappa_1^P \kappa_1^P \kappa_1^P \kappa_1^\perp \kappa_1^\perp$ maximal in $V_{\mathcal{C}[\mathcal{Q}_1 \rightarrow \mathcal{Q}_2] \rightarrow \mathcal{R}}$, where $\kappa_1^P \kappa_1^P \in (\kappa_1^P t') \cup \tilde{u}$ with $t' \in V_{\mathcal{C}[\mathcal{Q}_1 \rightarrow \mathcal{Q}_2]}$ and $u \in V_{\mathcal{R}}$. Reasoning as previous item, we see that for one of $i \in \{1, 2\}$ (depending on the form of context $\mathcal{C}_2$) the path $\kappa_1^P \kappa_1^P \kappa_1^P \kappa_1^\perp \kappa_1^\perp$ (where $\kappa_1^P$ now justifies $\kappa_1^Q$) is in $V_P$, maximal, and of the required form.

The result for the general case, where $\mathcal{P} = \mathcal{C}_1[ \mathcal{C}_2[\mathcal{Q}_1 \rightarrow \mathcal{Q}_2] \rightarrow \mathcal{R} ]$, finally comes from Lemma 97.

\[\square\]