

# Harmony in the Light of Computational Ludics\*

Alberto Naibo

IHPST (UMR 8590), Université Paris 1 Panthéon-Sorbonne, CNRS  
Paris, France  
alberto.naibo@univ-paris1.fr

Yuta Takahashi

Ochanomizu University  
Tokyo, Japan  
takahashi.yuta@is.ocha.ac.jp

Prawitz formulated the so-called inversion principle as one of the characteristic features of Gentzen’s intuitionistic natural deduction. In the literature on proof-theoretic semantics, this principle is often coupled with another that is called the recovery principle. By adopting the Computational Ludics framework, we reformulate these principles into one and the same condition, which we call the harmony condition. We show that this reformulation allows us to reveal two intuitive ideas standing behind these principles: the idea of “containment” present in the inversion principle, and the idea that the recovery principle is the “converse” of the inversion principle. We also formulate two other conditions in the Computational Ludics framework, and we show that each of them is equivalent to the harmony condition.

## 1 Introduction

This paper aims to study some of the characteristic features of the so-called *proof-theoretic semantics* within the framework of Computational Ludics. Generally, the main objective of proof-theoretic semantics is to explain the meaning of linguistic expressions in terms of proof-conditions rather than truth-conditions, which are typical of referentialist semantics. In particular, by taking inspiration from the Brouwer-Heyting-Kolmogorov explanation of logical connectives, proof-theoretic semantics rests on the idea that we know the meaning of a compound sentence when we know what counts as a *canonical proof* of it. And if proofs are formalised within the framework of natural deduction, then a canonical proof of a sentence  $A$  is nothing but a closed derivation ending with an introduction rule of the main connective of  $A$ .<sup>1</sup> The introduction rules play then a privileged role in fixing the meaning of a certain connective. It is in this sense that we should understand Gentzen’s remark, according to which the introduction rules of a connective represent, as it were, the “definitions” of this connective, while the elimination rules of such a connective are nothing but the “consequences” of these definitions (see [20, p. 80]).

However, according to Prawitz, the words “definition” and “consequence” are used here only in a sort of metaphorical way [17, p. 33, f.n. 1]. To assign a more precise sense to Gentzen’s remark, Prawitz formulated the so-called *inversion principle*. The idea behind this principle is that an elimination rule  $E$  of a certain connective should essentially behave as the “inverse” of the corresponding introduction rule(s), in the sense that by an application of  $E$  one simply “restores what had already been established if the major premise of the application was inferred by an application of an introduction rule” [17, p. 33].

---

\*The authors thank the anonymous reviewers for their valuable comments on the early version of this paper. Alberto Naibo’s work is partially supported by the ANR project *PROGRAMme* (ANR-17-CE38-0003-01), by the ANR-DFG project *FFIUM* (ANR-17-FRAL-0003) and by the ANR project *GoA* (ANR-20-CE27-0004). Yuta Takahashi’s work is supported by JSPS KAKENHI Grant Number JP21K12822. Part of the present work has been prepared during Yuta Takahashi’s stay at the IHPST as a postdoctoral researcher, with the support of JSPS (Japan Society for the Promotion of Science) Overseas Research Fellowship (July 2019–June 2021).

<sup>1</sup>Following [19, § 1.3], we say that a derivation in natural deduction is closed if it has no open assumption, otherwise we say that it is open.

We can explain how this idea is used by Prawitz to better specify Gentzen’s remark by means of an example. Consider the introduction ( $I$ ) and elimination ( $E$ ) rules for the implication ( $\rightarrow$ ):

$$\frac{\begin{array}{c} A \\ \vdots \\ d \\ B \end{array}}{A \rightarrow B} \rightarrow_I^n \qquad \frac{A \rightarrow B \quad A}{B} \rightarrow_E$$

If we accept that the  $\rightarrow_I$  rule defines the connective  $\rightarrow$ , in the sense that it determines the meaning of  $\rightarrow$ , then by stating  $A \rightarrow B$  we should not be allowed to deduce anything *more* than what we can already obtain from the sub-derivation  $d$ . Otherwise, the meaning of  $A \rightarrow B$  would be more informative than what is stipulated by the  $\rightarrow_I$  rule. It is in this sense that we should understand Prawitz’s characterisation of the inversion principle in terms of “containment”: the premise of an introduction rule of certain connective  $c$  already contains all of the information that is required to obtain the conclusion of the corresponding elimination rule. The application of the elimination rule is thus dispensable when its major premise is the conclusion of an introduction rule. In the case of implication, this means that

$$\frac{\begin{array}{c} A \\ \vdots \\ d_1 \\ B \end{array}}{A \rightarrow B} \rightarrow_I^n \quad \frac{\begin{array}{c} \vdots \\ d_2 \\ A \end{array}}{B} \rightarrow_E \qquad \text{can be transformed into} \qquad \frac{\begin{array}{c} \vdots \\ d_2 \\ [A] \\ \vdots \\ d_1 \\ B \end{array}}{B}$$

because the (open) derivation  $d_1$  of the premise of the  $\rightarrow_I$  rule, when combined with the derivation  $d_2$  of the minor premise of the  $\rightarrow_E$  rule, already “contains” a derivation of the conclusion of  $\rightarrow_E$  (see [17, p. 33]). It is precisely in terms of this relation of “containment” that one can make sense of Gentzen’s remark that the elimination rules are nothing but the consequences of the introduction rules of a certain connective. The proof-transformation that we have just presented corresponds to what is usually called a (local) reduction step of a *detour* (in this case, a  $\rightarrow$ -detour).<sup>2</sup> And if we work under the proofs-as-programs correspondence (i.e. the Curry-Howard correspondence), then a computational content can be assigned to this detour reduction step because it corresponds to a  $\beta$ -reduction step (preserving typing) in typed  $\lambda$ -calculus.

Several subsequent works [15, 8, 18] tried to improve Prawitz’s analysis of Gentzen’s remark by coupling the inversion principle with another principle, which we call here the *recovery principle* by following [18]. As we have seen, the inversion principle corresponds to a *no more* condition. However, if we want the  $\rightarrow_I$  rule to *completely* determine the meaning of the connective  $\rightarrow$ , then we have to impose an extra condition in addition to the one already imposed by the inversion principle. This extra condition consists in asking that by stating  $A \rightarrow B$  we should not be allowed to deduce anything *less* than what we can already obtain from the sub-derivation  $d$ . Otherwise, the meaning of  $A \rightarrow B$  would be less informative than what is stipulated by the  $\rightarrow_I$  rule. If we want to mimic Prawitz’s account of the inversion principle and formulate the recovery principle in terms of “containment”, then we should do it in a fashion which looks like the “converse” of the inversion principle itself: all of the information that is required to obtain the conclusion of an introduction rule of a certain connective  $c$  is already contained in

<sup>2</sup>When Prawitz formulated the inversion principle in his monograph on natural deduction in 1965, he was unaware of Gentzen’s unpublished works. In particular, Prawitz was not aware that in an unpublished version of his PhD thesis, which was only discovered in 2005 by Jan von Plato [16], Gentzen had already defined the detours reduction steps for the rules of intuitionistic logic. It is certainly for this reason that he considered that Gentzen’s remark on the relationship between introduction and elimination rules of natural deduction was stated in metaphorical terms (while we know today that Gentzen’s remark was probably based on some technical results similar to those later obtained by Prawitz himself).

the derivation of the major premise of the corresponding elimination rule of  $c$ . In the case of implication, this means that

$$\begin{array}{c} \vdots d \\ A \rightarrow B \end{array} \quad \text{can be transformed into} \quad \frac{\frac{\frac{\vdots d}{A \rightarrow B} \quad \frac{n}{A}}{B} \rightarrow_E}{A \rightarrow B} \rightarrow_I^n$$

The idea is that given a derivation  $d$  of the major premiss of the  $\rightarrow_E$  rule, one can extract from it everything that is required to apply the corresponding introduction rule  $\rightarrow_I$ .<sup>3</sup> As noted in [15, 11], under the proofs-as-programs correspondence, this transformation can be seen as an  $\eta$ -expansion step in typed  $\lambda$ -calculus.

When taken together, the inversion principle and the recovery principle guarantee a balance between the introduction and the elimination rules of a certain connective: the elimination rules are no more and no less informative than the introduction rules. Consequently, by borrowing a terminology introduced by Dummett, some authors say that the rules which satisfy both the inversion and the recovery principle are *harmonious* (see [8]). Moreover, as we already mentioned, asking for both the inversion and the recovery principle is a way of demanding that the meaning of a connective is completely determined by its inference rules (and more specifically, by its introduction rules). This means that to fix the meaning of such of a connective, we do not need to look for any specific context of where to fix the reference (or the denotation) of this connective. Traditionally, a linguistic expression whose meaning is independent, and thus invariant, from any referential (or denotational) context is usually identified with what we call a *logical constant*. It is for this reason that harmony is considered to play not only the role of a meaning criterion but also of a locality criterion (see [5, pp. 286–287]).

We propose here a way to clarify both of (1) the idea of “containment” present in Prawitz’s inversion principle, and of (2) the idea that the recovery principle plays the role of the “converse” of the inversion principle. These ideas are still informal, so one should provide each of them with a precise sense. For this purpose, we study the notion of harmony from the point of view of Girard’s *Ludics* [9]. More precisely, we adopt here Terui’s *Computational Ludics* [21], because its  $\lambda$ -calculus-style syntax is particularly useful for our purpose. In Section 2, we make ideas (1) and (2) more precise by reformulating the notion of harmony within Computational Ludics (see Definition 2.11). Consequently, we will *generalise* the notion of connective in Computational Ludics and consider not only the “good” (i.e. meaningful/logical) connectives satisfying our notion of harmony but also the “bad” (i.e. non-meaningful/non-logical) connectives not satisfying it.<sup>4</sup> In Section 3, we show that our notion of harmony is characterised by each of two conditions that make an essential use of the locative and interactive features proper to the Ludics approach. The first condition, which we call the *dual decomposability of connectives*, is a variant of the so-called internal completeness of connectives. The second, which we call the *dual decomposability of visitable paths*, is formulated in terms of the *regularity* of behaviours introduced by Fouqueré and Quatrini [7] in Girard’s Ludics and extended by Pavaux [13, 14] to Computational Ludics.

<sup>3</sup>The situation is more complicated for a disjunction. The particular format of its elimination rule means that to obtain  $A \vee B$  from a derivation  $\frac{\vdots d}{A \vee B}$ , one first has to apply a  $\vee_I$  rule to obtain  $A \vee B$  in each of the minor premisses of  $\vee_E$ , and then apply  $\vee_E$  itself to discharge the open assumptions present in the derivations of the two minor premisses. Thus, in contrast to the case of  $\rightarrow$ , the elimination rule is applied after the introduction rule(s) and not before. The same remark holds for the existential quantifier. One of the advantages of the Ludics framework is that we can have a homogeneous treatment of the recovery principle and we do not depend on the format of the elimination rule of the connective under analysis.

<sup>4</sup>Our generalisation allows us to take into account connectives that do not satisfy the Ludics counterparts of the inversion principle and the recovery principle (see Example 2.10 below).

## 2 Inversion and Recovery Principles in Computational Ludics

In Section 2.1, we give the basic definitions of Terui’s Computational Ludics by following [21, 3, 14]. Next, in Section 2.2, our notion of connective is introduced. Then, in Section 2.3, we will reformulate the inversion and recovery principles as the harmony condition in Computational Ludics.

### 2.1 From Derivations to Computational Designs

*Computational designs* (in short, *c-designs*) are the basic entities of Computational Ludics. They can be understood as abstract sequent derivations because the designs in Girard’s original Ludics are such entities (for Girard’s designs, see [9, 4, 12]). Consider the following procedure to extract an abstract sequent derivation from the leftmost derivation (note that the leftmost derivation contains the rule *Daimon*  $\boxtimes$ , which enables one to deduce any sequent):

$$\frac{\frac{\frac{\overline{\vdash B, C} \boxtimes}}{\vdash B \wp C} \wp}{\vdash A \oplus (B \wp C)} \oplus \quad \frac{}{\vdash D} \boxtimes}{\vdash (A \oplus (B \wp C)) \& D} \& \xrightarrow{(1)} \frac{\frac{\frac{}{(-, B \wp C, \{\{B, C\}\})} \boxtimes}}{(+, A \oplus (B \wp C), \{B \wp C\})} \boxtimes}{(-, (A \oplus (B \wp C)) \& D, \{\{A \oplus (B \wp C)\}, \{D\}\})} \boxtimes \xrightarrow{(2)} \frac{\frac{}{(-, \xi 12, \{\{1, 2\}\})} \boxtimes}{(+, \xi 1, \{2\})} \boxtimes}{(-, \xi, \{\{1\}, \{2\}\})} \boxtimes$$

In step (1), we encode the information on the rules’ applications into triples: for example, the triple  $(-, (A \oplus (B \wp C)) \& D, \{\{A \oplus (B \wp C)\}, \{D\}\})$  indicates that  $\vdash (A \oplus (B \wp C)) \& D$  is inferred from the two premises  $\vdash A \oplus (B \wp C)$  and  $\vdash D$  by a negative (i.e. reversible) rule. On the other hand, the triple  $(+, A \oplus (B \wp C), \{B \wp C\})$  indicates that  $\vdash A \oplus (B \wp C)$  is inferred from  $\vdash B \wp C$  by a positive (i.e. irreversible) rule. In step (2), we abstract the information on the positions (i.e. the *locations*) of the formulas by omitting the information on their contents. The formula  $(A \oplus (B \wp C)) \& D$  is replaced with its location  $\xi$ ; its first immediate subformula is denoted by its location  $\xi 1$ , where 1 indicates that it is the location of the first immediate subformula, and so on. The triples and the symbol  $\boxtimes$  in the rightmost derivation above can be considered to be actions performed to construct this derivation in the bottom-up way as in proof-search. These abstract sequent derivations are called *designs as desseins* in [9] and *untyped proofs* in [12]. These abstract sequent derivations can have infinitely long branches because one does not consider formulas anymore and so one can keep on decomposing a location  $\xi$  at infinity. Moreover, they can also have infinite width (i.e. infinite branching) because there are infinitary many actions that can be applied to a location  $\xi$ .

C-designs can be treated as abstract sequent derivations that are expressed in the style of generalised infinite  $\lambda$ -terms (for the precise definition of c-designs, see Definition 2.1 below). We will take a signature  $\mathcal{A} = (A, \text{ar})$  which is a pair of a set  $A$  of names and a mapping  $\text{ar}$  that assigns an arity to each name  $a \in A$ . Then, we consider positive actions and negative actions corresponding to positive triples  $(+, \xi, I)$  and negative triples  $(-, \xi', \mathcal{N})$ , respectively. In contrast to Girard’s designs, negative actions in c-designs include variable binding to obtain a generalisation of  $\lambda$ -abstraction. However, the notion of c-design preserves a fundamental feature of Girard’s Ludics: the absence of any essential distinction between syntactic and semantic level, i.e. between derivations and models (see e.g. [21, 3]). As we have just seen, c-designs are abstract sequent derivations possibly with infinite branches and infinite widths. They are thus suitable to work not only as derivations but also (counter-)models, as far as one considers infinite trees in extracting models from proof-search failure.<sup>5</sup>

<sup>5</sup>Note that the opposition between finite derivations and infinite (counter-)models disappears when logical systems satisfying the finite model property are considered. This is what happens, for instance, in the case of multiplicative additive linear logic **MALL**, as remarked in [3, p. 2].



c-design resulting from the simultaneous substitution of the negative c-design  $N_i$  for all occurrences of  $x_i$  in  $T$  for each  $i$ , where bound variables in  $T$  are renamed if necessary.

The  $\lambda$ -term-style syntax of c-designs enjoys some computational features which are similar to the ones of  $\lambda$ -calculus. We can define a notion of redex as a sort of  $\beta$ -redex called a *cut*, and the execution of a redex is defined as the cut reduction: a c-design  $T$  is a *cut* if  $T$  is a positive c-design of the form  $(\sum a(\vec{x}_a).P_a)|\bar{a}\langle N_1, \dots, N_k \rangle$ . The reduction rule for cuts is defined as  $(\sum a(\vec{x}_a).P_a)|\bar{a}\langle \vec{N} \rangle \longrightarrow P_a[\vec{N}/\vec{x}_a]$ . A c-design is *cut-free* if it has no cut. Note that, as in the case of  $\lambda$ -terms, a c-design can be treated both as a function and as a value. The known fact below (Theorem 2.3) shows that we have a limited form of the confluence of cut reduction.

In addition, we use the following notions concerning the classification of c-designs. A variable  $x$  occurring as  $N_0|\bar{a}\langle N_1, \dots, x, \dots, N_n \rangle$  in a c-design  $T$  is called an *identity* in  $T$ . A c-design  $T$  is *identity-free* if  $T$  is not a variable and there is no identity in  $T$ . Intuitively, an identity in a c-design  $T$  indicates that  $T$  can be “ $\eta$ -expanded” at the position of  $x$  (for a detailed explanation of the notion of identity, see [21, § 2.1]). A c-design  $T$  is *total* if  $T \neq \Omega$  holds. A c-design  $T$  is *linear* if for any of its subdesigns of the form  $N_0|\bar{a}\langle N_1, \dots, N_k \rangle$ , the sets  $\text{fv}(N_0), \dots, \text{fv}(N_k)$  are pairwise disjoint. A c-design  $T$  is *standard* if  $T$  is cut-free, identity-free, total, linear and  $\text{fv}(T)$  is finite.

We denote the reflexive and transitive closure of the reduction relation  $\longrightarrow$  by  $\longrightarrow^*$ , and write  $P \Downarrow Q$  if there is a c-design  $Q$  such that  $P \longrightarrow^* Q$  holds and  $Q$  is neither a cut nor  $\Omega$ , otherwise we write  $P \Uparrow$ . To define the normal form function on c-designs, we use the head normal form function  $hnf$  and corecursion. Let  $hnf : \mathcal{D} \rightarrow \mathcal{D}$  be the function preserving the polarity such that  $hnf(N) = N$  for any negative c-design  $N$ , and if  $P \Downarrow Q$  then  $hnf(P) = Q$ , otherwise  $hnf(P) = \Omega$ . By using the corecursive definition principle of functions on c-designs (for the proof of this principle, see [21, § 2.2]), we define the normal form function on c-designs as follows.

**Definition 2.2** (Normal Form Function on C-Designs). *The normal form function  $[[\cdot]] : \mathcal{D} \rightarrow \mathcal{D}$  on c-designs are defined as follows:*

$$\begin{aligned} [[P]] &= \mathfrak{X}, \text{ if } P \Downarrow \mathfrak{X}, \\ &= \Omega, \text{ if } P \Uparrow, \\ &= x|\bar{a}\langle [[N_1]], \dots, [[N_k]] \rangle, \text{ if } P \Downarrow x|\bar{a}\langle N_1, \dots, N_k \rangle, \\ [[N]] &= x, \text{ if } N = x, \\ &= \sum a(\vec{x}_a).[[P_a]], \text{ if } N = \sum a(\vec{x}_a).P_a. \end{aligned}$$

When  $[[T]] = \mathfrak{X}$  holds, we say that  $T$  *converges to*  $\mathfrak{X}$ . We have the following limited version of confluence (for its proof, see [3, Theorem 1.12]). It is limited in the sense that it implies the joinability only for the values of the normal form function: for example, we have  $[[[[T]][N/x]]] = [[T[[N]/x]]]$ , which says that  $[[T]][N/x]$  and  $T[[N]/x]$  are joinable with respect to their values of  $[[\cdot]]$ .

**Theorem 2.3** (Associativity). *For any c-design  $T$  and any negative c-designs  $N_1, \dots, N_n$ , we have*

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

## 2.2 The Computational Behaviour of C-Designs

Since c-designs are untyped objects, we cannot classify them with respect to their computational behavior in advance (i.e. a priori), but we can do it a posteriori by *testing* them with other c-designs. For this purpose, we first define *anti-designs*. Let  $x_0$  be an arbitrary but fixed variable. A positive c-design  $P$  is *atomic* if  $\text{fv}(P) \subseteq \{x_0\}$  holds. A negative c-design  $N$  is *atomic* if  $\text{fv}(N)$  is empty.

**Definition 2.4** (Anti-Designs). (1) An anti-design against positives is a finite set  $\{(x_1, N_1), \dots, (x_n, N_n)\}$  of pairs of a variable  $x_i$  and an atomic negative c-design  $N_i$  such that  $x_1, \dots, x_n$  are pairwise distinct. We say that  $\{x_1, \dots, x_n\}$  is the base of this anti-design. (2) An anti-design against negatives is a finite set  $\{P, (x_1, N_1), \dots, (x_n, N_n)\}$  such that  $P$  is an atomic positive c-design and  $\{(x_1, N_1), \dots, (x_n, N_n)\}$  is an anti-design against positives. We say that  $\{x_1, \dots, x_n\}$  is the base of this anti-design. Later on, we denote  $\{(x_1, N_1), \dots, (x_n, N_n)\}$  by  $[N_1/x_1, \dots, N_n/x_n]$ , and  $\{P, (x_1, N_1), \dots, (x_n, N_n)\}$  by  $[P, N_1/x_1, \dots, N_n/x_n]$ .

The orthogonality, which we define below, provides a way for testing c-designs with other c-designs: if a c-design  $T$  is orthogonal to an anti-design against it, this means that  $T$  passes the test in terms of this anti-design.

**Definition 2.5** (Orthogonality). (1) A positive c-design  $P$  and an anti-design  $[G] = [N_1/x_1, \dots, N_n/x_n]$  against positives are orthogonal if and only if  $P[N_1/x_1, \dots, N_n/x_n]$  is closed and converges to  $\boxtimes$ . (2) A negative c-design  $M$  and an anti-design  $[G] = [P, N_1/x_1, \dots, N_n/x_n]$  against negatives are orthogonal if and only if  $P[M[N_1/x_1, \dots, N_n/x_n]/x_0]$  is closed and converges to  $\boxtimes$ . When a c-design  $T$  and an anti-design  $[G]$  are orthogonal, we write  $T \perp [G]$ .

An anti-design is *cut-free* (resp. *standard*) if any c-design contained in it is cut-free (resp. standard). When  $\mathbf{T}$  is a set of cut-free c-designs of the same polarity and  $\mathbf{G}$  is a set of cut-free anti-designs of the same polarity, then we define as follows:

- If all c-designs in  $\mathbf{T}$  are atomic,  $\mathbf{T}^\perp$  is the set of all standard and atomic c-designs  $U$  with  $T \perp U$  for any  $T \in \mathbf{T}$ , where for any atomic  $P$  and  $N$ ,  $P \perp N : \Leftrightarrow N \perp P : \Leftrightarrow P[N/x_0]$  is closed and converges to  $\boxtimes$ . Otherwise,  $\mathbf{T}^\perp := \{[G] : [G] \text{ is standard and } T \perp [G] \text{ holds for any } T \in \mathbf{T}\}$ .
- $\mathbf{G}^\perp := \{T : T \text{ is standard and } T \perp [G] \text{ holds for any } [G] \in \mathbf{G}\}$ .

The reason why we restrict the elements of  $\mathbf{T}^\perp$  and  $\mathbf{G}^\perp$  to cut-free ones, in particular, standard ones is that we define *behaviours* as sets of standard c-designs. Note that one may define behaviours as sets of l-designs, following [21]: an *l-design* is an identity-free, total and linear c-design with finitely many free variables. By defining behaviour as sets of standard c-designs, i.e. sets of cut-free l-designs as in [2, 3], one can simplify some formulations concerning behaviours and we have adopted this approach. In our case, we can make the characterisations of the harmony condition in Section 3 simpler because a positive c-design in a behaviour is always of the form  $x|\bar{a}(N_1, \dots, N_k)$  due to its cut-freeness.

**Definition 2.6** (Behaviours). A set  $\mathbf{T}$  of standard c-designs of the same polarity is a behaviour if and only if  $\mathbf{T} = \mathbf{T}^{\perp\perp}$  holds. We say a behaviour  $\mathbf{T}$  is an a-behaviour if all c-designs in  $\mathbf{T}$  are atomic.

Behaviours correspond to types inhabited by c-designs, and they give a classification of c-designs (in fact, standard c-designs) in terms of the orthogonality or tests: a standard c-design  $T$  belongs to a behaviour  $\mathbf{B}$  iff  $T \perp [G]$  holds for any  $[G] \in \mathbf{B}^\perp$ . Note that if  $\mathbf{B}$  is an a-behaviour, then  $\mathbf{B}^\perp$  is also an a-behaviour.

Since behaviours correspond to types, a connective in Computational Ludics applies to behaviours, and returns a new behaviour. Our notion of connective is defined as follows:

**Definition 2.7** (Connectives). An  $n$ -ary connective  $\alpha$  is a triple  $(\vec{z}, \alpha^I, \alpha^E)$  of a finite sequence  $\vec{z}$  of variables and two finite sets of negative actions  $\alpha^I, \alpha^E$  satisfying the following three conditions.

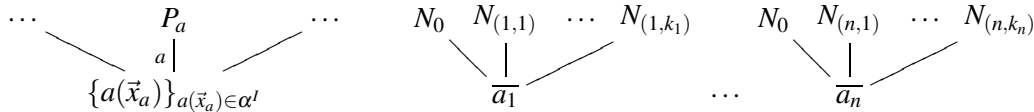
- The finite sequence  $\vec{z}$  consists of  $n$  distinct variables  $z_1, \dots, z_n$  with  $x_0 \notin \{z_1, \dots, z_n\}$ .
- The union  $\alpha^I \cup \alpha^E$  is a set  $\{a_1(\vec{x}_1), \dots, a_m(\vec{x}_m)\}$  of negative actions such that  $a_1, \dots, a_m$  are pairwise distinct names and for each  $i$  with  $1 \leq i \leq m$ , there is a natural number  $k$  and indices  $(i, 1), \dots, (i, k)$  with  $\{\vec{x}_i\} = \{x_{(i,1)}, \dots, x_{(i,k)}\} \subseteq \{\vec{z}\}$ . We denote  $\{x_{(i,1)}, \dots, x_{(i,k)}\}$  by  $X_i$ .

- The union  $\bigcup_{1 \leq i \leq m} X_i$  is equal to  $\{\vec{z}\}$ .

We stipulate that each variable in  $\vec{z}$  is bound in a connective  $(\vec{z}, \alpha^I, \alpha^E)$ .

As c-designs, two  $\alpha$ -equivalent connectives are identified (e.g.  $(x, y, \{a(x)\}, \{b(y, x)\})$  is identified with  $(v, z, \{a(v)\}, \{b(z, v)\})$ ). The reason why we imposed the condition  $x_0 \notin \{z_1, \dots, z_n\}$  is that, informally speaking, we want to keep  $x_0$  to be the location of atomic c-designs. This is useful for several of the formulations that follow; in particular, the formulation of game-semantic framework of [13, 14] in Section 3.2.

A connective  $\alpha = (\vec{z}, \alpha^I, \alpha^E)$  gives the abstract information for obtaining a set of introduction and elimination rules. In general,  $\alpha^I$  in a connective  $\alpha = (\vec{z}, \alpha^I, \alpha^E)$  determines the rule for  $\alpha$  which constructs a c-design from an  $\alpha^I$ -indexed family of positive c-designs as in the leftmost tree below. On the other hand, the set  $\alpha^E$  determines  $|\alpha^E|$  rules, where  $|\alpha^E|$  denotes the cardinality of the set  $\alpha^E$ , via the positive actions  $\bar{a}_1, \dots, \bar{a}_n$  corresponding to the negative actions in  $\alpha^E = \{a_1(\vec{x}_1), \dots, a_n(\vec{x}_n)\}$ . These rules are expressed as the remaining trees below:



We treat the leftmost rule as the introduction rule for  $\alpha$  and the other rules as the elimination rules for  $\alpha$  because the c-design  $(\sum_{\alpha^I} a(\vec{x}_a).P_a) | \bar{a}_i \langle N_{(i,1)}, \dots, N_{(i,k_i)} \rangle$  with  $a_i(\vec{x}_i) \in \alpha^E$  is a cut; namely, a redex with respect to the reduction  $\rightarrow$ , which is a generalisation of  $\beta$ -reduction.

In addition, the finite sequence  $\vec{z}$  in  $\alpha$  fixes the arity of the connective  $\alpha$ . To see this, we define the *semantic entailment* (introduced in [3, Definition 2.7]) and *behaviours composed by connectives* (a variant of the kind of behaviours defined in [21, Definition 4.11]). These behaviours are also crucial for our reformulation of harmony. A *positive context* is a finite set  $\{x_1 : \mathbf{P}_1, \dots, x_n : \mathbf{P}_n\}$  of pairs of a variable  $x_i$  and a positive a-behaviour  $\mathbf{P}_i$  such that  $x_1, \dots, x_n$  are pairwise distinct. A *negative context* is a finite set  $\{\mathbf{N}\} \cup \Gamma$  such that  $\mathbf{N}$  is a negative a-behaviour and  $\Gamma$  is a positive context.

**Definition 2.8** (Semantic Entailment). (1) Let  $P$  be a positive standard c-design with  $\text{fv}(P) \subseteq \{x_1, \dots, x_n\}$  and  $\{x_1 : \mathbf{P}_1, \dots, x_n : \mathbf{P}_n\}$  be a positive context. The entailment relation  $P \models x_1 : \mathbf{P}_1, \dots, x_n : \mathbf{P}_n$  holds if and only if for any  $M_1 \in \mathbf{P}_1^\perp, \dots, M_n \in \mathbf{P}_n^\perp$ ,  $P[M_1/x_1, \dots, M_n/x_n]$  converges to  $\boxtimes$ . (2) Let  $N$  be a negative standard c-design with  $\text{fv}(N) \subseteq \{x_1, \dots, x_n\}$  and  $\{x_1 : \mathbf{P}_1, \dots, x_n : \mathbf{P}_n, \mathbf{N}\}$  be a negative context. The entailment relation  $N \models x_1 : \mathbf{P}_1, \dots, x_n : \mathbf{P}_n, \mathbf{N}$  holds if and only if for any  $M_1 \in \mathbf{P}_1^\perp, \dots, M_n \in \mathbf{P}_n^\perp$  and any  $Q \in \mathbf{N}^\perp$ ,  $Q[N[M_1/x_1, \dots, M_n/x_n]/x_0]$  converges to  $\boxtimes$ .

**Definition 2.9** (Behaviours Composed by Connectives). For any name  $a \in A$  with  $\text{ar}(a) = n$  and any negative a-behaviours  $\mathbf{N}_1, \dots, \mathbf{N}_n$ , we define the set  $\bar{a} \langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle$  of negative c-designs as the set of all c-designs of the form  $x_0 | \bar{a} \langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle$  such that  $N_i \in \mathbf{N}_i$  holds for any  $i$  with  $1 \leq i \leq n$ .

Let  $\alpha$  be an arbitrary  $n$ -ary connective. For any positive a-behaviours  $\mathbf{P}_1, \dots, \mathbf{P}_n$  and any negative a-behaviours  $\mathbf{N}_1, \dots, \mathbf{N}_n$ , we define the positive a-behaviour  $\alpha^E \langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle$  and the negative a-behaviour  $\alpha^I \langle \mathbf{P}_1, \dots, \mathbf{P}_n \rangle$  as follows:

- $\alpha^E \langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle := (\bigcup_{a_i(\vec{x}_i) \in \alpha^E} \bar{a}_i \langle \mathbf{N}_{(i,1)}, \dots, \mathbf{N}_{(i,k_i)} \rangle)^{\perp\perp}$ , and
- $\alpha^I \langle \mathbf{P}_1, \dots, \mathbf{P}_n \rangle := \bigcap_{a_i(\vec{x}_i) \in \alpha^I} (\bar{a}_i \langle \mathbf{P}_{(i,1)}^\perp, \dots, \mathbf{P}_{(i,k_i)}^\perp \rangle^\perp)$ .

Note that we use  $\alpha^E$  and  $\alpha^I$  to define  $\alpha^E \langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle$  and  $\alpha^I \langle \mathbf{P}_1, \dots, \mathbf{P}_n \rangle$ , respectively. This is the main difference between the definition above and [21, Definition 4.11].



**Example 2.10.** Consider a ternary connective  $\gamma = (x_1, x_2, x_3, \{a(x_1, x_2), b(x_3)\}, \{c(x_1), d(x_2, x_3)\})$ . Moreover, let  $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$  be arbitrary positive  $a$ -behaviours. Then, if  $P \models \Gamma, x_1 : \mathbf{P}_1, x_2 : \mathbf{P}_2$  and  $Q \models \Gamma, x_3 : \mathbf{P}_3$  hold, we have  $a(x_1, x_2).P + b(x_3).Q \models \Gamma, \gamma^I(\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3)$ . This fact corresponds to the  $\gamma$ -introduction rule with behaviour assignment expressed as the leftmost rule below. On the other hand, the set  $\gamma^E$  gives the  $\gamma$ -elimination rules with behaviour assignment expressed as the remaining rules below.

$$\frac{P \models \Gamma, x_1 : \mathbf{P}_1, x_2 : \mathbf{P}_2 \quad Q \models \Gamma, x_3 : \mathbf{P}_3}{a(x_1, x_2).P + b(x_3).Q \models \Gamma, \gamma^I(\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3)} \quad \frac{N_1 \models \Gamma, \mathbf{P}_1^\perp}{x|\bar{c}\langle N_1 \rangle \models \Gamma, x : \gamma^E\langle \mathbf{P}_1^\perp, \mathbf{P}_2^\perp, \mathbf{P}_3^\perp \rangle} \quad \frac{N_2 \models \Gamma, \mathbf{P}_2^\perp \quad N_3 \models \Delta, \mathbf{P}_3^\perp}{x|\bar{d}\langle N_2, N_3 \rangle \models \Gamma, \Delta, x : \gamma^E\langle \mathbf{P}_1^\perp, \mathbf{P}_2^\perp, \mathbf{P}_3^\perp \rangle}$$

These examples show that the sequence  $x_1, x_2, x_3$  in  $\gamma$  fixes the arity of  $\gamma$  and that the occurrences of  $x_1, x_2$  and  $x_3$  in negative actions of  $\alpha^I$  (resp.  $\alpha^E$ ) determines the premises of the  $\gamma$ -introduction rule (resp. the  $\gamma$ -elimination rules). To illustrate the role of the order of variable-sequences in connectives, consider the connective  $\delta = (x_1, x_2, \{a(x_1), b(x_2)\}, \{c(x_2, x_1)\})$ , which provides the inference rules

$$\frac{P \models \Gamma, x_1 : \mathbf{P}_1 \quad Q \models \Gamma, x_2 : \mathbf{P}_2}{a(x_1).P + b(x_2).Q \models \Gamma, \mathbf{P}_1 \star \mathbf{P}_2} \quad \frac{N_2 \models \Gamma, \mathbf{P}_2^\perp \quad N_1 \models \Delta, \mathbf{P}_1^\perp}{x|\bar{c}\langle N_2, N_1 \rangle \models \Gamma, \Delta, x : \mathbf{P}_1^\perp \bar{\star} \mathbf{P}_2^\perp}$$

with  $\mathbf{P} \star \mathbf{Q} := \delta^I(\mathbf{P}, \mathbf{Q})$  and  $\mathbf{N} \bar{\star} \mathbf{M} := \delta^E\langle \mathbf{N}, \mathbf{M} \rangle$ . In the second rule, the order in which  $\mathbf{P}_1^\perp$  and  $\mathbf{P}_2^\perp$  appear in  $\mathbf{P}_1^\perp \bar{\star} \mathbf{P}_2^\perp$  is the reverse of the way they appear in the premises, since  $c(x_2, x_1)$  has the order in which  $x_2$  appears first.

### 2.3 A Reformulation of Harmony

Using our notion of connective, we define the *harmony condition* in Computational Ludics.

**Definition 2.11** (Harmony Condition). *Let  $\alpha$  be an  $n$ -ary connective. The connective  $\alpha$  satisfies the inversion condition if and only if  $\alpha^E \subseteq \alpha^I$  holds, and  $\alpha$  satisfies the recovery condition if and only if  $\alpha^I \subseteq \alpha^E$  holds. We say that the connective  $\alpha$  satisfies the harmony condition if and only if  $\alpha$  satisfies both the inversion condition and the recovery condition.*

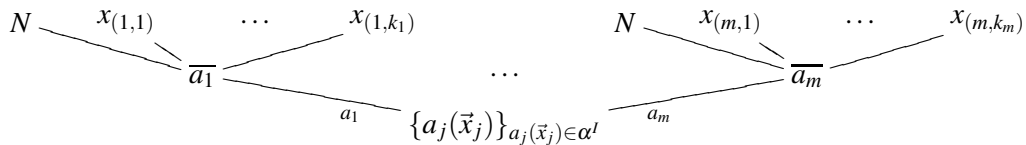
Notice that the connectives  $\gamma$  and  $\delta$  in Example 2.10 do not satisfy this harmony condition and so they are “bad” connectives in this sense.

The inversion condition above is a reformulation of Prawitz’s inversion principle in the following sense. Let  $\alpha$  be an  $n$ -ary connective, then the inversion condition for  $\alpha$  is equivalent to the following condition (see Proposition 2.12.(1) below):

( $\beta$ ) for any  $\alpha^I$ -indexed family  $\{P_{a_j}\}_{a_j(\vec{x}_j) \in \alpha^I}$  of positive total  $c$ -designs, any  $a_i(x_{(i,1)}, \dots, x_{(i,k)})$  in  $\alpha^E$  and any negative  $c$ -designs  $N_1, \dots, N_k$ , the  $c$ -design  $(\sum_{\alpha^I} a_j(\vec{x}_j).P_{a_j})|\bar{a}_i\langle N_1, \dots, N_k \rangle$  reduces to the  $c$ -design  $P_{a_i}[N_1/x_{(i,1)}, \dots, N_k/x_{(i,k)}]$  with  $P_{a_i}[N_1/x_{(i,1)}, \dots, N_k/x_{(i,k)}] \neq \Omega$ .

This condition means that any application of an  $\alpha$ -elimination rule after the  $\alpha$ -introduction rule restores one of the subdesigns that are premises of the latter rule; that is, the  $\beta$ -reduction is always available for  $\alpha$ . Therefore, our inversion condition corresponds to Prawitz’s inversion principle via its equivalence to ( $\beta$ ). Furthermore, our inversion condition makes precise and straightforward the idea of containment:  $\alpha^E$  is contained in  $\alpha^I$  in the set-theoretic sense.

Next, to explain the recovery condition, the  $\eta$ -expanded form of a negative  $c$ -design  $N$  with respect to a connective  $\alpha$  is defined as the negative  $c$ -design  $\sum_{\alpha^I} a_j(\vec{x}_j).(N|\bar{a}_j\langle x_{(j,1)}, \dots, x_{(j,k_j)} \rangle)$  such that  $\{x_{(j,1)}, \dots, x_{(j,k_j)}\}$  and  $\text{fv}(N)$  are disjoint for any  $a_j(\vec{x}_j) \in \alpha^I$ , where bound variables in  $\alpha$  are renamed if necessary. This expanded form can be depicted as the following tree:



If we take a unary name  $\lambda$  and put  $@ := \bar{\lambda}$ , then the  $\eta$ -expansion in  $\lambda$ -calculus can be expressed as the expansion of  $N$  to  $\lambda(x).(N|@ \langle x \rangle)$  and so the expanded form above is a generalisation of  $\eta$ -expansion in  $\lambda$ -calculus. Then, the following condition is equivalent to our recovery condition (see Proposition 2.12.(2) below):

- ( $\eta$ ) there is a function  $f$  mapping each negative action  $a_j(x_{(j,1)}, \dots, x_{(j,k)})$  in  $\alpha^I$  to a  $k$ -ary negative action  $c_{f(j)}(\vec{y}_{f(j)}) \in \alpha^E$  such that for any negative c-design  $N$ , the  $\eta$ -expanded form of  $N$  with respect to  $\alpha$  and the c-design  $\sum_{\alpha^I} a_j(x_{(j,1)}, \dots, x_{(j,k)}).(N|\overline{c_{f(j)}} \langle x_{(j,1)}, \dots, x_{(j,k)} \rangle)$  are equal.

This condition says that the  $\eta$ -expansion can be performed for  $\alpha$ , hence our recovery condition corresponds to the recovery principle via the condition ( $\eta$ ) because the recovery principle means the availability of  $\eta$ -expansion in natural deduction. In particular, note that our recovery condition is literally the converse of the inversion condition:  $\alpha^I$  is contained in  $\alpha^E$ . The following proposition summarises the correspondence between our inversion/recovery conditions and the inversion/recovery principles:

**Proposition 2.12.** *Let  $\alpha$  be an  $n$ -ary connective. (1) The connective  $\alpha$  satisfies the inversion condition if and only if  $\alpha$  satisfies the condition ( $\beta$ ). (2) The connective  $\alpha$  satisfies the recovery condition if and only if  $\alpha$  satisfies the condition ( $\eta$ ).*

*Proof.* (1) The “only if” direction is obvious. Conversely, suppose that  $\alpha^E \not\subseteq \alpha^I$  holds, and take a negative action  $a_i(\vec{x}_i)$  in  $\alpha^E \setminus \alpha^I$ . Moreover, let  $\{P_{a_j}\}_{a_j(\vec{x}_j) \in \alpha^I}$  be an  $\alpha^I$ -indexed family of positive total c-designs and  $\vec{N}_i$  be arbitrary negative c-designs. Then, we have  $(\sum_{\alpha^I} a_j(\vec{x}_j).P_{a_j})|\overline{a_i} \langle \vec{N}_i \rangle \longrightarrow P_{a_i}[\vec{N}_i/\vec{x}_i] = \Omega$  because  $P_{a_i} = \Omega$  holds, contradiction.

(2) The “only if” direction is obvious. Suppose that there is a negative action  $a_j(x_{(j,1)}, \dots, x_{(j,k)})$  in  $\alpha^I \setminus \alpha^E$ . By the definition of connectives, we have  $a_j \neq a_i$  for any  $k$ -ary negative action  $a_i$  in  $\alpha^E$ , hence  $\sum_{\alpha^I} a_j(x_{(j,1)}, \dots, x_{(j,k)}).(N|\overline{c_{f(j)}} \langle x_{(j,1)}, \dots, x_{(j,k)} \rangle)$  cannot be the  $\eta$ -expanded form. Contradiction.  $\square$

Our notion of connective is a generalisation of the notion of *logical connective* defined in [21, 2, 3]. In our framework, we define that a connective  $\alpha$  is *logical* if  $\alpha$  satisfies the harmony condition. Below we abbreviate a logical connective  $(\vec{z}, \{a_1(\vec{x}_1), \dots, a_n(\vec{x}_n)\}, \{a_1(\vec{x}_1), \dots, a_n(\vec{x}_n)\})$  as  $(\vec{z}, \{a_1(\vec{x}_1), \dots, a_n(\vec{x}_n)\})$ . Then, it is obvious that logical connectives in our sense coincide with ones in the sense of [21, 2, 3]. The connectives of the linear fragment **MALLP** of polarised linear logic are defined in [21] as instances of logical connectives. For example, the connective  $\&$ , which is called *With*, can be defined as the logical connective  $(x_1, x_2, \{\pi_1(x_1), \pi_2(x_2)\})$ , and this gives the following inference rules:

$$\frac{P \Vdash \Gamma, x_1 : \mathbf{P}_1 \quad Q \Vdash \Gamma, x_2 : \mathbf{P}_2}{\pi_1(x_1).P + \pi_2(x_2).Q \Vdash \Gamma, \&^I(\mathbf{P}_1, \mathbf{P}_2)} \quad \frac{N \Vdash \Gamma, \mathbf{P}_1^\perp}{x|\overline{\pi_1} \langle N \rangle \Vdash \Gamma, x : \&^E(\mathbf{P}_1^\perp, \mathbf{P}_2^\perp)} \quad \frac{M \Vdash \Gamma, \mathbf{P}_2^\perp}{x|\overline{\pi_2} \langle M \rangle \Vdash \Gamma, x : \&^E(\mathbf{P}_1^\perp, \mathbf{P}_2^\perp)}$$

If we put  $\mathbf{N}_1 \oplus \mathbf{N}_2 := \&^E \langle \mathbf{N}_1, \mathbf{N}_2 \rangle$  as in [21] and write  $\mathbf{P}_1 \& \mathbf{P}_2$  instead of  $\&^I(\mathbf{P}_1, \mathbf{P}_2)$ , then the rules above are exactly the  $\&$ -rule and the  $\oplus$ -rules in one-sided sequent calculus. This is compatible with our explanations of  $\&^I$  as the introduction rule and  $\&^E$  as the elimination rules, because the  $\&$ -introduction rule corresponds to the  $\&$ -right rule in sequent calculus and the  $\&$ -elimination rules correspond to the  $\oplus$ -right rules via the De Morgan equivalence between  $(A \& B)^\perp$  and  $A^\perp \oplus B^\perp$ . Another pair of examples from **MALLP**-connectives is the pair of lifting operators: consider the logical connective  $\uparrow = (x_1, \{\uparrow(x_1)\})$ , then we have the inference rules

$$\frac{P \Vdash \Gamma, x_1 : \mathbf{P}}{\uparrow(x_1).P \Vdash \Gamma, \uparrow \mathbf{P}} \quad \frac{N \Vdash \Gamma, \mathbf{P}^\perp}{x|\overline{\uparrow} \langle N \rangle \Vdash \Gamma, x : \downarrow \mathbf{P}^\perp}$$

where  $\uparrow \mathbf{P} := \uparrow^I(\mathbf{P})$  and  $\downarrow \mathbf{N} := \uparrow^E \langle \mathbf{N} \rangle$ .

As explained in [3], a logical connective of the form

$$(x_{(1,1)}, \dots, x_{(1,k_1)}, \dots, x_{(m,1)}, \dots, x_{(m,k_m)}, \{a_1(x_{(1,1)}, \dots, x_{(1,k_1)}), \dots, a_m(x_{(m,1)}, \dots, x_{(m,k_m)})\})$$

induces the following inference rules (here we suppress the term-information):

$$\frac{\Gamma, \mathbf{P}_{(1,1)}, \dots, \mathbf{P}_{(1,k_1)} \quad \dots \quad \Gamma, \mathbf{P}_{(m,1)}, \dots, \mathbf{P}_{(m,k_m)}}{\Gamma, \&_{1 \leq i \leq m} \wp_{1 \leq j \leq k_i} \mathbf{P}_{(i,j)}}$$

$$\frac{\Gamma_1, \mathbf{P}_{(1,1)}^\perp \quad \dots \quad \Gamma_{k_1}, \mathbf{P}_{(1,k_1)}^\perp}{\Gamma_1, \dots, \Gamma_{k_1}, \oplus_{1 \leq i \leq m} \otimes_{1 \leq j \leq k_i} \mathbf{P}_{(i,j)}^\perp} \quad \frac{\Gamma_1, \mathbf{P}_{(m,1)}^\perp \quad \dots \quad \Gamma_{k_m}, \mathbf{P}_{(m,k_m)}^\perp}{\Gamma_1, \dots, \Gamma_{k_m}, \oplus_{1 \leq i \leq m} \otimes_{1 \leq j \leq k_i} \mathbf{P}_{(i,j)}^\perp}$$

where  $\&_{1 \leq i \leq n}$  is the  $n$ -ary With and  $\wp_{1 \leq i \leq n}$  is the  $n$ -ary multiplicative disjunction (Par) with their duals  $\oplus_{1 \leq i \leq n}$ ,  $\otimes_{1 \leq i \leq n}$ . This shows that logical connectives in Computational Ludics include *synthetic* connectives ([4]) such as the combination of  $\&$  and  $\wp$  (or  $\oplus$  and  $\otimes$ ) but they do not cover the combination of connectives of opposite polarities such as  $\otimes$  and  $\wp$ . Therefore, Acclavio-Maieli's generalised connectives in [1] are not subsumed under logical connectives of Computational Ludics, because the former covers the combination of  $\otimes$  and  $\wp$ . Moreover, we conjecture that Computational Ludics cannot deal with *non-decomposable* logical connectives (i.e. logical connectives which cannot be decomposed into standard **MALL** connectives), which are studied in [1]. On the other hand, logical connectives of Computational Ludics subsume additive connectives, synthetic connectives and the units  $\perp, \top, \mathbf{0}, \mathbf{1}$ , and we conjecture that the framework of [1] cannot deal with these connectives and units.<sup>6</sup>

While logical connectives are the connectives satisfying the harmony condition, there are inharmonious connectives, as we have seen some of them in Example 2.10. The failure of the harmony condition can be sometimes tricky. Consider a connective  $\alpha_0 = (x_1, x_2, \{a(x_1), b(x_2)\}, \{c(x_1), b(x_2)\})$ . Neither the inversion condition nor the recovery condition are satisfied by  $\alpha_0$ , but  $\alpha_0$  has the negative action  $b(x_2)$  which is a common element of both  $\alpha_0^I$  and  $\alpha_0^E$ . This means that some  $\beta$ -reduction steps are definable for  $\alpha_0$ , although not all of the  $\beta$ -reduction steps are. Indeed, for any positive  $c$ -designs  $P$  and  $Q$  with  $P \neq \Omega \neq Q$ , we have  $a(x_1).P + b(x_2).Q | \bar{b} \langle N \rangle \longrightarrow Q | N/x_2 \neq \Omega$  but  $a(x_1).P + b(x_2).Q | \bar{c} \langle N \rangle \longrightarrow \Omega$ . Hence, though the connective  $\alpha_0$  is not a logical one, it is a connective that is not deprived of any meaning. More precisely, it is not completely deprived of any computational meaning, as it allows some  $\beta$ -reduction steps.

### 3 Two Characterisations of Harmony Condition

In this section, we give two conditions each of which is equivalent to the harmony condition defined in the previous section. This will show that the harmony condition is in fact equivalent to a form of completeness which is proper to the Ludics point of view: the absence of any fundamental distinction between derivations and models is what makes it possible to pass from one to another, and vice versa (as to this viewpoint, see Section 2.1). In Section 3.1, we propose the first condition called the *dual decomposability of connectives*. Informally, a connective  $\alpha$  is dually decomposable if the ‘‘introduction’’ behaviour

<sup>6</sup>As to our connectives not satisfying the harmony condition, they have the following difference from Acclavio-Maieli's generalised connectives: the latter connectives always satisfy a form of the inversion principle, because these connectives are designed to satisfy cut reduction, which is indeed nothing but the sequent calculus counterpart of detour reduction in natural deduction.

$\alpha^I(\mathbf{P}_1, \dots, \mathbf{P}_n)$  (resp. the “elimination” behaviour  $\alpha^E\langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle$ ) is decomposed into  $\mathbf{P}_1, \dots, \mathbf{P}_n$  (resp.  $\mathbf{N}_1, \dots, \mathbf{N}_n$ ) via the negative actions in  $\alpha^E$  (resp.  $\alpha^I$ ). Specifically, we will have  $\alpha^E\langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle = \bigcup_{a_i(\vec{x}_i) \in \alpha^I} \overline{a_i}\langle \mathbf{N}_{(i,1)}, \dots, \mathbf{N}_{(i,k)} \rangle \cup \{\mathfrak{X}\}$  in the case of the “elimination” behaviour. Following [9, p. 409], one can see a form of completeness here: if the set  $\mathbf{E} = \bigcup_{a_i(\vec{x}_i) \in \alpha^E} \overline{a_i}\langle \mathbf{N}_{(i,1)}, \dots, \mathbf{N}_{(i,k)} \rangle$  is treated as a set of derivations composed from  $\mathbf{N}_1, \dots, \mathbf{N}_n$  by  $\alpha^E$ ,  $\mathbf{E}^\perp$  serves as a set of models which are orthogonal to any derivation in  $\mathbf{E}$ . Then, the biorthogonal  $\mathbf{E}^{\perp\perp}$  corresponds to the set of derivations validated by these models, and  $\mathbf{E}^{\perp\perp}$  is equal to  $\alpha^E\langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle$  by definition. The dual decomposability of connectives says that the derivations in  $\mathbf{E}^{\perp\perp}$  except  $\mathfrak{X}$  are already included in  $\bigcup_{a_i(\vec{x}_i) \in \alpha^I} \overline{a_i}\langle \mathbf{N}_{(i,1)}, \dots, \mathbf{N}_{(i,k)} \rangle$ , which is a set of derivations composed from  $\mathbf{N}_1, \dots, \mathbf{N}_n$  by  $\alpha^I$ . In this sense,  $\bigcup_{a_i(\vec{x}_i) \in \alpha^I} \overline{a_i}\langle \mathbf{N}_{(i,1)}, \dots, \mathbf{N}_{(i,k)} \rangle$  is “complete”.

In Section 3.2, we propose the second condition which is equivalent to the harmony condition. We call this condition the *dual decomposability of visitable paths*. A visitable path is a sequence of actions induced by the interaction between the elements of an orthogonal pair of a c-design and an anti-design. In other words, a visitable path is an observable trace in the interaction between some c-design and anti-design (i.e. some programs), even if these programs are treated as black boxes (for a detailed discussion on the observability in Ludics, see [6]). As remarked in [7, 13, 14], visitable paths are closely related to the notion of interaction in game semantics. The dual decomposability of visitable paths says that one can find the decomposability and the completeness mentioned above not only in c-designs but also in these observable traces. Consider, for instance, a visitable path induced by some c-design  $T$  in  $\alpha^E\langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle$  and some anti-design against  $T$ . Typically, such a visitable path has an action  $x_0 \overline{a_i}\langle \vec{x}_i \rangle$  with  $a_i(\vec{x}_i) \in \alpha^I$  as its first element, and the remaining sequence is obtained by “shuffling” some visitable paths in  $\mathbf{N}_{(i,1)}, \dots, \mathbf{N}_{(i,k)}$ .

### 3.1 Dual Decomposability of Connectives

To formulate the intuition behind the dual decomposability of connectives precisely, we define *counter sets* by adapting the definition of  $\overline{\alpha^c}\langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle$  and  $\alpha^c(\mathbf{P}_1, \dots, \mathbf{P}_n)$  in [21, p. 2068] to our setting.

**Definition 3.1** (Counter Sets). *Let  $\alpha$  be an  $n$ -ary connective. For any positive  $a$ -behaviours  $\mathbf{P}_1, \dots, \mathbf{P}_n$  and any negative  $a$ -behaviours  $\mathbf{N}_1, \dots, \mathbf{N}_n$ , we define the counter set  $\alpha^I(\mathbf{P}_1, \dots, \mathbf{P}_n)^C$  for  $\alpha^I(\mathbf{P}_1, \dots, \mathbf{P}_n)$  and the counter set  $\alpha^E\langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle^C$  for  $\alpha^E\langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle$  as follows.*

- $\alpha^I(\mathbf{P}_1, \dots, \mathbf{P}_n)^C := \bigcup_{a_i(\vec{x}_i) \in \alpha^I} \overline{a_i}\langle \mathbf{P}_{(i,1)}^\perp, \dots, \mathbf{P}_{(i,k)}^\perp \rangle$ , and
- $\alpha^E\langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle^C$  is defined as the set of all negative  $c$ -designs  $N$  of the following form: for some  $a_i(\vec{x}_i) \in \alpha^E$ , some  $x_{(i,l)} \in \{\vec{x}_i\}$  and some  $Q \in \mathbf{N}_{(i,l)}^\perp$ ,

$$N = a_i(\vec{x}_i) \cdot Q[x_{(i,l)}/x_0] + b_1(\vec{x}_{b_1}) \cdot \mathfrak{X} + \dots + b_m(\vec{x}_{b_m}) \cdot \mathfrak{X},$$

where  $\alpha^E \setminus \{a_i(\vec{x}_i)\} = \{b_1(\vec{x}_{b_1}), \dots, b_m(\vec{x}_{b_m})\}$  holds. Below we use the following abbreviation:  $a_i(\vec{x}_i) \cdot Q[x_{(i,l)}/x_0] + \mathfrak{X}_{\alpha^E} := a_i(\vec{x}_i) \cdot Q[x_{(i,l)}/x_0] + b_1(\vec{x}_{b_1}) \cdot \mathfrak{X} + \dots + b_m(\vec{x}_{b_m}) \cdot \mathfrak{X}$ .

For any negative  $a$ -behaviours  $\mathbf{N}_1, \dots, \mathbf{N}_n$ , we define

$$[\mathbf{N}_1/x_1, \dots, \mathbf{N}_n/x_n] := \{[N_1/x_1, \dots, N_n/x_n] : N_i \in \mathbf{N}_i \text{ for any } i \text{ with } 1 \leq i \leq n\}.$$

**Definition 3.2** (Dual Decomposability of Connectives). *Let  $\alpha$  be an  $n$ -ary connective. Then,  $\alpha$  is dually decomposable if and only if  $\alpha$  satisfies the following conditions:*

1.  $\alpha^E\langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle = \alpha^I(\mathbf{N}_1^\perp, \dots, \mathbf{N}_n^\perp)^C \cup \{\mathfrak{X}\}$ , and

2.  $\sum a(\vec{x}_a).P_a \in \alpha^I(\mathbf{P}_1, \dots, \mathbf{P}_n)$  holds if and only if  $P_{a_i} \in [\mathbf{P}_{(i,1)}^\perp/x_{(i,1)}, \dots, \mathbf{P}_{(i,k)}^\perp/x_{(i,k)}]^\perp$  holds for any  $a_i(x_{(i,1)}, \dots, x_{(i,k)})$  in  $\alpha^E$ .

For any logical connective  $\alpha = (\vec{z}, \alpha_0)$ , the dual decomposability of  $\alpha$  is essentially equivalent to the internal completeness of  $\alpha$  formulated in [21, § 4.4] because  $\alpha^I(\mathbf{P}_1, \dots, \mathbf{P}_n)^C$  and  $\alpha^E(\mathbf{N}_1, \dots, \mathbf{N}_n)^C$  are equal to  $\overline{\alpha^c}(\mathbf{P}_1^\perp, \dots, \mathbf{P}_n^\perp)$  and  $\alpha^c(\mathbf{N}_1^\perp, \dots, \mathbf{N}_n^\perp)$  in [21], respectively. By using this equivalence, one can prove Lemma 3.3 and Proposition 3.4 below in a manner similar to the proof of [21, Lemma 4.13, Theorem 4.14] because any connective satisfying the harmony condition is a logical connective. We will prove Lemma 3.3 for readers' convenience, but omit a proof of Proposition 3.4. Note that in [21], c-designs in behaviours may include cuts and so head normal c-designs are used in [21, Lemma 4.13]. Here, instead, any c-design in behaviours is cut-free; we thus need not use head normal c-designs.

**Lemma 3.3.** *Let  $\alpha = (\vec{z}, \alpha_0)$  be an  $n$ -ary logical connective. We have the following assertions:*

1.  $(\alpha_0 \langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle^C)^\perp \subseteq (\bigcup_{a_i(\vec{x}_i) \in \alpha_0} \overline{a_i} \langle \mathbf{N}_{(i,1)}, \dots, \mathbf{N}_{(i,k)} \rangle) \cup \{\boxtimes\}$ .
2.  $\alpha_0 \langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle \subseteq (\alpha_0 \langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle^C)^\perp$ .
3. If  $\sum a(\vec{x}_a).P_a \in (\alpha_0(\mathbf{P}_1, \dots, \mathbf{P}_n)^C)^\perp$  holds then for any  $a_i(x_{(i,1)}, \dots, x_{(i,k)}) \in \alpha_0$ , we have

$$P_{a_i} \in [\mathbf{P}_{(i,1)}^\perp/x_{(i,1)}, \dots, \mathbf{P}_{(i,k)}^\perp/x_{(i,k)}]^\perp.$$

4.  $\alpha_0(\mathbf{P}_1, \dots, \mathbf{P}_n) = (\alpha_0(\mathbf{P}_1, \dots, \mathbf{P}_n)^C)^\perp$ .

*Proof.* (1.) Assume that  $P \in (\alpha_0 \langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle^C)^\perp$  holds. The case of  $P = \boxtimes$  is trivial, so let  $P$  be  $x_0 \overline{b} \langle M_1, \dots, M_m \rangle$ . By assumption,  $b = a_i$  holds for some  $a_i(\vec{x}_i) \in \alpha_0$ . Fix an arbitrary  $x_{(i,l)}$  and an arbitrary  $Q \in \mathbf{N}_{(i,l)}^\perp$ , then we have  $P \perp a_i(\vec{x}_i).Q[x_{(i,l)}/x_0] + \boxtimes \alpha_0$ . Therefore,  $Q \perp M_l$  holds for any  $Q \in \mathbf{N}_{(i,l)}^\perp$ . It follows that each  $M_l$  belongs to  $\mathbf{N}_{(i,l)}$ , hence  $P \in \bigcup_{a_i(\vec{x}_i) \in \alpha_0} \overline{a_i} \langle \mathbf{N}_{(i,1)}, \dots, \mathbf{N}_{(i,k)} \rangle$  holds.

(2.) Consider  $P \in \alpha_0 \langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle$  and  $N = a_i(\vec{x}_i).Q[x_{(i,l)}/x_0] + \boxtimes \alpha_0 \in \alpha_0 \langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle^C$ . We show  $P \perp N$ , and it suffices to verify that  $N \in (\bigcup_{a_i(\vec{x}_i) \in \alpha_0} \overline{a_i} \langle \mathbf{N}_{(i,1)}, \dots, \mathbf{N}_{(i,k)} \rangle)^\perp$  holds. This holds by the definition of  $\alpha_0 \langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle^C$ .

(3.) Assume that  $\sum a(\vec{x}_a).P_a \in (\alpha_0(\mathbf{P}_1, \dots, \mathbf{P}_n)^C)^\perp$  holds, and consider an arbitrary  $a_i(\vec{x}_i) \in \alpha_0$ . Then, for any  $N_1 \in \mathbf{P}_{(i,1)}^\perp, \dots, N_k \in \mathbf{P}_{(i,k)}^\perp$ , we have  $Q := x_0 \overline{a_i} \langle N_1, \dots, N_k \rangle \in \overline{a_i} \langle \mathbf{P}_{(i,1)}^\perp, \dots, \mathbf{P}_{(i,k)}^\perp \rangle$ , hence  $\sum a(\vec{x}_a).P_a$  and  $Q$  are orthogonal. Therefore, we have

$$P_{a_i} [N_1/x_{(i,1)}, \dots, N_k/x_{(i,k)}] \longrightarrow \boxtimes$$

and so  $P_{a_i} \in [\mathbf{P}_{(i,1)}^\perp/x_{(i,1)}, \dots, \mathbf{P}_{(i,k)}^\perp/x_{(i,k)}]^\perp$  holds.

- (4.) This follows from  $\bigcap_{a_i(\vec{x}_i) \in \alpha_0} (\overline{a_i} \langle \mathbf{P}_{(i,1)}^\perp, \dots, \mathbf{P}_{(i,k)}^\perp \rangle)^\perp = (\bigcup_{a_i(\vec{x}_i) \in \alpha_0} \overline{a_i} \langle \mathbf{P}_{(i,1)}^\perp, \dots, \mathbf{P}_{(i,k)}^\perp \rangle)^\perp$ .  $\square$

By the lemma above, we have the following proposition. The converse of this proposition will be obtained by Proposition 3.24 below, which finishes not only the first characterisation of the harmony condition but also the second.

**Proposition 3.4.** *If a connective  $\alpha$  satisfies the harmony condition, then  $\alpha$  is dually decomposable.*

### 3.2 Dual Decomposability of Visitable Paths

In this subsection, we discuss our second characterisation of the harmony condition. There are three notions that are crucial for this characterisation: *interaction sequences*, *visitable paths* and the *regularity* of behaviours. We follow [13, 14] in defining these notions. We first introduce several notions concerning sequences of actions, in order to define interaction sequences.

**Definition 3.5** (Located Actions). *A located action is one of the following expressions: (1) Daimon  $\boxtimes$ , (2) an expression  $x|\bar{a}\langle x_1, \dots, x_n \rangle$  containing a variable  $x$  and a proper positive action  $\bar{a}$  followed by  $x_1, \dots, x_n$  such that  $\text{ar}(a) = n$  holds and  $x, x_1, \dots, x_n$  are pairwise distinct, (3) an expression  $a^x(x_1, \dots, x_n)$  which consists of a variable  $x$  and a proper negative action  $a(x_1, \dots, x_n)$  with  $x \notin \{x_1, \dots, x_n\}$ .*

In the definition above, we made a slight modification of the notations in [13, 14]: we denote negative located actions by  $a^x(x_1, \dots, x_n)$  instead of  $a_x(x_1, \dots, x_n)$ . The empty sequence is denoted by  $\varepsilon$ . In addition, we use the following variables:  $\kappa$  for located actions,  $\kappa^+$  for positive located actions and  $\kappa^-$  for negative located actions. Hereafter, the word “actions” always means located actions. When  $\kappa$  is of the form  $x|\bar{a}\langle x_1, \dots, x_n \rangle$  or  $a^x(x_1, \dots, x_n)$ , we say  $a$  is the *name* of  $\kappa$ ,  $x$  is the *address* of  $\kappa$  and  $x_1, \dots, x_n$  are the *arguments* of  $\kappa$ . Located actions except  $\boxtimes$  are called *proper located actions*.

The basic entities in this subsection are the following sequences of actions.

**Definition 3.6** (Alternated Justified Sequences). *A finite sequence  $s = \kappa_1 \cdots \kappa_n$  of actions is an alternated justified sequence (in short, aj-sequence) if and only if  $s$  satisfies all of the following conditions:*

- *Alternation: the polarity of  $\kappa_i$  is the opposite of the polarity of  $\kappa_{i+1}$  for any  $i$  with  $1 \leq i \leq n$ .*
- *Linearity: each variable occurring in  $s$  is the address of at most one action in  $s$ .*
- *Daimon: if  $\boxtimes$  appears in  $s$  then  $\boxtimes = \kappa_n$  holds.*
- *Justification: for any proper action  $\kappa_i$  in  $s$ , either (1) there is a unique action  $\kappa_j$  of the opposite polarity such that  $j < i$  holds and the arguments of  $\kappa_j$  includes the address of  $\kappa_i$  or (2) there is no  $\kappa_j$  in  $s$  such that the arguments of  $\kappa_j$  includes the address of  $\kappa_i$ . We say that  $\kappa_i$  is justified by  $\kappa_j$  and denote  $\kappa_j$  by  $\text{just}(\kappa_i)$  if (1) holds, otherwise we say  $\kappa_i$  is initial.*

*We say that  $x$  is free in an aj-sequence  $s$  if and only if  $x$  occurs in  $s$  only as the address of some action in  $s$ , and that  $x$  is bound in  $s$  if and only if  $x$  occurs in  $s$  as an argument of some action in  $s$ .*

Note that the empty sequence  $\varepsilon$  is trivially an aj-sequence and that we adopted Barendregt’s variable convention (cf. § 2). We identify two aj-sequences  $s_1$  and  $s_2$  that are identical modulo renaming of bound variables of  $s_1$  and  $s_2$ . For example,  $(x|\bar{a}\langle y_1, y_2 \rangle)(a_{y_1}(z_1, z_2))$  and  $(x|\bar{a}\langle v_1, v_2 \rangle)(a_{v_1}(w_1, w_2))$  are the same aj-sequence. If  $\kappa = x|\bar{a}\langle x_1, \dots, x_n \rangle$  (resp.  $\kappa = a^x(x_1, \dots, x_n)$ ) holds, we write  $a^x(x_1, \dots, x_n)$  (resp.  $x|\bar{a}\langle x_1, \dots, x_n \rangle$ ) as  $\bar{\kappa}$ . If  $s = \kappa_1 \cdots \kappa_n$  is a non-empty sequence of proper actions, we denote  $\bar{\kappa}_1 \cdots \bar{\kappa}_n$  by  $\bar{s}$ . Moreover, we put  $\bar{\varepsilon} := \varepsilon$ . Let  $s = \kappa_1 \cdots \kappa_n$  be a finite sequence of actions ( $n \geq 0$ ) such that  $\kappa_n$  is the only occurrence of  $\boxtimes$  if  $\boxtimes$  occurs in  $s$ . We define the *dual*  $\tilde{s}$  of  $s$  as follows: if  $\boxtimes$  occurs in  $s$  then  $\tilde{s} := \bar{\kappa}_1 \cdots \bar{\kappa}_{n-1}$ , otherwise  $\tilde{s} := \bar{s}\boxtimes$ . We in particular have  $\tilde{\boxtimes} = \varepsilon$ ,  $\tilde{\varepsilon} = \boxtimes$  and  $\tilde{\tilde{s}} = s$ .

Next, we define *paths*, which subsume some interaction sequences as typical examples. But we first define *views* and *anti-views* of aj-sequences to introduce the notion of path.

**Definition 3.7** (Views and Anti-Views of Alternated Justified Sequences). *Let  $s$  be an aj-sequence. We define the view  $\lceil s \rceil$  of  $s$  by induction. (1) If  $s = \varepsilon$  holds, then  $\lceil s \rceil := \varepsilon$ . (2) If  $s = s'\kappa^+$  holds, then  $\lceil s \rceil := \lceil s' \rceil \kappa^+$ . (3) Let  $s$  be  $s'\kappa^-$ . If  $\kappa^-$  is initial, then  $\lceil s \rceil := \kappa^-$ , otherwise  $\lceil s \rceil := \lceil s_0 \rceil \kappa^-$  where  $s_0$  is the prefix of  $s$  such that  $\kappa^-$  is justified by the last action of  $s_0$ .*

*The anti-view  $\lfloor s \rfloor$  of  $s$  is defined as  $\lfloor s \rfloor := \lceil s_0 \rceil$  with  $s_0 = \tilde{s}$ .*

**Definition 3.8** (Paths). *A path is an aj-sequence  $s$  satisfying 1. and 2. below.*

1. *Proponent-visibility: For any prefix  $s_0\kappa^+$  of  $s$  with  $\kappa^+$  proper, if  $\kappa^+$  is justified in  $s_0$  then  $\text{just}(\kappa^+)$  occurs in  $\ulcorner s_0 \urcorner$ .*
2. *Opponent-visibility: For any prefix  $s_0\kappa^-$  of  $s$ , if  $\kappa^-$  is justified in  $s_0$  then  $\text{just}(\kappa^-)$  occurs in  $\llcorner s_0 \llcorner$ .*

A non-empty path is called *positive* (resp. *negative*) if its first action is positive (resp. negative), and the empty path  $\varepsilon$  is defined as a negative path. When  $D$  is a set of sequences of actions and  $\kappa$  is a proper action, we denote the set  $\{\kappa s : s \in D\}$  of sequences by  $\kappa D$ .

One can also consider views and paths occurring in c-designs or anti-designs, where c-designs and anti-designs are treated as trees or forests formed by views (see Figure 1). These notions of views and paths are used in our proof too.

**Definition 3.9** (Views and Paths of C-Designs and Anti-Designs). *Let  $P$  be a positive c-design and  $N$  be a negative c-design with  $x \notin \text{fv}(N)$ , and assume that both  $P$  and  $N$  are cut- and identity-free. We define the two sets  $\mathbb{V}(P)$  and  $\mathbb{V}(N)_x$  of sequences of actions simultaneously:*

- $\mathbb{V}(\Omega) := \emptyset$ ,  $\mathbb{V}(\boxplus) := \{\boxplus\}$  and  $\mathbb{V}(x|\bar{a}(\vec{N})) := \{\kappa_a^+\} \cup \bigcup_{i \leq \text{ar}(a)} \kappa_a^+ \mathbb{V}(N_i)_{y_i}$ , where  $\vec{y} = y_1, \dots, y_k$  are fresh and  $\kappa_a^+ = x|\bar{a}(\vec{y})$  holds,
- $\mathbb{V}(N) := \mathbb{V}(N)_{x_0}$ ,  $\mathbb{V}(\sum a(\vec{x}_a).P_a)_x := \{\varepsilon\} \cup \bigcup \{\kappa_a^- \mathbb{V}(P_a) \cup \{\kappa_a^-\} : P_a \neq \Omega\}$  with  $\kappa_a^- = a^x(\vec{x}_a)$ .

*A sequence  $s$  of actions is a view of  $P$  (resp. a view of  $[N/x]$ ) if and only if  $s \in \mathbb{V}(P)$  (resp.  $s \in \mathbb{V}(N)_x$ ) holds. A path  $p$  is a path of  $P$  (resp. a path of  $[N/x]$ ) if and only if for any non-empty prefix  $s$  (resp. any prefix  $s$ ) of  $p$ ,  $\ulcorner s \urcorner$  is a view of  $P$  (resp. a view of  $[N/x]$ ).*

*A view of a cut-free anti-design  $[G]$  is a view of some member of  $[G]$ . A path of a cut-free anti-design  $[G]$  against positives (resp. a cut-free anti-design  $[G]$  against negatives) is a negative path (resp. a positive path)  $s$  such that for any prefix  $s_0$  (resp. any non-empty prefix  $s_0$ ) of  $s$ ,  $\ulcorner s_0 \urcorner$  is a view of  $[G]$ .*

Intuitively, a view of a c-design  $T$  is a branch (or one of its prefixes) of Pavaux's tree-representation  $\mathcal{T}(T)$  of the c-design  $T$  (for the details, see [13, § 3.1] and [14, § 1.2]). Then, a path of  $T$  is a sequence in  $\mathcal{T}(T)$  traced by proceeding along possibly several branches from the root. For instance, in this representation of c-designs, the positive c-design  $P$  and the anti-design  $[N/x_0]$  with

$$P = x_0|\bar{a}\langle b(x_1).(x_1|\bar{c}\langle \rangle), b(x_2).(x_2|\bar{c}\langle \rangle) \rangle, \quad N = a(y_1, y_2). \left( y_1|\bar{b}\langle c(\cdot).(y_2|\bar{b}\langle a(y_5, y_6).\boxplus + c(\cdot).\boxplus) \rangle \right)$$

are depicted as the left-hand tree and the right-hand tree in Figure 1, respectively. The two aj-sequences

$$x_0|\bar{a}\langle y_1, y_2 \rangle b^{y_1}(x_1) x_1|\bar{c}\langle \rangle \quad \text{and} \quad x_0|\bar{a}\langle y_1, y_2 \rangle b^{y_2}(x_2) x_2|\bar{c}\langle \rangle$$

are views of  $P$ , namely the left-hand branch and the right-hand branch of  $P$ . The aj-sequence

$$x_0|\bar{a}\langle y_1, y_2 \rangle b^{y_1}(x_1) x_1|\bar{c}\langle \rangle b^{y_2}(x_2) x_2|\bar{c}\langle \rangle$$

is a path of  $P$ . Notice that views and paths of some c-design are indeed views and paths in the sense of Definitions 3.7 and 3.8.

As a further step toward the definition of interaction sequences, we define *multi-designs*, which were introduced in [14]. Multi-designs are generalisations of both c-designs and anti-designs. For the need of multi-designs in defining interaction sequences, see [14, p. 41].

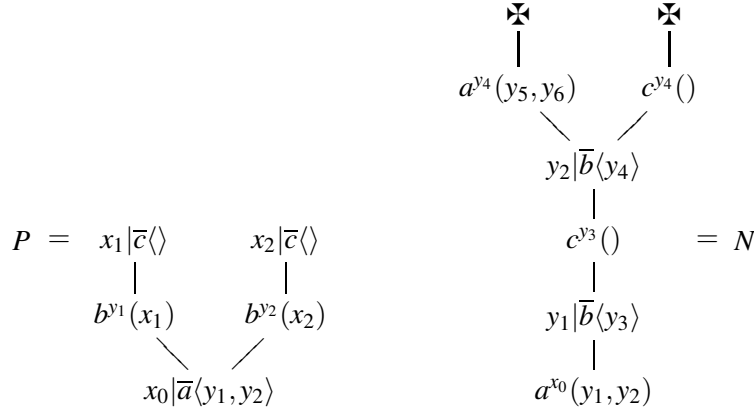


Figure 1: Examples of Pavaux's tree representation of c-designs

**Definition 3.10** (Multi-Designs). (1) A *negative multi-design* is a finite set  $\{(x_1, N_1), \dots, (x_k, N_k)\}$  of pairs of a variable and a negative design such that  $\text{fv}(N_1), \dots, \text{fv}(N_k)$  are pairwise disjoint and  $\text{fv}(N_i) \cap \{x_1, \dots, x_k\}$  is empty for any  $i$  with  $1 \leq i \leq k$ . (2) A *positive multi-design* is a finite set  $\{P, (\vec{x}, \vec{N})\}$  such that  $P$  is a positive design,  $\{(\vec{x}, \vec{N})\} = \{(x_1, N_1), \dots, (x_k, N_k)\}$  is a negative multi-design,  $\text{fv}(P)$  and  $\text{fv}(N_i)$  are disjoint for any  $i$  with  $1 \leq i \leq k$  and  $\text{fv}(P) \cap \{x_1, \dots, x_k\}$  is empty.

For any multi-design  $\mathfrak{D}$ , we define the normal form  $\llbracket \mathfrak{D} \rrbracket$  of  $\mathfrak{D}$  as

$$\llbracket \mathfrak{D} \rrbracket := \{(x, \llbracket N \rrbracket) : (x, N) \in \mathfrak{D}\} \cup \{\llbracket P \rrbracket : P \in \mathfrak{D}\}.$$

A multi-design  $\mathfrak{D}$  is called *standard* if any c-design in  $\mathfrak{D}$  is standard. For any  $(x, N)$  in some multi-design  $\mathfrak{D}$ , we denote  $(x, N)$  by  $[N/x]$ . Moreover, when  $\mathfrak{D} = \{[N_1/x_1], \dots, [N_k/x_k]\}$  is a negative multi-design, we denote the result of substituting  $N_i$  for  $x_i$  in a multi-design  $\mathfrak{E}$  for each  $i$  by  $\mathfrak{E}[\mathfrak{D}]$ . Note that for any positive c-design  $P$ ,  $\{P\}$  is a multi-design and that any anti-design is a multi-design. A *view* of a multi-design  $\mathfrak{D}$  is a view of some c-design in  $\mathfrak{D}$ . A *path* of a multi-design  $\mathfrak{D}$  is a path  $s$  of the same polarity as  $\mathfrak{D}$  such that for any prefix  $s_0$  of  $s$ ,  $\lceil s_0 \rceil$  is a view of  $\mathfrak{D}$ . For any multi-design  $\mathfrak{D}$ , we denote the set  $\bigcup_{T \in \mathfrak{D}} \text{fv}(T)$  of free variables in  $\mathfrak{D}$  by  $\text{fv}(\mathfrak{D})$ , and the set  $\{x : [N/x] \in \mathfrak{D} \text{ for some } N\}$  of *negative places* of  $\mathfrak{D}$  by  $\text{np}(\mathfrak{D})$ .

For example, in Figure 2, the singleton  $\mathfrak{D}$  of the rightmost tree and the set  $\mathfrak{E}$  of the remaining trees are multi-designs. In the  $\lambda$ -term-style notations,  $\mathfrak{D}$  is  $\{y_1 | \bar{b} \langle c \rangle . (y_2 | \bar{b} \langle a(y_5, y_6) . \mathfrak{X} + c \rangle . \mathfrak{X}) \rangle \}$ , and  $\mathfrak{E}$  is  $\{(y_1, b(x_1) . (x_1 | \bar{c} \langle \rangle)), (y_2, b(x_2) . (x_2 | \bar{c} \langle \rangle))\}$ . Though  $\mathfrak{E}$  is also an anti-design,  $\mathfrak{D}$  is not an anti-design, because  $\mathfrak{D}$  contains a non-atomic positive c-design. Notice that  $\mathfrak{D} \cup \mathfrak{E}$  is not a multi-design, because the free variables  $y_1, y_2$  in  $\mathfrak{D}$  belong to  $\text{np}(\mathfrak{E})$ .

The following definition provides some necessary conditions for the interaction between multi-designs.

**Definition 3.11** (Compatibility and Quasi Closed Compatibility). *Two multi-designs  $\mathfrak{D}$  and  $\mathfrak{E}$  are compatible if and only if*

- both  $\text{fv}(\mathfrak{D}) \cap \text{fv}(\mathfrak{E})$  and  $\text{np}(\mathfrak{D}) \cap \text{np}(\mathfrak{E})$  are empty, and
- either they are negative and there is a variable  $x \in \text{np}(\mathfrak{D}) \cup \text{np}(\mathfrak{E})$  such that  $x \notin \text{fv}(\mathfrak{D}) \cup \text{fv}(\mathfrak{E})$  holds, or they are of opposite polarities.

*Two multi-designs  $\mathfrak{D}$  and  $\mathfrak{E}$  are quasi closed compatible if and only if they are of opposite polarities, compatible and satisfy the condition that  $\text{fv}(\mathfrak{D}) \subseteq \text{np}(\mathfrak{E})$  and  $\text{fv}(\mathfrak{E}) \subseteq \text{np}(\mathfrak{D})$  hold.*



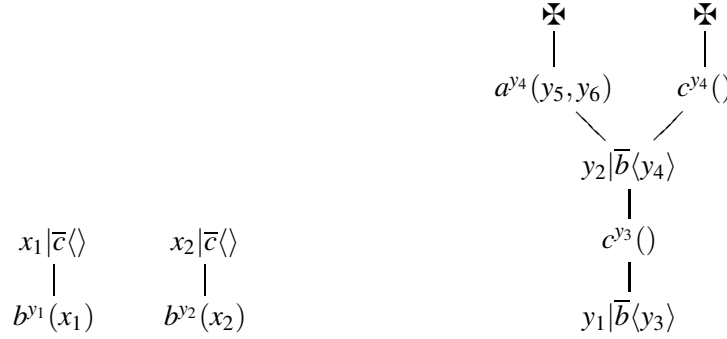


Figure 2: Examples of Pavaux's tree representation of multi-designs

We can apply a *cut* to any two compatible multi-designs. Though the notion of cut is not used in the definition of interaction sequences, this notion will be used to define the orthogonality on multi-designs and formulate Proposition 3.16 below.

**Definition 3.12** (Cut of Multi-Designs). *For any two compatible multi-designs  $\mathcal{D}$  and  $\mathcal{E}$ , the cut  $\text{Cut}_{\mathcal{D}|\mathcal{E}}$  of  $\mathcal{D}$  and  $\mathcal{E}$  is defined by induction on the number of elements in  $\mathcal{E}$ : if  $\mathcal{E}$  is empty then we define  $\text{Cut}_{\mathcal{D}|\mathcal{E}} := \mathcal{D}$ . Let  $\mathcal{E}$  be non-empty.*

1. *If  $P \in \mathcal{E}$  holds, we put  $S := \{[M/y] \in \mathcal{D} : y \in \text{fv}(P)\}$  and define  $\text{Cut}_{\mathcal{D}|\mathcal{E}} := \text{Cut}_{(\mathcal{D} \setminus S) \cup \{P[S]\} | \mathcal{E} \setminus \{P\}}$ .*
2. *If  $[N/x] \in \mathcal{E}$  holds, we put  $S := \{[M/y] \in \mathcal{D} : y \in \text{fv}(N)\}$  and define*
  - (a)  *$\text{Cut}_{\mathcal{D}|\mathcal{E}} := \text{Cut}_{(\mathcal{D} \setminus S) \cup \{[N[S]/x]\} | \mathcal{E} \setminus \{[N/x]\}}$ , if  $x \notin \text{fv}(\mathcal{D})$  holds,*
  - (b)  *$\text{Cut}_{\mathcal{D}|\mathcal{E}} := \text{Cut}_{(\mathcal{D} \setminus S)[N[S]/x] | \mathcal{E} \setminus \{[N/x]\}}$ , if  $x \in \text{fv}(\mathcal{D})$  holds.*

The cut  $\text{Cut}_{\mathcal{D}|\mathcal{E}}$  of multi-designs  $\mathcal{D}, \mathcal{E}$  is well-defined above because  $\text{Cut}_{\mathcal{D}|\mathcal{E}}$  is determined uniquely regardless of the order to apply 1, 2.(a) and 2.(b). We say that two quasi closed compatible multi-designs  $\mathcal{D}$  and  $\mathcal{E}$  are *orthogonal* and write  $\mathcal{D} \perp \mathcal{E}$  if  $\mathfrak{X} \in \llbracket \text{Cut}_{\mathcal{D}|\mathcal{E}} \rrbracket$  holds. Note that this definition of the orthogonality between two multi-designs is broader than the one in [14, Definition 2.1.8], though this broader notion is in fact used in the proof of [14, Proposition 2.2.12]. The reason why we used the broader definition is that we want two multi-designs such as  $\mathcal{D} = \{\mathfrak{X}\}$  and  $\mathcal{E} = \{[N/x]\}$  with  $N$  closed to be orthogonal but they are not in the sense of [14, Definition 2.1.8].

On the basis of the definitions above, we define the notion of interaction sequence.

**Definition 3.13** (Interaction Sequences). *For any two standard multi-designs  $\mathcal{D}$  and  $\mathcal{E}$  such that they are quasi closed compatible, the interaction sequence  $\langle \mathcal{D} \leftarrow \mathcal{E} \rangle$  of  $\mathcal{D}$  with  $\mathcal{E}$  is defined as follows: let  $P$  be the unique positive design in  $\mathcal{D} \cup \mathcal{E}$ .*

- *Let  $P = \mathfrak{X}$  be the case. We define  $\langle \mathcal{D} \leftarrow \mathcal{E} \rangle := \mathfrak{X}$  if  $P \in \mathcal{D}$  holds, otherwise  $\langle \mathcal{D} \leftarrow \mathcal{E} \rangle := \varepsilon$ .*
- *If  $P = \Omega$  holds then we define  $\langle \mathcal{D} \leftarrow \mathcal{E} \rangle := \varepsilon$ .*
- *Assume that  $P = x|\bar{a}\langle \vec{M} \rangle$  holds. If  $P \in \mathcal{D}$  holds then there is a unique negative design  $N$  such that  $[N/x] \in \mathcal{E}$  holds, otherwise there is a unique negative design  $N$  such that  $[N/x] \in \mathcal{D}$ . Let  $N$  be of the form  $\sum b(\vec{y}_b).P_b$ , and we define*

$$\langle \mathcal{D} \leftarrow \mathcal{E} \rangle := \begin{cases} x|\bar{a}\langle \vec{y}_a \rangle \langle (\mathcal{D} \setminus \{P\}) \cup \{[\vec{M}/\vec{y}_a]\} \leftarrow (\mathcal{E} \setminus \{[N/x]\}) \cup \{P_a\} \rangle, & \text{if } P \in \mathcal{D}, \\ \alpha^x(\vec{y}_a) \langle (\mathcal{D} \setminus \{[N/x]\}) \cup \{P_a\} \leftarrow (\mathcal{E} \setminus \{P\}) \cup \{[\vec{M}/\vec{y}_a]\} \rangle, & \text{else.} \end{cases}$$

During the construction of interaction sequences, one usually decomposes a multi-design. For example, consider the c-designs  $P$  and  $N$  in Figure 1 again. Then, the interaction sequence  $\langle \{P\} \leftarrow \{[N/x_0]\} \rangle$  is defined, where the first step for constructing it shortens  $N$  to the rightmost tree in Figure 2 and decomposes  $P$  into the remaining trees in Figure 2. In addition, note that the path

$$x_0 | \bar{a}(y_1, y_2) \quad b^{y_1}(x_1) \quad x_1 | \bar{c} \langle \rangle \quad b^{y_2}(x_2) \quad x_2 | \bar{c} \langle \rangle$$

of  $P$  is equal to  $\langle \{P\} \leftarrow \{[N/x_0]\} \rangle$ . Since  $\langle \mathfrak{D} \leftarrow \mathfrak{E} \rangle = \langle \widetilde{\mathfrak{E}} \leftarrow \mathfrak{D} \rangle$  holds for any orthogonal pair of standard multi-designs  $\mathfrak{D}, \mathfrak{E}$  (see [14, Lemma 2.2.5]) and the pair of  $P, N$  is orthogonal, the path

$$x_0 | \bar{a}(y_1, y_2) \quad y_1 | \bar{b}(x_1) \quad c^{x_1}(\cdot) \quad y_2 | \bar{b}(x_2) \quad c^{x_2}(\cdot) \quad \blacktriangleright$$

of  $N$  is equal to  $\langle \{P\} \leftarrow \widetilde{\{[N/x_0]\}} \rangle = \langle \{[N/x_0]\} \leftarrow \{P\} \rangle$ .

Next, we define visitable paths, which are interaction sequences induced by some orthogonal pair of a c-design and an anti-design. When  $T$  is a positive c-design (resp. a negative c-design), we abbreviate  $\langle \{T\} \leftarrow [G] \rangle$  (resp.  $\langle \{(x_0, T)\} \leftarrow [G] \rangle$ ) as  $\langle T \leftarrow [G] \rangle$ , and  $\langle [G] \leftarrow \{T\} \rangle$  (resp.  $\langle [G] \leftarrow \{(x_0, T)\} \rangle$ ) as  $\langle [G] \leftarrow T \rangle$ .

**Definition 3.14** (Visitable Paths). *Let  $\mathbf{T}$  be a set of standard c-designs of the same polarity, and  $\mathbf{G}$  be a set of standard anti-designs of the same polarity and base.*

- A path  $p$  is visitable in  $\mathbf{T}$  if and only if for some  $T \in \mathbf{T}$  and  $[G] \in \mathbf{T}^\perp$ ,  $\langle T \leftarrow [G] \rangle = p$  holds.
- A path  $p$  is visitable in  $\mathbf{G}$  if and only if for some  $T \in \mathbf{G}^\perp$  and  $[G] \in \mathbf{G}$ ,  $\langle [G] \leftarrow T \rangle = p$  holds.

We denote the set of all visitable paths in  $\mathbf{T}$  (resp.  $\mathbf{G}$ ) by  $V(\mathbf{T})$  (resp.  $V(\mathbf{G})$ ).

We have the following lemma and proposition, which will be used in the proofs of Lemma 3.23.(2) and Proposition 3.24. One can prove the assertions 1, 2 and 3 of the lemma below in the same way to Lemma 2.2.6, Lemma 2.2.10 and Lemma 3.1.5 in [14], respectively.

**Lemma 3.15.** *We have the following assertions:*

1. Let  $\mathfrak{D}, \mathfrak{E}$  be multi-designs with  $\langle \mathfrak{D} \leftarrow \mathfrak{E} \rangle$  defined. If  $\mathfrak{D}$  is positive then any non-empty prefix of  $\langle \mathfrak{D} \leftarrow \mathfrak{E} \rangle$  is a path of  $\mathfrak{D}$ , otherwise any prefix of  $\langle \mathfrak{D} \leftarrow \mathfrak{E} \rangle$  is a path of  $\mathfrak{D}$ . In particular, if  $\langle \mathfrak{D} \leftarrow \mathfrak{E} \rangle$  is finite, then it is a path of  $\mathfrak{D}$ .
2. Assume that a positive multi-design  $\mathfrak{D}$  with  $\Omega \notin \mathfrak{D}$  and a multi-design  $\mathfrak{E}$  are cut-free and satisfy the following two conditions: (i)  $\mathfrak{D}$  and  $\mathfrak{E}$  are quasi closed compatible and have a finite interaction, and (ii) for any path  $s\kappa^+$  of  $\mathfrak{D}$  (resp.  $\mathfrak{E}$ ) such that  $\kappa^+$  is proper and  $\bar{s}$  is a path of  $\mathfrak{E}$  (resp.  $\mathfrak{D}$ ),  $\bar{s}\kappa^+$  is a path of  $\mathfrak{E}$  (resp.  $\mathfrak{D}$ ). Then,  $\mathfrak{D} \perp \mathfrak{E}$  holds.
3. Let  $\mathbf{B}$  be an arbitrary  $a$ -behaviour. If  $p \in V(\mathbf{B})$  holds, then for any positive-ended prefix (resp. negative-ended prefix)  $s$  of  $p$ , we have  $s \in V(\mathbf{B})$  (resp.  $s\blacktriangleright \in V(\mathbf{B})$ ).

When  $s = \kappa_1 \cdots \kappa_n$  is a sequence of actions, a *subsequence* of  $s$  is a sequence  $\kappa_{i_1} \cdots \kappa_{i_k}$  with  $1 \leq i_1 < \cdots < i_k \leq n$ , and we denote by  $s \upharpoonright s'$  the subsequence of  $s$  any of whose actions occurs in a sequence  $s'$ . Moreover, when  $p$  is a path of a multi-design  $\mathfrak{D}$  and  $\mathfrak{E}$  is a multi-design with  $\mathfrak{E} \subseteq \mathfrak{D}$ , we denote the longest subsequence of  $p$  that is a path of  $\mathfrak{E}$  by  $p \upharpoonright \mathfrak{E}$ . For a proof of the following proposition, see [14, Proposition 2.2.12].

**Proposition 3.16** (Associativity for Paths). *For any cut-free multi-designs  $\mathfrak{D}, \mathfrak{E}$  and  $\mathfrak{F}$  such that  $\mathfrak{E} \cup \mathfrak{F}$  is a multi-design with  $\mathfrak{E}$  and  $\mathfrak{F}$  disjoint, if  $\mathfrak{D} \perp (\mathfrak{E} \cup \mathfrak{F})$  holds, then  $\langle \mathfrak{E} \leftarrow \llbracket \text{Cut}_{\mathfrak{F}|\mathfrak{D}} \rrbracket \rangle = \langle \mathfrak{E} \cup \mathfrak{F} \leftarrow \mathfrak{D} \rangle \upharpoonright \mathfrak{E}$  holds.*

To define regular behaviours, we first define the *stable* and *observational orderings*, *intersection* and *incarnation*.

**Definition 3.17** (Stable Ordering and Observational Ordering). *The stable ordering  $\sqsubseteq$  on c-designs is defined as the largest binary relation on c-designs such that if  $T \sqsubseteq U$  holds then one of the following conditions holds:*

1.  $T = \boxtimes = U$ ,
2.  $T = \Omega$  and  $U \in \mathcal{D}^+$ ,
3.  $T = N_0 | \bar{a} \langle N_1, \dots, N_n \rangle$ ,  $U = M_0 | \bar{a} \langle M_1, \dots, M_n \rangle$  and  $N_k \sqsubseteq M_k$  for any  $k$  with  $0 \leq k \leq n$ ,
4.  $T = x = U$ ,
5.  $T = \sum a(\vec{y}^a).P_a$ ,  $U = \sum a(\vec{y}^a).Q_a$  and  $P_a \sqsubseteq Q_a$  for any  $a \in A$ .

*The observational ordering  $\preceq$  on c-designs is defined as the largest binary relation on c-designs such that if  $T \preceq U$  holds then one of 1, 2, 4 above and the following conditions 3' and 5' holds:*

- 3'.  $T = N_0 | \bar{a} \langle N_1, \dots, N_n \rangle$  and either  $U = M_0 | \bar{a} \langle M_1, \dots, M_n \rangle$  and  $N_k \preceq M_k$  for any  $k$  with  $0 \leq k \leq n$  or  $U = \boxtimes$ ,
- 5'.  $T = \sum a(\vec{y}^a).P_a$ ,  $U = \sum a(\vec{y}^a).Q_a$  and  $P_a \preceq Q_a$  for any  $a \in A$ .

**Definition 3.18** (Intersection). *The intersection  $T \cap U$  of c-designs  $T$  and  $U$  is defined by corecursion:*

1.  $\boxtimes \cap \boxtimes = \boxtimes$ ,
2.  $\Omega \cap P = P \cap \Omega = \Omega$ ,
3.  $x | \bar{a} \langle N_1, \dots, N_k \rangle \cap x | \bar{a} \langle M_1, \dots, M_k \rangle = x | \bar{a} \langle N_1 \cap M_1, \dots, N_k \cap M_k \rangle$  if  $N_i \cap M_i$  is defined for each  $i$  with  $1 \leq i \leq k$ ,
4.  $\sum a(\vec{x}_a).P_a \cap \sum a(\vec{x}_a).Q_a = \sum a(\vec{x}_a).P_a \cap Q_a$  if  $P_a \cap Q_a$  is defined for each  $a \in A$ .
5. *In other cases,  $T \cap U$  is not defined.*

The stable ordering  $T \sqsubseteq U$  means that  $U$  is more defined than  $T$ . On the other hand, the observational ordering  $T \preceq U$  means that  $U$  is more likely to converge than  $T$  when they interact with other c-designs. The intersection  $T \cap U$  corresponds to the common part of  $T$  and  $U$  delineated by  $\Omega$  if  $T \cap U$  is defined.

**Definition 3.19** (Incarnation). *Let  $\mathbf{T}$  be a behaviour. The incarnation  $|U|_{\mathbf{T}}$  of  $U$  in  $\mathbf{T}$  is defined as  $|U|_{\mathbf{T}} := \bigcap \{U' \in \mathbf{T} : U' \sqsubseteq U\}$ . We say  $U$  is material in  $\mathbf{T}$  if and only if  $U = |U|_{\mathbf{T}}$  holds, and denote the set of all material designs in  $\mathbf{T}$  by  $|\mathbf{T}|$ .*

Note that  $|U|_{\mathbf{T}}$  is the minimal c-design  $U'$  in  $\mathbf{T}$  such that  $U' \sqsubseteq U$  holds. Next, we define the *shuffle* of two paths, which is a set of paths made by interleaving the actions in  $p$  and  $q$ . Shuffling two paths is a key ingredient for the regularity of behaviour.

**Definition 3.20** (Shuffles). *The shuffle of two paths and the shuffle of two sets of paths are sets of paths defined as follows:*

1. *Let  $p$  and  $q$  be paths. The shuffle  $p \sqcup q$  of  $p$  and  $q$  is defined by distinguishing cases:*
  - *If  $p$  and  $q$  are negative paths, then  $p \sqcup q$  is the set of all paths  $r$  such that any action in  $r$  occurs in  $p$  or  $q$  and both of  $r \upharpoonright p = p$  and  $r \upharpoonright q = q$  hold,*
  - *if  $p$  and  $q$  are positive paths of the same first action  $\kappa^+$ , that is,  $p = \kappa^+ p'$  and  $q = \kappa^+ q'$  hold, then  $p \sqcup q$  is the set of all paths  $r$  such that  $r = \kappa^+ u$  holds for some  $u \in p' \sqcup q'$ , and*
  - *otherwise,  $p \sqcup q$  is not defined.*

2. Let  $D$  and  $D'$  be sets of paths. The shuffle  $D \sqcup D'$  of  $D$  and  $D'$  is defined as the set of all paths  $q$  such that for some  $p \in D$  and  $p' \in D'$  with  $p \sqcup p'$  defined,  $q \in p \sqcup p'$  holds.

For instance, the shuffle of  $x_1|\bar{b}\langle y_1, y_2 \rangle a^{y_1}(y_3) y_3|\bar{c}\langle \rangle$  and  $x_1|\bar{b}\langle y_1, y_2 \rangle a^{y_2}(y_4) y_4|\bar{d}\langle \rangle$  is

$$\left\{ \left( x_1|\bar{b}\langle y_1, y_2 \rangle a^{y_1}(y_3) y_3|\bar{c}\langle \rangle a^{y_2}(y_4) y_4|\bar{d}\langle \rangle \right), \left( x_1|\bar{b}\langle y_1, y_2 \rangle a^{y_2}(y_4) y_4|\bar{d}\langle \rangle a^{y_1}(y_3) y_3|\bar{c}\langle \rangle \right) \right\}.$$

Note that  $\varepsilon \in D \sqcup D'$  holds if and only if  $\varepsilon$  belongs to both of  $D$  and  $D'$ . We omit parentheses in consecutive application of  $\sqcup$  because  $p \sqcup q$  and  $D \sqcup D'$  are associative.

**Definition 3.21** (Regular Behaviours). *An  $a$ -behaviour  $\mathbf{B}$  is regular if and only if the following conditions are satisfied: (1) for any  $T \in |\mathbf{B}|$  and any positive-ended path  $p$  of  $T$ ,  $p \in V(\mathbf{B})$  holds, (2) for any  $T \in |\mathbf{B}^\perp|$  and any positive-ended path  $p$  of  $T$ ,  $p \in V(\mathbf{B}^\perp)$  holds, and (3) both  $V(\mathbf{B})$  and  $V(\mathbf{B}^\perp)$  are closed under  $\sqcup$ .*

Roughly speaking, a behaviour  $\mathbf{B}$  is regular if  $\mathbf{B}$  is atomic and any positive-ended path of its material c-designs belongs to  $V(\mathbf{B})$ , which is closed under  $\sqcup$ , and a similar condition holds for  $\mathbf{B}^\perp$ . When  $\mathbf{N}$  is a negative  $a$ -behaviour, we denote by  $V(x, \mathbf{N})$  the set of all paths obtained by replacing the address  $x_0$  of the first actions of paths in  $V(\mathbf{N})$  with  $x$ . Inspired by the results in [14, § 3.2], we define the second condition equivalent to the harmony condition as follows. This condition says that visitable paths made by a connective  $\alpha$  from regular behaviours are dually decomposable as in our first condition.

**Definition 3.22** (Dual Decomposability of Visitable Paths). *Let  $\alpha$  be an  $n$ -ary connective. Visitable paths of  $\alpha$  are dually decomposable if and only if for any negative regular behaviours  $\mathbf{N}_1, \dots, \mathbf{N}_n$  and any positive  $a$ -behaviours  $\mathbf{P}_1, \dots, \mathbf{P}_n$ , we have*

$$\begin{aligned} \bullet V(\alpha^E \langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle) &= \{ \mathfrak{X} \} \cup \bigcup_{a_i(\bar{x}_i) \in \alpha^E} x_0 |\bar{a}_i \langle \bar{x}_i \rangle (V(x_{(i,1)}, \mathbf{N}_{(i,1)}) \sqcup \dots \sqcup V(x_{(i,k)}, \mathbf{N}_{(i,k)})), \\ \bullet V(\alpha^I \langle \mathbf{P}_1, \dots, \mathbf{P}_n \rangle) &= \{ \varepsilon \} \cup \bigcup_{a_i(\bar{x}_i) \in \alpha^E} a_i^{x_0}(\bar{x}_i) V([\mathbf{P}_{(i,1)}^\perp / x_{(i,1)}, \dots, \mathbf{P}_{(i,k)}^\perp / x_{(i,k)}]^\perp). \end{aligned}$$

For any connective  $\alpha = (\bar{z}, \alpha^I, \alpha^E)$ , we denote the connective  $(\bar{z}, \beta^I, \beta^E)$  with  $\beta^I = \alpha^E$  and  $\beta^E = \alpha^I$  by  $\alpha_\perp$ . Intuitively,  $\alpha_\perp$  is a connective whose introduction and elimination rules are harmonious with  $\alpha^E$  and  $\alpha^I$ , respectively. When  $p$  is a path of a c-design  $T$ , let  $p^c$  be the c-design obtained by replacing with  $\mathfrak{X}$  any positive subdesign  $P$  of  $T$  such that  $P = \Omega$  holds or the first action of elements of  $\mathbb{V}(P)$  does not occur in  $p$ . Note that  $p^c$  is a unique  $\preceq$ -maximal c-design  $U$  such that  $p$  is a path of  $U$ . We define the *bi-view*  $\langle s \rangle$  of an aj-sequence  $s$  as  $\langle \varepsilon \rangle := \varepsilon$ ,  $\langle s \mathfrak{X} \rangle = \langle s \rangle \mathfrak{X}$  and

$$\langle s \kappa \rangle := \begin{cases} \kappa, & \text{if } \kappa \text{ is initial in } s \text{ and } \kappa \neq \mathfrak{X}, \\ \langle s_0 \rangle \kappa, & \text{if } \kappa \text{ is justified by the last action of } s_0 \text{ and } \kappa \neq \mathfrak{X}. \end{cases}$$

The main lemma for our second characterisation of the harmony condition is as follows:

**Lemma 3.23.** *We have the following assertions:*

1. *If a connective  $\alpha$  is dually decomposable, then  $\alpha_\perp$  is dually decomposable.*
2. *For any negative regular behaviours  $\mathbf{N}_1, \dots, \mathbf{N}_n$ , we have*

$$V([\mathbf{N}_1/x_1, \dots, \mathbf{N}_n/x_n]) = V(x_1, \mathbf{N}_1) \sqcup \dots \sqcup V(x_n, \mathbf{N}_n).$$

*Proof.* (1.) Assume that  $\alpha$  is dually decomposable. We have

$$\begin{aligned} (\alpha_\perp)^I \langle \mathbf{P}_1, \dots, \mathbf{P}_n \rangle &= \bigcap_{a_i(\bar{x}_i) \in \alpha^E} (\bar{a}_i \langle \mathbf{P}_1^\perp, \dots, \mathbf{P}_n^\perp \rangle^\perp) = \left( \bigcup_{a_i(\bar{x}_i) \in \alpha^E} \bar{a}_i \langle \mathbf{P}_1^\perp, \dots, \mathbf{P}_n^\perp \rangle \right)^\perp \\ &= \alpha^E \langle \mathbf{P}_1^\perp, \dots, \mathbf{P}_n^\perp \rangle^\perp = (\alpha^I \langle \mathbf{P}_1, \dots, \mathbf{P}_n \rangle^C \cup \{ \mathfrak{X} \})^\perp \\ &= (\alpha^I \langle \mathbf{P}_1, \dots, \mathbf{P}_n \rangle^C)^\perp \cap \{ \mathfrak{X} \}^\perp = (\alpha^I \langle \mathbf{P}_1, \dots, \mathbf{P}_n \rangle^C)^\perp \\ &= \bigcap_{a_i(\bar{x}_i) \in \alpha^I} (\bar{a}_i \langle \mathbf{P}_1^\perp, \dots, \mathbf{P}_n^\perp \rangle^\perp) = \bigcap_{a_i(\bar{x}_i) \in (\alpha_\perp)^E} (\bar{a}_i \langle \mathbf{P}_1^\perp, \dots, \mathbf{P}_n^\perp \rangle^\perp), \end{aligned}$$

so the one half of the dual decomposability of  $\alpha_\perp$  holds. On the other hand, we have

$$\begin{aligned} (\alpha_\perp)^E \langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle &= \left( \bigcup_{a_i(\vec{x}_i) \in \alpha^I} \bar{a}_i \langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle \right)^{\perp\perp} = \left( \bigcap_{a_i(\vec{x}_i) \in \alpha^I} (\bar{a}_i \langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle^\perp) \right)^\perp \\ &= \alpha^I \langle \mathbf{N}_1^\perp, \dots, \mathbf{N}_n^\perp \rangle^\perp. \end{aligned}$$

It is obvious that  $(\bigcup_{a_i(\vec{x}_i) \in \alpha^E} \bar{a}_i \langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle) \cup \{\mathfrak{X}\} \subseteq \alpha^I \langle \mathbf{N}_1^\perp, \dots, \mathbf{N}_n^\perp \rangle^\perp$  holds, because  $\alpha$  is dually decomposable. Conversely, assume that  $P \in \alpha^I \langle \mathbf{N}_1^\perp, \dots, \mathbf{N}_n^\perp \rangle^\perp$  holds. If  $P = \mathfrak{X}$  holds then the assertion is obvious, so let  $P = x_0 \bar{b} \langle M_1, \dots, M_m \rangle$  be the case. By the dual decomposability of  $\alpha$  again, we have  $\sum_{a_i(\vec{x}_i) \in \alpha^E} \cdot \mathfrak{X} \in \alpha^I \langle \mathbf{N}_1^\perp, \dots, \mathbf{N}_n^\perp \rangle$ . There is a negative action  $a_j(\vec{x}_j) \in \alpha^E$  with  $P = x_0 \bar{a}_j \langle M_{(j,1)}, \dots, M_{(j,k)} \rangle$  because  $P \perp \sum_{a_i(\vec{x}_i) \in \alpha^E} \cdot \mathfrak{X}$  holds. Moreover, for any  $Q \in \mathbf{N}_{(j,l)}^\perp$ ,  $N := a_j(\vec{x}_j) \cdot Q[x_{(j,l)}/x_0] + b_1(\vec{x}_{b_1}) \cdot \mathfrak{X} + \dots + b_m(\vec{x}_{b_m}) \cdot \mathfrak{X}$  belongs to  $\alpha^I \langle \mathbf{N}_1^\perp, \dots, \mathbf{N}_n^\perp \rangle$ , where  $\alpha^E \setminus \{a_j(\vec{x}_j)\} = \{b_1(\vec{x}_{b_1}), \dots, b_m(\vec{x}_{b_m})\}$  holds. We have  $P[N/x_0] \longrightarrow Q[M_{(j,l)}/x_{(j,l)}] \longrightarrow \mathfrak{X}$  by  $P \perp N$ , hence  $M_{(j,l)} \in \mathbf{N}_{(j,l)}$  holds for any  $l$  with  $1 \leq l \leq k$ . Therefore,  $P \in \bigcup_{a_i(\vec{x}_i) \in \alpha^E} \bar{a}_i \langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle$  holds. Now we have  $(\bigcup_{a_i(\vec{x}_i) \in \alpha^E} \bar{a}_i \langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle) \cup \{\mathfrak{X}\} = \alpha^I \langle \mathbf{N}_1^\perp, \dots, \mathbf{N}_n^\perp \rangle^\perp$ , so it follows that  $(\alpha_\perp)^E \langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle = (\alpha_\perp)^I \langle \mathbf{N}_1^\perp, \dots, \mathbf{N}_n^\perp \rangle^C \cup \{\mathfrak{X}\}$  holds.

(2.) Put  $\mathbf{G} := [\mathbf{N}_1/x_1, \dots, \mathbf{N}_n/x_n]$ .

( $\subseteq$ ) Let  $p \in V(\mathbf{G})$  be the case. If  $p = \varepsilon$  holds then the assertion is obvious, so let  $p$  be non-empty. By definition,  $p = \langle [G] \leftarrow P \rangle$  holds for some  $[G] = [\mathbf{N}_1/x_1, \dots, \mathbf{N}_n/x_n] \in \mathbf{G}$  and some c-design  $P \in \mathbf{G}^\perp$ . Fix an arbitrary  $i$  with  $1 \leq i \leq n$ , and put  $[G'] := [G] \setminus [N_i/x_i]$ . By the definition of  $\mathfrak{C}ut$ , we have  $P[G'] \in \mathfrak{C}ut_{[G']|P}$  whether  $\text{fv}(P) = \{x_1, \dots, x_n\}$  holds or  $\text{fv}(P) \subset \{x_1, \dots, x_n\}$  holds. Then, by Proposition 3.16, we have

$$p \uparrow [N_i/x_i] = \langle [G] \leftarrow P \rangle \uparrow [N_i/x_i] = \langle [N_i/x_i] \leftarrow [\mathfrak{C}ut_{[G']|P}] \rangle = \langle [N_i/x_i] \leftarrow [P[G']] \rangle.$$

On the other hand, we have  $[[[P[G']]][M/x_i]] = [[[[P[G']]][[M]/x_i]] = [[P[G']][M/x_i]] = \mathfrak{X}$  for any  $M \in \mathbf{N}_i$  by Theorem 2.3, hence  $[P[G']] \in \mathbf{N}_i^\perp$  holds. Moreover, the address of the first action of  $p \uparrow [N_i/x_i]$  is  $x_i$  because  $\langle [N_i/x_i] \leftarrow [P[G']] \rangle$  is a path of  $[N_i/x_i]$  by Lemma 3.15.(1). Therefore,  $p \uparrow [N_i/x_i] = \langle [N_i/x_i] \leftarrow [P[G']] \rangle \in V(x_i, \mathbf{N}_i)$  holds for any  $i$ , so  $p$  belongs to  $V(x_1, \mathbf{N}_1) \sqcup \dots \sqcup V(x_n, \mathbf{N}_n)$ .

( $\supseteq$ ) Let  $p \in V(x_1, \mathbf{N}_1) \sqcup \dots \sqcup V(x_n, \mathbf{N}_n)$  be the case, and assume that we have shown  $\tilde{p}^c \in \mathbf{G}^\perp$ . By  $p \in V(x_1, \mathbf{N}_1) \sqcup \dots \sqcup V(x_n, \mathbf{N}_n)$ ,  $p$  is a path of  $[\vec{N}]$  for some  $[\vec{N}] \in \mathbf{G}$ . Then,  $p \in V(\mathbf{G})$  holds because one can show  $p = \langle [\vec{N}] \leftarrow \tilde{p}^c \rangle$ . Therefore, it suffices to verify that  $\tilde{p}^c \in \mathbf{G}^\perp$  holds.

We suppose that  $\tilde{p}^c \notin \mathbf{G}^\perp$  holds and deduce a contradiction. By  $\tilde{p}^c \notin \mathbf{G}^\perp$ ,  $[\vec{N}] \perp \tilde{p}^c$  does not hold for some  $[\vec{N}] = [N_1/x_1, \dots, N_n/x_n] \in \mathbf{G}$ . The interaction of  $[\vec{N}]$  and  $\tilde{p}^c$  cannot be infinite because  $\tilde{p}^c$  is a completion by means of  $\mathfrak{X}$  and  $p$  is a shuffle of some visitable paths. By Lemma 3.15.(2), there are a path  $t$  of  $[\vec{N}]$  and a negative action  $\kappa^-$  such that  $\overline{t\kappa^-}$  is a path of  $\tilde{p}^c$  and  $t\kappa^-$  is not a path of  $[\vec{N}]$ . Therefore, there is a path  $t$  satisfying the following property (\*): for some anti-design  $[\vec{N}] \in \mathbf{G}$ , (i)  $[\vec{N}] \perp \tilde{p}^c$  does not hold, (ii)  $t$  is a path of  $[\vec{N}]$  and (iii) for some negative action  $\kappa^-$ ,  $\overline{t\kappa^-}$  is a path of  $\tilde{p}^c$  and  $t\kappa^-$  is not a path of  $[\vec{N}]$ . Choose a minimal path  $t$  with respect to length such that  $t$  satisfies the property (\*). We show the following claims (a)–(d), and the claim (d) contradicts the property (\*) of  $t$ .

- (a)  $\tilde{t}^c \in \mathbf{G}^\perp$  holds,
- (b)  $t \in V(x_1, \mathbf{N}_1) \sqcup \dots \sqcup V(x_n, \mathbf{N}_n)$  holds,
- (c) for any  $v \in V(x_1, \mathbf{N}_1) \sqcup \dots \sqcup V(x_n, \mathbf{N}_n)$  and any  $\kappa_1^-$  such that  $\overline{v\kappa_1^-}$  is a path of  $\tilde{p}^c$ , we have  $v\kappa_1^- \cdot \mathfrak{X} \in V(x_1, \mathbf{N}_1) \sqcup \dots \sqcup V(x_n, \mathbf{N}_n)$ ,
- (d)  $t\kappa^-$  is a path of  $[\vec{N}]$ .

(a) Suppose that  $\tilde{t}^c \notin \mathbf{G}^\perp$  holds, hence  $[\vec{N}_1] \perp \tilde{t}^c$  does not hold for some  $[\vec{N}_1] \in \mathbf{G}$ . By Lemma 3.15.(2), there are a path  $t_0$  and a negative action  $\overline{\kappa_2^-}$  such that  $t_0$  is a path of  $[\vec{N}_1]$ ,  $t_0 \overline{\kappa_2^-}$  is a path of  $\tilde{t}^c$  and  $t_0 \overline{\kappa_2^-}$  is not a path of  $[\vec{N}_1]$ . One can see that  $t_0 \overline{\kappa_2^-}$  is a path of  $\tilde{p}^c$ , because views of  $\tilde{t}^c$  are views of  $\tilde{p}^c$ . Then,  $[\vec{N}_1] \perp \tilde{p}^c$  does not hold, otherwise  $t_0 \overline{\kappa_2^-}$  would be a prefix of  $\langle [\vec{N}_1] \leftarrow \tilde{p}^c \rangle$  and so  $t_0 \overline{\kappa_2^-}$  is a path of  $[\vec{N}_1]$  by Lemma 3.15.(1). Moreover,  $t_0$  is strictly shorter than  $t$  because  $t_0 \overline{\kappa_2^-}$  is a path of  $\tilde{t}^c$ . This contradicts the minimality of  $t$  with respect to length.

(b) We have  $t = \langle [\vec{N}] \leftarrow \tilde{t}^c \rangle$  because  $t$  is a path of  $[\vec{N}]$  and  $\tilde{t}$  is a path of  $\tilde{t}^c$ . Then, as in the proof of  $(\subseteq)$  above, we have  $t \upharpoonright [N_i/x_i] \in V(x_i, \mathbf{N}_i)$  for any  $i$  by the claim (a).

(c) Let  $v$  be an element of  $V(x_1, \mathbf{N}_1) \sqcup \cdots \sqcup V(x_n, \mathbf{N}_n)$ , and  $\kappa_1^-$  be a negative action such that  $v \overline{\kappa_1^-}$  is a path of  $\tilde{p}^c$ . By  $p, v \in V(x_1, \mathbf{N}_1) \sqcup \cdots \sqcup V(x_n, \mathbf{N}_n)$ , there are  $p_1, \dots, p_n$  and  $v_1, \dots, v_n$  such that  $p \in p_1 \sqcup \cdots \sqcup p_n$  and  $v \in v_1 \sqcup \cdots \sqcup v_n$  hold and we have  $p_i, v_i \in V(x_i, \mathbf{N}_i)$  for any  $i$  with  $1 \leq i \leq n$ . Assume that  $\kappa_1^-$  is an action in  $p_j$ . It suffices to show  $v_j \kappa_1^- \boxtimes \in V(x_j, \mathbf{N}_j)$ . By the definition of bi-views, we have  $\langle v_j \kappa_1^- \rangle = \langle v \overline{\kappa_1^-} \rangle$ . Moreover, for the prefix  $p' \kappa_1^-$  of  $p$ ,  $\langle v \overline{\kappa_1^-} \rangle = \langle p' \kappa_1^- \rangle$  holds because  $v \overline{\kappa_1^-}$  is a path of  $\tilde{p}^c$ . We have  $\langle p' \kappa_1^- \rangle = \langle p'_j \kappa_1^- \rangle$  for the prefix  $p'_j \kappa_1^-$  of  $p_j$  by the definition of bi-views again, hence  $\langle v_j \kappa_1^- \rangle = \langle p'_j \kappa_1^- \rangle$  holds. We have  $p'_j \kappa_1^- \boxtimes \in V(x_j, \mathbf{N}_j)$  by  $p_j \in V(x_j, \mathbf{N}_j)$  and Lemma 3.15.(3). Then,  $\langle v_j \kappa_1^- \boxtimes \rangle$  is a path of some  $N_j \in |\mathbf{N}_j|$  because  $\langle v_j \kappa_1^- \rangle = \langle p'_j \kappa_1^- \rangle$  holds, hence we have  $\langle v_j \kappa_1^- \boxtimes \rangle \in V(x_j, \mathbf{N}_j)$  by the regularity of  $\mathbf{N}_j$ . Then,  $v_j \kappa_1^- \boxtimes \in v_j \sqcup \langle v_j \kappa_1^- \boxtimes \rangle$  holds, so we have  $v_j \kappa_1^- \boxtimes \in V(x_j, \mathbf{N}_j)$  by the closedness of  $V(x_j, \mathbf{N}_j)$  under  $\sqcup$ .

(d) By the claim (b), we have  $t \in V(x_1, \mathbf{N}_1) \sqcup \cdots \sqcup V(x_n, \mathbf{N}_n)$ . Moreover,  $t \overline{\kappa}^-$  is a path of  $\tilde{p}^c$  by the property  $(*)$  of  $t$ , hence  $t \overline{\kappa}^- \boxtimes \in V(x_1, \mathbf{N}_1) \sqcup \cdots \sqcup V(x_n, \mathbf{N}_n)$  holds by the claim (c). Then, we have  $t_i \overline{\kappa}^- \boxtimes \in V(x_i, \mathbf{N}_i)$  for some  $\mathbf{N}_i$  and some  $t_i$ . The sequence  $t_i$  is a path of  $N_i \in \mathbf{N}_i$ , so  $t_i \overline{\kappa}^-$  is a prefix of a path of  $N_i$ . Therefore,  $\lceil t \overline{\kappa}^- \rceil$  is a view of  $N_i$  because we have  $\lceil t \overline{\kappa}^- \rceil = \lceil t_i \overline{\kappa}^- \rceil$ . The sequence  $t$  is a path of  $[\vec{N}]$ , hence  $t \overline{\kappa}^-$  is a path of  $[\vec{N}]$ .  $\square$

By the lemma above, we have the following proposition:

**Proposition 3.24.** (1) *If a connective  $\alpha$  is dually decomposable, then visitable paths of  $\alpha$  are dually decomposable.* (2) *if visitable paths of a connective  $\alpha$  are dually decomposable, then  $\alpha$  satisfies the harmony condition.*

*Proof.* (1.) First, we show

$$V(\alpha^E \langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle) = \{ \boxtimes \} \cup \bigcup_{a_i(\vec{x}_i) \in \alpha^l} x_0 | \overline{a_i} \langle \vec{x}_i \rangle (V(x_{(i,1)}, \mathbf{N}_{(i,1)}) \sqcup \cdots \sqcup V(x_{(i,k)}, \mathbf{N}_{(i,k)})).$$

$(\subseteq)$  Assume that  $p \in V(\alpha^E \langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle)$  holds. If  $p = \boxtimes$  holds then the assertion is trivial and so let  $p \neq \boxtimes$  be the case. By the dual decomposability of  $\alpha$ ,  $p = \langle P \leftarrow N \rangle$  holds for some  $a_i(\vec{x}_i) \in \alpha^l$ , some  $P = x_0 | \overline{a_i} \langle N_{(i,1)}, \dots, N_{(i,k)} \rangle$  with  $N_{(i,j)} \in \mathbf{N}_{(i,j)}$  for any  $j$ , and some  $N = \sum b(\vec{y}_b) \cdot P_b \in \alpha^E \langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle^\perp$ . Therefore, by the definition of interaction sequences and the renaming of bound variables if necessary,  $p$  is equal to  $x_0 | \overline{a_i} \langle \vec{x}_i \rangle p'$  with  $p' \in V([\mathbf{N}_{(i,1)}/x_{(i,1)}, \dots, \mathbf{N}_{(i,k)}/x_{(i,k)}])$ . By Lemma 3.23.(2), we have

$$p \in x_0 | \overline{a_i} \langle \vec{x}_i \rangle (V(x_{(i,1)}, \mathbf{N}_{(i,1)}) \sqcup \cdots \sqcup V(x_{(i,k)}, \mathbf{N}_{(i,k)})).$$

$(\supseteq)$  The case of  $\boxtimes$  is obvious. Assume that  $p \in x_0 | \overline{a_i} \langle \vec{x}_i \rangle (V(x_{(i,1)}, \mathbf{N}_{(i,1)}) \sqcup \cdots \sqcup V(x_{(i,k)}, \mathbf{N}_{(i,k)}))$  holds for some  $a_i(\vec{x}_i) \in \alpha^l$ . Then, we have  $p = x_0 | \overline{a_i} \langle \vec{x}_i \rangle \langle [G] \leftarrow P \rangle$  for some

$$[G] = [N_1/x_{(i,1)}, \dots, N_k/x_{(i,k)}] \in \mathbf{G} := [\mathbf{N}_{(i,1)}/x_{(i,1)}, \dots, \mathbf{N}_{(i,k)}/x_{(i,k)}]$$

and some  $P \in \mathbf{G}^\perp$  by Lemma 3.23.(2). Define  $Q := x_0|\overline{a_i}\langle N_1, \dots, N_k \rangle$ , then  $Q \in \alpha^I(\mathbf{N}_1^\perp, \dots, \mathbf{N}_n^\perp)^c$  holds and so we have  $Q \in \alpha^E\langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle$  by the dual decomposability of  $\alpha$ . On the other hand, define  $M := a_i(\vec{x}_i).P + \sum_\beta \cdot \mathbf{X}$  with  $\beta = \alpha^I \setminus \{a_i(\vec{x}_i)\}$ , then we have  $M \in \alpha^E\langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle^\perp$  by the dual decomposability again. By  $p = \langle Q \leftarrow M \rangle$ , the assertion holds.

Next, we show

$$V(\alpha^I(\mathbf{P}_1, \dots, \mathbf{P}_n)) = \{\varepsilon\} \cup \bigcup_{a_i(\vec{x}_i) \in \alpha^E} a_i^{x_0}(\vec{x}_i)V([\mathbf{P}_{(i,1)}^\perp/x_{(i,1)}, \dots, \mathbf{P}_{(i,k)}^\perp/x_{(i,k)}]^\perp).$$

( $\subseteq$ ) Assume that  $p \in V(\alpha^I(\mathbf{P}_1, \dots, \mathbf{P}_n))$  holds. If  $p = \varepsilon$  holds then the assertion obviously holds, so let  $p$  be non-empty. By definition, we have  $p = \langle N \leftarrow P \rangle$  with  $N \in \alpha^I(\mathbf{P}_1, \dots, \mathbf{P}_n)$  and  $P \in \alpha^I(\mathbf{P}_1, \dots, \mathbf{P}_n)^\perp$ . By the dual decomposability of  $\alpha$ ,  $N = \sum a(\vec{y}_a).P_a$  holds and  $P_{a_i} \in [\mathbf{P}_{(i,1)}^\perp/x_{(i,1)}, \dots, \mathbf{P}_{(i,k)}^\perp/x_{(i,k)}]^\perp$  holds for any  $a_i(\vec{x}_i) \in \alpha^E$ . We have the following equation (\*)

$$\alpha^I(\mathbf{P}_1, \dots, \mathbf{P}_n)^\perp = \left( \bigcup_{a_i(\vec{x}_i) \in \alpha^I} \overline{a_i}(\mathbf{P}_{(i,1)}^\perp, \dots, \mathbf{P}_{(i,k)}^\perp) \right)^{\perp\perp} = (\alpha_\perp)^E\langle \mathbf{P}_1^\perp, \dots, \mathbf{P}_n^\perp \rangle = (\alpha_\perp)^I(\mathbf{P}_1, \dots, \mathbf{P}_n)^c \cup \{\mathbf{X}\}$$

by the dual decomposability of  $\alpha$  and Lemma 3.23.(1), hence  $P = x_0|\overline{a_i}\langle N_1, \dots, N_k \rangle$  holds for some  $a_i(\vec{x}_i) \in \alpha^E$  and  $N_j \in \mathbf{P}_{(i,j)}^\perp$  holds for any  $j$ . Therefore,  $p \in a_i^{x_0}(\vec{x}_i)V([\mathbf{P}_{(i,1)}^\perp/x_{(i,1)}, \dots, \mathbf{P}_{(i,k)}^\perp/x_{(i,k)}]^\perp)$  holds by the definition of interaction sequences.

( $\supseteq$ ) It suffices to consider the case of non-empty sequences. Assume that

$$p \in \bigcup_{a_i(\vec{x}_i) \in \alpha^E} a_i^{x_0}(\vec{x}_i)V([\mathbf{P}_{(i,1)}^\perp/x_{(i,1)}, \dots, \mathbf{P}_{(i,k)}^\perp/x_{(i,k)}]^\perp)$$

holds and put  $\mathbf{G} := [[\mathbf{P}_{(i,1)}^\perp/x_{(i,1)}, \dots, \mathbf{P}_{(i,k)}^\perp/x_{(i,k)}]$ . By definition, there are  $P \in \mathbf{G}^\perp$  and

$$[H] = [N_1/x_{(i,1)}, \dots, N_k/x_{(i,k)}] \in \mathbf{G}$$

such that  $p = a_i^{x_0}(\vec{x}_i)\langle P \leftarrow [H] \rangle$  and  $a_i(\vec{x}_i) \in \alpha^E$  hold. Define  $N := a_i(\vec{x}_i).P + \sum_\beta \cdot \mathbf{X}$  with  $\beta = \alpha^E \setminus \{a_i(\vec{x}_i)\}$  and  $Q := x_0|\overline{a_i}\langle N_1, \dots, N_k \rangle$ . By the dual decomposability of  $\alpha$ , we have  $N \in \alpha^I(\mathbf{P}_1, \dots, \mathbf{P}_n)$ . Moreover, by  $Q \in (\alpha_\perp)^I(\mathbf{P}_1, \dots, \mathbf{P}_n)^c$ , we have  $Q \in \alpha^I(\mathbf{P}_1, \dots, \mathbf{P}_n)^\perp$  by the equation (\*) in the previous case. Therefore, we have  $p \in V(\alpha^I(\mathbf{P}_1, \dots, \mathbf{P}_n))$  because  $p = \langle N \leftarrow Q \rangle$  holds.

(2.) Assume that  $\alpha$  does not satisfy the harmony condition. We suppose that visitable paths of  $\alpha$  is dually decomposable, and deduce a contradiction. If there is a negative action  $a_i(x_i) \in \alpha^I \setminus \alpha^E$ , consider a path

$$p = x_0|\overline{a_i}\langle \vec{x}_i \rangle p' \in \bigcup_{a_j(\vec{x}_j) \in \alpha^I} x_0|\overline{a_j}\langle \vec{x}_j \rangle (V(x_{(j,1)}, \mathbf{N}_{(j,1)}) \sqcup \dots \sqcup V(x_{(j,k)}, \mathbf{N}_{(j,k)})).$$

We have  $p \in V(\alpha^E\langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle)$ , so  $p = \langle Q \leftarrow M \rangle$  holds for some  $Q \in \alpha^E\langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle$ . The c-design  $Q$  is of the form  $x_0|\overline{a_i}\langle N_{(i,1)}, \dots, N_{(i,m)} \rangle$  with  $a_i(x_i) \in \alpha^I \setminus \alpha^E$  because  $p$  is a path of  $Q$ , hence one can find

$$N \in \alpha^E\langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle^\perp = \left( \bigcup_{a_j(\vec{x}_j) \in \alpha^E} \overline{a_j}(\mathbf{N}_{(j,1)}, \dots, \mathbf{N}_{(j,k)}) \right)^\perp$$

such that  $Q \perp N$  does not hold. Contradiction.

If there is a negative action  $a_i(x_i) \in \alpha^E \setminus \alpha^I$ , consider a path

$$p = a_i^{x_0}(\vec{x}_i)p' \in \bigcup_{a_j(\vec{x}_j) \in \alpha^E} a_j^{x_0}(\vec{x}_j)V([\mathbf{P}_{(j,1)}^\perp/x_{(j,1)}, \dots, \mathbf{P}_{(j,k)}^\perp/x_{(j,k)}]^\perp).$$

We have  $p \in V(\alpha^I(\mathbf{P}_1, \dots, \mathbf{P}_n))$ , so  $p = \langle M \leftarrow Q \rangle$  holds for some  $Q \in \alpha^I(\mathbf{P}_1, \dots, \mathbf{P}_n)^\perp$ . The dual  $\tilde{p} = \langle Q \leftarrow M \rangle$  is a path of  $Q$ , hence  $Q$  is of the form  $x_0 | \bar{a}_i \langle N_{(i,1)}, \dots, N_{(i,m)} \rangle$  with  $a_i(x_i) \in \alpha^E \setminus \alpha^I$ . Therefore, one can find

$$N \in \alpha^I(\mathbf{P}_1, \dots, \mathbf{P}_n)^{\perp\perp} = \bigcap_{a_j(\bar{x}_j) \in \alpha^I} \bar{a}_j \langle \mathbf{P}_{(j,1)}^\perp, \dots, \mathbf{P}_{(j,k)}^\perp \rangle^\perp$$

such that  $Q \perp N$  does not hold. Contradiction.  $\square$

Our two characterisations of the harmony condition are obtained by Propositions 3.4 and 3.24.

**Corollary 3.25** (Characterisation of Harmony). *Let  $\alpha$  be a connective. The following three assertions are equivalent: (1)  $\alpha$  satisfies the harmony condition, (2)  $\alpha$  is dually decomposable and (3) visitable paths of  $\alpha$  is dually decomposable.*

## 4 Concluding Remarks and Future Work

By means of Computational Ludics, we have first reformulated the inversion principle and the recovery principle into the harmony condition. Then, we have shown that the harmony condition is equivalent to both the dual decomposability of connectives and the dual decomposability of visitable paths.

However, a thorough analysis of the fundamental features of proof-theoretic semantics by means of the Computational Ludics tools is far from being definitely achieved. First, the proof-theoretic semantics literature has considered other principles such as *deducibility of identicals* or the *uniqueness* (see [11]) to capture the necessary condition that a set of rules has to satisfy to define a meaningful and logical connective. Examining how these principles can be reformulated in Computational Ludics would be a crucial step for future works. Second, as shown in [3], in Computational Ludics it is possible to have a logical connective (i.e. a connective satisfying the harmony condition) for the non-linear case, which does not enjoy the internal completeness. To fully appreciate the relationship between the logicity and the internal completeness, we will explore the non-linear case.

Concerning the philosophical scope of our work, let us remark that, as we mentioned in the introduction, proof-theoretic semantics has been traditionally developed within the framework of natural deduction. However, as noted in [19, § 1.2], natural deduction is somehow “biased towards intuitionistic logic”. The possibility of associating each connective to a set of introduction rules and then justifying a corresponding set of elimination rules by means of detour reduction works straightforwardly when the intuitionistic rules are considered (on the contrary, the classical rule of *reductio ad absurdum*, or of indirect proof, cannot be easily classified as an introduction rule nor as an elimination rules, and this makes it difficult to define a suitable notion of detour for it; see [10]). A monistic point of view is thus often associated with proof-theoretic semantics, according to which intuitionistic logic is the only right and meaningful logic. The analysis of harmony that we have offered here aims to show that when the notion of proof is formalised within a framework different from natural deduction, then other connectives—different from the intuitionistic ones—can be justified. We took here Computational Ludics as an alternative framework to natural deduction, and we showed that this choice allows for the justification of linear connectives. In this sense, our work can eventually be seen as a contribution to the idea that proof-theoretic semantics is compatible with a *pluralistic* rather than a monistic view of logic. We also claim that our use of Computational Ludics as an alternative framework to natural deduction is legitimised by the fact that it allows us to obtain a more perspicuous formulation of harmony than the one that is usually proposed in the case of (intuitionistic) natural deduction.



## References

- [1] Matteo Acclavio & Roberto Maieli (2020): *Generalized Connectives for Multiplicative Linear Logic*. In: *28th EACSL Annual Conference on Computer Science Logic, CSL 2020, January 13-16, 2020, Barcelona, Spain*, pp. 6:1–6:16, doi:10.4230/LIPIcs.CSL.2020.6.
- [2] Michele Basaldella, Alexis Saurin & Kazushige Terui (2010): *From Focalization of Logic to the Logic of Focalization*. *Electr. Notes Theor. Comput. Sci.* 265, pp. 161–176, doi:10.1016/j.entcs.2010.08.010.
- [3] Michele Basaldella & Kazushige Terui (2010): *On the meaning of logical completeness*. *Logical Methods in Computer Science* 6(4), doi:10.2168/LMCS-6(4:11)2010.
- [4] Pierre-Louis Curien (2005): *Introduction to linear logic and ludics, part II*. CoRR abs/cs/0501039. arXiv:cs/0501039.
- [5] Michael Dummett (1991): *The Logical Basis of Metaphysics*. Duckworth, London.
- [6] Claudia Faggian (2006): *Interactive observability in Ludics: The geometry of tests*. *Theor. Comput. Sci.* 350(2-3), pp. 213–233, doi:10.1016/j.tcs.2005.10.042.
- [7] Christophe Fouqueré & Myriam Quatrini (2018): *Study of Behaviours via Visitable Paths*. *Logical Methods in Computer Science* 14(2), doi:10.23638/LMCS-14(2:7)2018.
- [8] Nissim Francez & Roy Dyckhoff (2012): *A Note on Harmony*. *J. Philosophical Logic* 41(3), pp. 613–628, doi:10.1007/s10992-011-9208-0.
- [9] Jean-Yves Girard (2001): *Locus Solum: From the rules of logic to the logic of rules*. *Mathematical Structures in Computer Science* 11(3), pp. 301–506, doi:10.1017/S096012950100336X.
- [10] Giulio Guerrieri & Alberto Naibo (2020): *The problem of harmony in classical logic*. In Igor Sedlár & Martin Blichka, editors: *The Logica Yearbook 2019*, College Publications, London.
- [11] Alberto Naibo & Mattia Petrolo (2015): *Are Uniqueness and Deducibility of Identicals the Same?* *Theoria* 81(2), pp. 143–181, doi:10.1111/theo.12051.
- [12] Alberto Naibo, Mattia Petrolo & Thomas Seiller (2016): *On the Computational Meaning of Axioms*. In Juan Redmond, Olga Pombo Martins & Ángel Nepomuceno Fernández, editors: *Epistemology, Knowledge and the Impact of Interaction*, Springer International Publishing, pp. 141–184, doi:10.1007/978-3-319-26506-3\_5.
- [13] Alice Pavaux (2017): *Inductive and Functional Types in Ludics*. In: *26th EACSL Annual Conference on Computer Science Logic, CSL 2017, August 20-24, 2017, Stockholm, Sweden*, pp. 34:1–34:20, doi:10.4230/LIPIcs.CSL.2017.34.
- [14] Alice Pavaux (2017): *Inductive, Functional and Non-Linear Types in Ludics*. Ph.D. thesis, Université Paris 13.
- [15] Frank Pfenning & Rowan Davies (2001): *A judgmental reconstruction of modal logic*. *Math. Struct. Comput. Sci.* 11(4), pp. 511–540, doi:10.1017/S0960129501003322.
- [16] Jan von Plato (2008): *Gentzen's Proof of Normalization for Natural Deduction*. *Bull. Symb. Log.* 14(2), pp. 240–257, doi:10.2178/bsl/1208442829.
- [17] Dag Prawitz (1965): *Natural Deduction: A Proof-Theoretical Study*. Almqvist & Wiksell, Stockholm.
- [18] Peter Schroeder-Heister (2014): *The Calculus of Higher-Level Rules, Propositional Quantification, and the Foundational Approach to Proof-Theoretic Harmony*. *Studia Logica* 102(6), pp. 1185–1216, doi:10.1007/s11225-014-9562-3.
- [19] Peter Schroeder-Heister (2018): *Proof-Theoretic Semantics*. In Edward N. Zalta, editor: *The Stanford Encyclopedia of Philosophy*, spring 2018 edition, Metaphysics Research Lab, Stanford University. Available at <https://plato.stanford.edu/archives/spr2018/entries/proof-theoretic-semantics/>.
- [20] Manfred E. Szabo, editor (1969): *The Collected Papers of Gerhard Gentzen*. North-Holland, Amsterdam.
- [21] Kazushige Terui (2011): *Computational ludics*. *Theor. Comput. Sci.* 412(20), pp. 2048–2071, doi:10.1016/j.tcs.2010.12.026.