GAME THEORETIC ANALYSIS OF CALL-BY-VALUE COMPUTATION

KOHEI HONDA† NOBUKO YOSHIDA‡

ABSTRACT. We present a general semantic universe of call-by-value computation based on elements of game semantics, and validate its appropriateness as a semantic universe by the full abstraction result for call-by-value PCF, a generic typed programming language with call-by-value evaluation. The key idea is to consider the distinction between call-by-name and call-by-value as that of the structure of information flow, which determines the basic form of games. In this way the call-by-name computation and call-by-value computation arise as two independent instances of sequential functional computation with distinct algebraic structures. We elucidate the type structures of the universe following the standard categorical framework developed in the context of domain theory. Mutual relationship between the presented category of games and the corresponding call-by-name universe is also clarified.

1. INTRODUCTION

The *call-by-value* is a mode of calling procedures widely used in imperative and functional programming languages, e.g. [1, 40, 47], in which one evaluates arguments before applying them to a concerned procedure. The semantics of higher-order computation based on call-by-value evaluation has been widely studied by many researchers in the context of domain theory, cf. [46, 47, 31, 42, 19, 53, 16, 17], through which it has become clear that the semantic framework needed to capture the call-by-value computation has a basic difference from the one for call-by-name computation (see [23, 55] for basic introduction to the topic). The difference between the semantics of call-by-value and that of call-by-name in this context may roughly be captured as the difference in the classes of involved functions: in call-by-name, we take any continuous functions between pointed cpos, while, in call-by-value, one takes *strict* continuous functions. The latter is also equivalently presentable as partial continuous functions between (possibly bottomless) cpos. This distinction leads to a basic algebraic difference of the induced categorical universe compared to the call-by-name universe, as has been studied in [16, 19].

The present paper offers a semantic analysis of call-by-value computation from a different angle, based on elements of game semantics. In game semantics, computation is modelled as specific classes of interacting processes (called *strategies*), which, together with a suitable notion of composition, form a categorical universe with appropriate type structures. One may compare this approach to Böhm trees [8] or to sequential algorithms [9] (cf. [29]), in both of which computation is modelled not by set-theoretic functions of a certain kind but by objects with internal structures which reflect computational behaviour of the concerned class of computation. Game semantics has its origin in Logics [15, 10] and has been used for the semantics analysis of programming languages, especially for characterising the notion of sequentiality [12, 45, 4, 27, 43], cf.[44]. By concentrating on specific forms of interaction which should obey a few basic constraints, the approach makes it possible to extract desired classes of interacting processes at the high-level of abstraction, offering suitable semantic universes for varied calculi and programming languages including those with imperative features, cf. [2, 4, 5, 6, 28, 33]. The forms of interaction in these universes are however inherently call-by-name: it has not been clear how the call-by-value computation can be captured in the setting of game semantics, in spite of its equally significant status as a mode of computation.

(†) LFCS, Department of Computer Science, University of Edinburgh, The King's Buildings, Mayfield Road, Edinburgh, EH9 3JZ, UK. e-mail: kohei@dcs.ed.ac.uk. (‡) Computer Science, School of Cognitive and Computing Sciences, University of Sussex, Brighton BN1 9QH. e-mail: nobuko@cogs.susx.ac.uk.

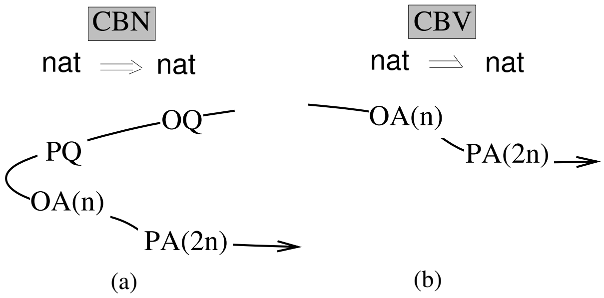

Figure 1

In the present work it will be shown that a general semantic universe of the call-by-value higher-order computation can indeed be simply constructed, employing basic elements of the foregoing game semantics, but with a key difference in the structures of interaction. More specifically, we find that the distinction between call-by-name and call-by-value in game semantics arises as the one in the form of the flow of information. Let us illustrate this point by simple examples. Figure 1 (a) depicts how a function which doubles a given natural number is modelled in the foregoing game semantics ("O" for Opponent, "P" for Player, "A" for Answer, and "Q" for Question). Computation starts when Opponent asks a question on the right, requesting an answer: then Player (the function) asks what the argument is on the left, from which the number is received, and finally it returns to the right to answer the initial question by the double of the received number. In Figure 1 (b), the same function is modelled in the call-by-value game. This time the flow starts at the *left* component, which already carries a value: then the function just returns the answer on the right. One may notice that this means the interaction should start from an *answer*, which might be regarded as an anomaly in the preceding convention in game semantics. However, it turns out that this parameter of games — whether one initiates a game by answers or by questions — is orthogonal to other basic elements of the game semantics, leading to a simple construction of a categorical universe in which representative functional calculi based on call-by-value evaluation can be faithfully interpreted. The independence of the parameter suggests we may obtain a suitable universe to model, say, imperative call-by-value computation by simply altering other parameters, cf. [6]. We also note that the possibility to model "data-driven computation" in contrast to "demand-driven computation" in games is discussed in an early paper on game semantics by Abramsky and Jagadeesan [2].

The main technical contribution of the present work is the validation of the semantic exactness with which the induced universe captures the call-by-value sequential higher-order computation through the full abstraction result for the call-by-value version of PCF [47, 53], a paradigmatic functional calculus. The interpretation is done first in the "intensional" category of games and strategies, in which all compact elements of homsets with appropriate types are definable by the language, then is transferred to its extensional quotient where we obtain the fully abstract model, as has been done in other game-based semantics [4, 27]. The result seems the first one of the kind in this context[1] and is easily extendible to other languages, as we shall indicate in Section 7. We also clarify the relationship between the present universe of games and the corresponding call-by-name universe by showing they are faithfully embeddable to each other. These results indicate, together with the preceding results on call-by-name PCF [4, 27], that the two basic notions of calling procedures in higher-order computation are representable in the game-based semantic framework in an exact way, and that they arise as two independent, though mutually related, semantic universes with equal status (which parallels the findings in the domain theoretic universes, cf. [16]). It is also notable that, as we clarify later, the universe of games for call-by-value computation assumes basic type structures which have arisen through the categorical analysis of

[1] There is an independent and concurrent similar result by Riecke and Sandholm [52] which however does not use game semantics. See Section 7 for discussions.

domain-theoretic universes for call-by-value, or partial, computation, cf.[48, 41, 42, 17], though with a strong intensional flavour. In particular each of the intensional category and its extensional quotient is isomorphic to the Kleisli category of a strong monad on the respective subcategory of total maps, from which such notions as pairing and exponentiation arise (which reminds us of pCPO, the category of possibly bottomless cpos and partial continuous functions). These suggest an abstract notion of "call-by-value computation" may be delineated apart from the standard domain theoretic constructions, cf. [19, 16].

The structure of games we shall use is a conservative extension of the constructions by Hyland and Ong in [27]. When restricted to the class of types they used, our construction is essentially equivalent to games in [27]. In particular their category of games arises as a full subcategory of the call-by-name counterpart of our universe of games, as discussed in Section 6. Having call-by-value as well as call-by-name leads to a general but simple framework of games and strategies from which both call-by-value and call-by-name universes arise by simply restricting concerned classes of types. We hope that these and related constructions would offer a useful tool for further study in this area.

This paper is a full version of the technical summary in [25]. It is based on an earlier full version [26], but substantially revises it in presentation. Full proofs as well as extended discussions are provided. In the rest of the paper, Section 2 introduces the basic notion of games and strategies. Section 3 studies the algebraic structures of the category of games. Section 4 briefly outlines the basic results on its extensional quotient. Section 5 gives interpretation of call-by-value PCF in these two universes, establishing the inequational full abstraction. Section 6 discusses the mutual embeddings between the call-by-name and the call-by-value universes. Section 7 gives discussions and point out future issues. To make the main sections easier to read, many proofs are relegated to Appendices. They in particular include the proofs of the basic properties of the operational structures of call-by-value Hyland-Ong games. The appendices also include a brief introduction to the π-calculus representation of call-by-value strategies.

Acknowledgements. Our deep thanks go to Marcelo Fiore for his suggestions concerning pertinent categorical structures. We heartily thank anonymous referees for their beneficial suggestions. We are grateful to Samson Abramsky, Vincent Danos, Guy McCusker, Pasquale Malacaria, Paul Mellies and Jon Riecke for their comments and/or discussions, and to N. Raja for his hospitality in Bombay.

2. Games and Strategies

Hyland and Ong [28] introduced a notion of *arenas* and *innocent strategies* for their full abstraction result for PCF. Strategies are a class of interacting processes representing the interactive aspects of programs' behaviours, while arenas give a basic notion of types for these processes. In [28], these notions are used to represent the interactive behaviour of call-by-name sequential higher-order functions. Here we give the corresponding notions for representing call-by-value computation. For readability, most of the proofs in this section are given in Appendix A.

We start with a generalisation of the original notion of arenas, which is then restricted to arenas for call-by-value. The construction is related to the notion of sorting of the π-calculus, as discussed in Remark 2.3 later. The following presentation, including that of 2.5, is due to Marcelo Fiore. By a *forest* we mean a directed graph $(\Sigma, \mapsto)$ (Σ is the set of nodes, $\mapsto$ is the relation on Σ representing directed edges) in which every connected component is a tree. A *root* of a forest is the root of some tree in the forest.

2.1. **Definition.**

(i) (pre-arenas) A *pre-arena* is a forest $(\Sigma, \mapsto)$ whose nodes are called *sorts*, ranged over by $x, y, \ldots$, together with the labelling functions $l_1 : \Sigma \to \{\mathrm{O}, \mathrm{P}\}$ and $l_2 : \Sigma \to \{\mathrm{Q}, \mathrm{A}\}$, which satisfies, whenever $x \mapsto x'$ (which we read "x justifies x'"): (1) If $l_1(x) = \mathrm{O}$ then $l_1(x') = \mathrm{P}$, while if $l_1(x) = \mathrm{P}$ then $l_1(x') = \mathrm{O}$ ("Opponent justifies Player, Player justifies Opponent"), and (2) If $l_2(x) = \mathrm{A}$ then $l_2(x') = \mathrm{Q}$ ("Question may justify both Answer and Question, but Answer only justifies Question"). $A, B, C, \ldots$ range over pre-arenas. A sort in a pre-arena is *initial* if it is a root of the underlying forest. $\mathrm{init}(A)$ denotes the set of initial sorts in A. If a sort is (non-)initial in a pre-arena, then an element of a sort is also said (non-)initial.

(ii) (arenas) A *cbv-arena*, or often simply an *arena*, is a pre-arena such that, for each $x \in \mathrm{init}(A)$, we have $l_1(x) = \mathrm{P}$ and $l_2(x) = \mathrm{A}$.

(iii) (operations on pre-arenas) Let A and B be pre-arenas. $\overline{A}$ denotes the pre-arena which is the same as A except the O-P labelling is dualised, i.e. $l_1(x) = \mathrm{O}$ (resp.$l_1(x) = \mathrm{P}$) in A iff $l_1(x) = \mathrm{P}$ (resp. $l_1(x) = \mathrm{O}$) in $\overline{A}$. $A \uplus B$ denotes the pre-arena whose sorts are given by the disjoint union of those of A and B, and whose edges and labelling functions are inherited from A and B. This extends to the arbitrary disjoint union, written e.g. $\uplus_i A_i$.

2.2. **Conventions.** Fix some prearena. If its sort x is labelled by Q (resp. A), then x is a *question* (resp. an *answer*). Similarly, if x is labelled by O (resp. P), then x is *by Opponent* (resp. *by Player*). We may combine these designations by saying x is a *Player question*, *a Player answer*, *an Opponent question*, or *an Opponent answer*, which are often abbreviated as P-question, P-answer, O-question, or O-answer. Correspondingly we often use the combined labels, PQ, PA, OQ, and OA. An answer which is initial is sometimes called a *signal*.

2.3. **Remark.**

(i) Notice pre-arenas in general allow their initial sorts to be answers as well as to be questions (cf. Introduction). An inevitable consequence of this is to have a justification structure in which answers justify questions. It is notable that the same notion arises from the necessity of having sum types in call-by-name games, as discussed by McCusker in [33].

(ii) In the terminology of the π-calculus, a pre-arena corresponds to a specific kind of *sorting* [38], in particular to the input/output sorting studied by Pierce and Sangiorgi [49]. More concretely, a sort in a pre-arena labelled by O, standing for Opponent, corresponds to a sort (in the sense of the π-calculus) the input-only capability, while a sort labelled by P, standing for Player, corresponds to the output-only capability. This is why we used the term "sorting" in [25] for what is now called a pre-arena. The present terminology may however be better to emphasise a specific notion (and form) of sorting which Hyland-Ong games employ; accordingly we now call *arenas* for "types" in [25]. We observe that the difference between call-by-name and call-by-value in interactive behaviours, hence in sorting, is originally discussed in [37], which is indeed the starting point of the present work. We also note that, from the viewpoint of a sorting of the π-calculus, we may as well have graphs instead of forests (as we did in the conference version of the present paper [25]), though this does not affect the resulting operational, and, as a consequence, categorical, structure. We here preferred forests for the sake of the simplicity of the presentation.

(iii) The conference version also used a more general presentation of prearenas in which each sort is a set, instead of a structureless element. This means there are several kinds of actions in one sort, which is useful to concisely present the notion of value passing of ground values. In particular, we can interpret $\mathrm{PCF_V}$ only using arenas which are trees, even though the resulting categories are equivalent. Here we regard sorts as structureless elements, again for the simplicity of presentation.

2.4. **Examples.**

(i) $\mathbf{0}$ is the empty pre-arena, which is an arena by definition. $\mathbf{1}$ is a pre-arena with a (distinguished) unique PA-labelled sort, which is again an arena. nat is made as $\mathbf{1}$ using ω, the set of natural numbers, as sorts. Similarly bool is made by using $\{\mathsf{true}, \mathsf{false}\}$ instead. In this way any set induces an arena, called a *ground arena*.

(ii) $\overline{\mathsf{nat}}$ is the pre-arena with the same sort as nat which is however labelled by OA. $\overline{\mathsf{nat}} \uplus \mathsf{nat}$ is the sorting with two copies of ω, one labelled by OA and another labelled by PA.

(iii) We define $\mathsf{nat} \Rightarrow \mathsf{nat}$ as an arena whose sorts are the disjoint union of (1) a distinguished sort $*$ labelled PA, (2) ω, each labelled OQ and justified by $*$, and (3) $\omega \times \omega$, each labelled PA, where each $\langle i, j \rangle \in \omega \times \omega$ is justified by $i \in \omega$.

A sequence of elements of some set X can be considered as a partial function from ω to X defined for a finite initial segment of ω and undefined for the rest. We call this associated initial segment, *indices* of the sequence. As an example, the sequence abc has $\{0, 1, 2\}$ as its indices. ε denotes the empty sequence. The following notion represents, essentially speaking, a history of interactions between two processes, one called Player and another called Opponent.

2.5. **Definition.** (action sequence) Let A and B be arenas. An *action sequence from A to B* is a sequence of sorts in $\overline{A} \uplus B$, say $x_0x_1...x_{n-1}$ $(n \geq 0)$, together with the *justification relation* $\curvearrowright$ on its indices (saying "i justifies j" when $i \curvearrowright j$), satisfying: (1) If $i \curvearrowright j$, then both $i \lneq j$ and $x_i \mapsto x_k$, (2) $(i \curvearrowright k \ \wedge \ j \curvearrowright k) \ \Rightarrow \ i = j$ ("each action has a unique justifier if any"), (3) x_0 is initial in the $\overline{A}$-component of $\overline{A} \uplus B$. Other than that, there is at most one unjustified $j \neq 0$ and then x_j is initial in the B component, (4) $(i \curvearrowright j \ \wedge \ i \curvearrowright k \ \wedge \ x_{j,k}$ is an Answer$) \ \Rightarrow \ j = k$ ("a question is answered at most once"), and (5) if x_{i-1} is by Opponent (resp. Player) then x_i is by Player (resp. Opponent) for $1 \leq i \leq n$ ("Opponent and Player strictly alternate"). $s, s', \ldots$ range over action sequences. $s^{A \multimap B}$ denotes s is from A to B.

2.6. **Notation.**

(i) We write $\underline{s}$ for the sequence underlying s, $\curvearrowright_s$ or often simply $\curvearrowright$ for the associated justification relation, and $|s|$ for the length of $\underline{s}$. For $s^{A \multimap B}$, $s \restriction A$ (resp. $s \restriction B$) denotes the projection of the underlying sequence onto the first (resp. second) component, together with $\curvearrowright$ inherited from s (which may not be an action sequence). We often simply write e.g. $s \stackrel{\text{def}}{=} x_0..x_{n-1}$, leaving the associated $\curvearrowright$ implicit. In spite of this, prefix and equality are considered by incorporating justification relation.

(ii) An index of an action sequence is often called an *action*. An action (say i) may be denoted by the associated element in the sequence (say x_i) if no confusion arises. An action i in an action sequence $x_0..x_{n-1}$ is *by Opponent*, *by Player*, *an O-question*, *an O-answer*, *a P-question*, *a P-answer*, if x_i is such. If an action is by Opponent (resp. by Player), we often say it is an *O-action* (resp. *P-action*). An action in an action sequence is *free* if it is not justified. An action i in $s^{A \multimap B} \stackrel{\text{def}}{=} x_0..x_{n-1}$ is *from A* (resp. *from B*) if x_i is from the first (resp. second) component of the sum $\overline{A} \uplus B$, sometimes written x_i^A (resp. x_i^B).

Action sequences are further constrained by two conditions below. The first condition, *bracketing*, specifies the proper nesting of call-return sequences in functional computation. The second condition, *visibility*, says that an action can only be done with respect to its context called its "view", which is a specific subsequence of the preceding action sequence.

2.7. **Definition.**

(i) (bracketing condition) Say an action j *answers* another action i if $i \curvearrowright j$ and j is an answer. Then an action sequence is *well-bracketed* when a later asked question is always answered first, that is, given an action sequence $x_0..x_{n-1}$, if i is answered by j and $i \lneq k \lneq j$ with x_k being a question, then k should be answered by l such that $l \lneq j$.

(ii) (views and visibility) Let $s \stackrel{\text{def}}{=} x_0..x_{n-1}$ be an action sequence. Define $\mathsf{PV}(s)$ as:

(pv0): $\mathsf{PV}(\varepsilon) = \emptyset$,
(pv1): $\mathsf{PV}(sx_i) = \{i\}$ if x_i is a free O-action,
(pv2): $\mathsf{PV}(s_0x_is_1x_j) = \mathsf{PV}(s_0) \cup \{i, j\}$ when $i \curvearrowright j$ and x_j is an O-action, and
(pv3): $\mathsf{PV}(s_0x_i) = \mathsf{PV}(s_0) \cup \{i\}$ if x_i is a P-action,

and $\mathsf{OV}(s)$ as:

(ov0): $\mathsf{OV}(\varepsilon) = \emptyset$,
(ov1): $\mathsf{OV}(sx_i) = \{0, i\}$ when x_i is a free P-action,

(ov2): $\mathsf{OV}(s_0 x_i s_1 x_j) = \mathsf{OV}(s_0) \cup \{i, j\}$ when $i \curvearrowright j$ and x_j is a P-action, and

(ov3): $\mathsf{OV}(s_0 x_i) = \mathsf{OV}(s_0) \cup \{i\}$ if x_i is an O-action.

Then s *satisfies the visibility condition* when, for each prefix $s' x_i$ of s, *either* i is free *or*, if not, $j \curvearrowright i$ and $j \in \mathsf{PV}(s')$ (resp. $j \in \mathsf{OV}(s')$) if i is a P-action (resp. an O-action). In such a case we say each action in s is *visible from its view*. Under the same s and assuming $\mathsf{PV}(s) = \{i_0, i_1, ..., i_{m-1}\}$ (resp. $\mathsf{OV}(s) = \{i_0, i_1, ..., i_{m-1}\}$) with $i_k \lneq i_{k+1}$ for each k, we set $\ulcorner s \urcorner$ (resp. $\llcorner s \lrcorner$) as the sequence $x_{i_0}..x_{i_{m-1}}$ together with the relation $\curvearrowright$ on the index given by: $k \curvearrowright l$ iff $i_k \curvearrowright i_l$ in s. $\ulcorner s \urcorner$ (resp. $\llcorner s \lrcorner$) is called the *P-view* (resp. *O-view*) of s while $\ulcorner \cdot \urcorner$ and $\llcorner \cdot \lrcorner$ are called the *view functions.*

(iii) (legal position) An action sequence is *legal* when it is well-bracketed and satisfies the visibility condition. A legal action sequence is sometimes called a *legal position.*

Clearly a prefix of a legal action sequence is again legal. We can also show the P-view and the O-view of a legal action sequence are legal (see Appendix A.2). We shall often call a legal action sequence which is its own P-view (resp. O-view), a *P-view* (resp. an *O-view*). Another notable property of the legal action sequences follows, which will be frequently used from now on.

2.8. **Proposition.** (switching condition) If $x_0..x_i x_{i+1}..x_{n-1}$ is an action sequence from A to B, if one of x_i or x_{i+1} is from A and another is from B, then x_{i+1} is necessarily a P-action.

PROOF: See Appendix A.3. □

The main definition of this section follows.

2.9. **Definition and Proposition.** (strategy) Let A and B be arenas. An *innocent strategy from A to B*, or simply a *strategy from A to B*, is a prefix-closed set σ of legal action sequences from A to B such that:

(determinacy) For s ending with an O-action, $sx, sy \in \sigma$ implies $sx = sy$.

(contingency completeness) If $s \in \sigma$ is empty or it ends with a P-action, and if $s' \stackrel{\text{def}}{=} sx$ is legal, then $s' \in \sigma$.

(innocence) If $s_1 x, s_2 \in \sigma$, x is a P-action, and $\ulcorner s_1 \urcorner = \ulcorner s_2 \urcorner$, then $\ulcorner s_2 y \urcorner \in \sigma$ such that $\ulcorner s_1 x \urcorner = \ulcorner s_2 y \urcorner$ (so in particular $x = y$).

$\sigma, \tau, \ldots$ range over strategies. We write $\sigma : A \to B$ when σ is a strategy from A to B. Given two strategies $\sigma, \tau : A \to B$, we write $\sigma \sqsubseteq \tau$ when $\sigma \subset \tau$. For a strategy σ, the *innocent function of* σ, denoted f_σ, is the partial function over the set of P-views from A to B such that $f_\sigma(s) = s'$ iff $s, s' \in \sigma$, s is odd-length, and s is a prefix of s' such that $|s| = |s'| - 1$. Then we have:

(i) $\sigma \sqsubseteq \tau$ if and only if $f_\sigma \subset f_\tau$.

(ii) For each A and B, the set of innocent strategies from A to B ordered by $\sqsubseteq$ is an algebraic cpo, indeed a dI-domain.

The proof of the clauses (i) and (ii) is just as in Hyland and Ong [27], using (i) to establish (ii). For completeness we list the proof in Appendix A.10. We note that we could have first formulated the arena $A \rightleftharpoons B$, then define strategies based on such arenas, as is done for the case of CBN games [27]. However in the CBV games, the arena $A \rightleftharpoons B$ (which we shall define in Definition 3.6) adds an additional move to what we originally have A and B, and, as a result, the composition of strategies becomes less straightforward. For more discussions, see our discussions after Definition 3.6. Simple examples of strategies follow.

2.10. **Examples.**

(i) (undefined) For any arenas A and B, there is a strategy from A to B whose innocent function is totally undefined so that it is least w.r.t. the ordering $\sqsubseteq$. We write this strategy $\perp_{A \to B}$.

(ii) (ground value) Let A be a ground arena. Then the set of strategies from $\mathbf{1}$ to A which is not $\bot$ is in bijection to the set of nodes in A, responding to the unique initial O-signal at $\mathbf{1}$ by selecting a specific node of S^A, and no more action is possible.

(iii) (first-order function) The set of strategies from nat to nat precisely correspond to the set of partial functions from ω to ω, because for each initial O-action $n \in \omega$, $\sigma : \mathsf{nat} \rightarrow \mathsf{nat}$ either does not react or returns some natural number, and there is no further action.

(iv) (higher-order function) A strategy σ from $\mathsf{nat}{\Rightarrow}\mathsf{nat}$ to nat, corresponding to the behaviour of an open call-by-value PCF-term, $x : \iota \rightarrow \iota \triangleright \mathsf{succ}(x3) : \iota$, has the following behaviour: after receiving a signal at $\mathsf{nat}{\Rightarrow}\mathsf{nat}$, which gives a function to σ, it asks the result of applying 3 to that function, and, on receiving the answer, returns its successor to the co-domain.

(v) (identity strategy) Let A be an arbitrary arena. Then the *identity on* A, denoted id_A, is a strategy from A to A (where we write A^{l} and A^{r} to denote the first and the second components of the pre-arena $\overline{A} \uplus A$), defined inductively as follows: (1) $\varepsilon \in \mathsf{id}_A$, (2) if $s \in \mathsf{id}_A$ is even-length, then for each sx which is legal we have $sx \in \mathsf{id}_A$ and (3) if $sx \in \mathsf{id}_A$ is odd-length and $s \restriction A^{\mathrm{l}} = s \restriction A^{\mathrm{r}}$, then $sxy \in \mathsf{id}_A$ such that $sxy \restriction A^{\mathrm{l}} = sxy \restriction A^{\mathrm{r}}$ (notice, thus, x and y are the same value projected differently). We can check inductively that $s \in \mathsf{id}_A$ with s even length implies $s \restriction A^{\mathrm{l}} = s \restriction A^{\mathrm{r}}$ (so this condition in (3) is redundant). This means id_A reacts to any O-action by copying it to another component, including the justification. Such a behaviour is often called *copy-cat.* Clearly the behaviour is contingently complete and deterministic. For innocence, notice (3) immediately implies Player acts with the same action (i.e. the dual of the last action) for the same P-view; but the justification to that action is the O-action immediately preceding the P-action which justifies the last O-action, so the relative position of that action in the view is always the same, as required. As we shall show soon, id_A does function as identity.

Strategies denote a certain kind of deterministic processes, and are, as such, precisely representable as (name passing) synchronisation trees, see Appendix B. The presentation is often useful for describing, and reasoning about, strategies: indeed the full abstraction result was originally obtained in this setting [24]. The following inductive definition of composition of strategies, given after a lemma, is suggested by such presentation, which coincides with, on the one hand, various presentations of composition of strategies in Hyland-Ong games, and, on the other hand, the standard expansion law on name passing processes [35, 39].

2.11. **Lemma.** (zipper lemma) Legal action sequences $s_1^{A\rightarrow B}$ and $s_2^{B\rightarrow C}$ are *composable*, written $s_1 \asymp s_2$, when $s_1 \restriction B = s_2 \restriction B$. Suppose s_1 and s_2 are composable. Then *either* (1) $s_1 \restriction B = s_2 = \varepsilon$, (2) both end with actions from B, *or* (3) s_1 (resp. s_2) ends with an action from A (resp. from C) and s_2 (resp. s_1) ends with an action from B.

PROOF: See Appendix A.11. $\square$

2.12. **Definition.** (composition) Given $\sigma : A \rightarrow B$ and $\tau : B \rightarrow C$, define $\sigma;\tau$, the *composition of* σ *and* τ (in this order), as:

$$\sigma;\tau \stackrel{\text{def}}{=} \{s_1;s_2 \mid s_1 \in \sigma,\ s_2 \in \tau,\ s_1 \asymp s_2\}$$

where we define $s_1^{A\rightarrow B};s_2^{B\rightarrow C}$ for legal s_1 and s_2 such that $s_1 \asymp s_2$ as an action sequence whose underlying sequence is given inductively: (1) $\varepsilon;\varepsilon \stackrel{\text{def}}{=} \varepsilon$, (2) $s_1x^B;s_2y^B \stackrel{\text{def}}{=} s_1;s_2$, (3) $s_1x^A;s_2 \stackrel{\text{def}}{=} (s_1;s_2)\cdot x^A$ and $s_1;s_2x^C \stackrel{\text{def}}{=} (s_1;s_2)\cdot x^C$, and whose justification relation is such that it satisfies $(s_1;s_2) \restriction A = s_1 \restriction A$ and $(s_1;s_2) \restriction C = s_2 \restriction C$.

Remark. Two clauses of (3) above are not ambiguous by Lemma 2.11. That the resulting sequence is an action sequence (in particular it is strictly alternating) is because of the switching condition.

2.13. **Example.** (identity) As an example, we consider the composition of id_A and $\sigma : A \to B$, where id_A is the identity strategy given in Example 2.10 (v), and show the result indeed coincides with σ. We first note that, for each legal sequence s_2 from A to B, there is a unique even-length $s_1 \in \mathsf{id}_A$ such that $s_1 \upharpoonright A^{\mathrm{r}} = s_2 \upharpoonright A$ (therefore $s_1 \upharpoonright A^{\mathrm{l}} = s_2 \upharpoonright A$ too), which is easy by induction on $|s|$. So suppose $s_2 \in \sigma$ and take such $s_1 \in \mathsf{id}_A$. We show $s_1 ; s_2 = s_2$ by induction on the length of s_2. In the base case, we have $s_2 = \varepsilon$, so just take $s_1 = \varepsilon$, so that $s_1 ; s_2 = \varepsilon = s_2$, as required. For induction, assume $s_2 = s_2' x$. By induction hypothesis there is even-length $s_1' \in \mathsf{id}_A$ such that $s_1' ; s_2' = s_2'$. Below we write $\overline{x}$ for the move which interacts with x (when projected, they are the same sort), to help the understanding.

(x is from B.) Then simply take $s_1 \stackrel{\mathrm{def}}{=} s_1'$, from which we get: $s_1 ; s_2 \stackrel{\mathrm{def}}{=} s_1' ; s_2' x = (s_1' ; s_2') \cdot x = s_2' \cdot x$. Since the justification on the B-component does not change, the result coincides with s_2.

(x is an P-action from A.) Then take $s_1 \stackrel{\mathrm{def}}{=} s_1' \overline{x}^{A^{\mathrm{r}}} x^{A^{\mathrm{l}}} \in \mathsf{id}_A$ such that $s_1' \overline{x}^{A^{\mathrm{r}}} \upharpoonright A^{\mathrm{r}} = s_2 \upharpoonright A$ (this decides s_1 uniquely). We now calculate: $s_1 ; s_2 \stackrel{\mathrm{def}}{=} s_1' \overline{x}^{A^{\mathrm{r}}} x^{A^{\mathrm{l}}} ; s_2' x^A = (s_1' \overline{x}^{A^{\mathrm{r}}} ; s_2' x^A) \cdot x^{A^{\mathrm{l}}} = (s_1' ; s_2') \cdot x^A = s_2' \cdot x$. (notice only an action from A^{r} in id_A and that from A in σ interact). The justification to the last x is as in s_1, that is as in s_2, as required.

(x is an O-action from A.) We take $s_1 \stackrel{\mathrm{def}}{=} s_1' x^{A^{\mathrm{l}}} y^{A^{\mathrm{r}}} \in \mathsf{id}_A$ such that $s_1' x^{A^{\mathrm{l}}} \upharpoonright A^{\mathrm{l}} = s_2 \upharpoonright A$. Then by a similar calculation (which we omit), we know $s_1 ; s_2 = s_2$, as required.

Thus we know $\mathsf{id}_A ; \sigma \supset \sigma$. For another direction, we simply note that, if $s_1 \in \mathsf{id}_A$ is composable with $s_2 \in \sigma$, then such s_1 is either the even-length sequence we used above or its prefix, so that all resulting sequences are generated by the induction above (due to the prefix-closure, cf. Proposition 2.14 (i) below), concluding the proof. $\square$

2.14. **Proposition.**

(i) Given $\sigma : A \to B$ and $\tau : B \to C$, $\sigma ; \tau$ is an innocent strategy from A to C.

(ii) ; is associative and is with the left-right identity for each A.

(iii) ; is bi-continuous with respect to the ordering $\sqsubseteq$.

Proof: See Appendix A.14. $\square$

We now have a category.

2.15. **Definition and Proposition.** The category $\mathcal{CBV}$ is given by the following data:

- Objects: cbv-arenas.
- Arrows: strategies, composed by ; and ordered by $\sqsubseteq$.

Then $\mathcal{CBV}$ is enriched over CPO, the category of possibly bottomless cpos and continuous functions. Moreover with respect to the induced ordering, each homset owns a least element $\perp$ with respect to which the composition is left strict, that is, for each $\perp : C \to A$ and $\sigma : A \to B$, we have $\perp ; \sigma = \perp$. Finally $\mathbf{0}$ is initial and weakly terminal in $\mathcal{CBV}$.

Proof: By the structure of CPO, enrichment means the order in each homset gives a cpo and composition is continuous, both of which we already know from Proposition 2.14. For strictness, notice $s \in \perp : C \to A$ iff $s \upharpoonright C \in \{\varepsilon\} \cup \{x \mid x \text{ is an initial O-signal of } \overline{C}\}$ and $s \upharpoonright A = \varepsilon$, hence $s \in \perp ; \sigma \;\Leftrightarrow\; s \in \perp$. $\mathbf{0}$ is initial since if $\tau : \mathbf{0} \to A$ then τ cannot contain any initial O-signal, which means we can determine $\tau = \{\varepsilon\}$. $\mathbf{0}$ is weakly terminal, i.e. for each A there is an arrow from A to $\mathbf{0}$, because at least $\perp : A \to \mathbf{0}$ exists (note this means any object is weakly terminal). $\square$

We study this universe in detail in the next section.

3. Intensional Universe

Type structures of a semantic universe offer the basic articulation of its algebraic properties needed, for example, for interpreting various programming languages in it. This section clarifies the basic type structure of $\mathcal{CBV}$ in the light of the distinction between total and partial maps. We first introduce the notion of totality, cf. [20].

3.1. **Definition.** σ is *total* when $\tau;\sigma = \bot$ implies $\tau = \bot$. We write $\sigma\Downarrow$ when σ is total.

Immediately identities are total, if σ and τ are total then so is $\sigma;\tau$, if σ is not total then $\sigma;\tau$ is not total. From the first and the last we also deduce isomorphisms are total. Also if σ_1 is total and $\sigma_1 \sqsubseteq \sigma_2$, then σ_2 is total [using the monotonicity of composition, we have $\tau;\sigma_2 = \bot \;\Rightarrow\; \tau;\sigma_1 = \bot \;\Rightarrow\; \tau = \bot$]. There are a couple of different characterisations of total maps.

3.2. **Proposition.** The following four statements are equivalent.

(i) **(def)** $\sigma : A \to B$ is total.

(ii) **(elm)** $\forall \tau : \mathbf{1} \to A.\,(\tau \neq \bot \;\Rightarrow\; \tau;\sigma \neq \bot)$.

(iii) **(wpb)** The square $\langle \mathbf{0} \to A \xrightarrow{\sigma} B,\ \mathbf{0} \to \mathbf{0} \to B\rangle$ is a weak pullback.

(iv) **(beh)** σ immediately emits a P-signal for each initial O-signal.

Remark. Among these characterisations, **(elm)** relates Definition 3.1 to the standard notion of totality, noticing that the set of total maps from $\mathbf{1}$ to A gives a possibly bottomless cpo for each A, which may be regarded as "elements" of A (note also the only non-total map from $\mathbf{1}$ to A is the bottom map). **(wpb)** is categorically most natural. **(beh)** is the basic behavioural characterisation. Notice how the same notion arises equivalently in set-theoretic, arrow-theoretic and behavioural forms.

To prove Proposition 3.2 we need the following notion.

3.3. **Definition and Lemma.** (insensitive arrows) Say an arrow $\tau : C \to A$ is *insensitive*[2] if it factors through $\mathbf{0}$, i.e. $\tau : C \xrightarrow{\tau'} \mathbf{0} \to A$ for some τ'. Then $\tau : C \to A$ is insensitive iff we have $\tau;\sigma = \tau$ for any $\sigma : A \to A$. Alternatively, $\tau : C \to A$ is insensitive iff for each $s \in \tau$ we have $s \upharpoonright A = \varepsilon$.

Proof: Write i_A for the unique arrow from $\mathbf{0}$ to A, then we can set $\tau = \tau';i_A$ for an insensitive τ. For the first statement, the "if" direction is direct from the definition. For the "only if" direction, we calculate: $\tau;\sigma \stackrel{\text{def}}{=} \tau';i_A;\sigma \;=\; \tau';i_A = \tau$, as required. For the second characterisation, the "if" direction is, by composing τ with say $\bot_{A\to A}$, we have $\tau;\bot_{A\to A} = \tau$ so τ factors through $\mathbf{0}$. For the "only if" direction, suppose τ is insensitive, then $\tau = \tau';i_C$, which means $s \in \tau$ is written $s = s_1;\varepsilon$ for some $s_1 \in \tau'$, but then $s \upharpoonright C = \varepsilon \upharpoonright C = \varepsilon$, as required. □

Notice, at the behavioural level, insensitivity means the strategy does nothing at the co-domain. A typical example of insensitive arrows is $\bot$. We now prove Proposition 3.2.

Proof of Proposition 3.2. We prove the following chain of implications:

$$\textbf{(def)}\Rightarrow\textbf{(elm)}\Rightarrow\textbf{(beh)}\Rightarrow\textbf{(wpb)}\Rightarrow\textbf{(def)}$$

which establish their equivalence.

(def)⇒(elm): Immediate.

(elm)⇒(beh): Suppose **(elm)** holds for σ, and take $\tau \neq \bot : \mathbf{1} \to A$ which immediately returns a signal at A after the unique signal at $\mathbf{1}$ and is undefined thereafter. In that case if $\tau;\sigma \neq \bot : \mathbf{1} \to B$ then σ itself should immediately returns a signal at any initial signal at A, thus the behavioural characterisation holds.

(beh)⇒(wpb): Suppose σ satisfies **(beh)**. First clearly $\langle \mathbf{0} \to A \xrightarrow{\sigma} B,\ \mathbf{0} \to \mathbf{0} \to B\rangle$ commutes because $\mathbf{0}$ is initial in $\mathcal{CBV}$. Suppose $\langle C \xrightarrow{\tau} A \xrightarrow{\sigma} B,\ C \xrightarrow{\tau'} \mathbf{0} \to B\rangle$ commutes. We first show that the existence of a weak universal arrow is equivalent to τ being insensitive. Indeed, if τ is insensitive,

[2] We could not find a standard terminology for such an arrow so far.

then we know, by Lemma 3.3, $\tau = \tau;\sigma;\bot_{B\to A} = \tau';i_B;\bot_{B\to A} = \tau';i_A$, which shows we can take τ' as a weak universal arrow. The other direction is by definition. Thus we are only to show τ is insensitive. But if τ is not insensitive, it gives, at some point, a P-signal at A, to which σ immediately reacts to B, which implies $\tau;\sigma$ is not insensitive, so τ should be insensitive, hence done.

(wpb)⇒(def): Suppose **(wpb)** holds. Suppose $\tau;\sigma = \bot_{C\to B}$ for $\tau : C \to A$. Then τ is insensitive by **(wpb)**, but then we can apply any arrow of type $B \to A$, say $\bot_{B\to A}$, to get $\tau = \tau;\sigma;\bot_{B\to A} = \bot_{C\to B};\bot_{B\to A} = \bot_{C\to A}$, so σ should be total, as required. □

Other than identities and isomorphisms, the following gives basic examples of total maps.

3.4. **Examples.**

(i) The unique arrow from $\mathbf{0}$ to any type is total, by definition. Just as the unique function from the empty set is, by definition, total. Just as well, there is no total map to $\mathbf{0}$ except from itself.

(ii) There is a unique total map denoted $!_A : A \to \mathbf{1}$ for each A, given as that which reacts to the initial action (if any) by the unique P-answer at $\mathbf{1}$. By switching condition (cf. Proposition 2.8), no more action is possible.

(iii) A strategy $\sigma : \mathsf{nat} \to \mathsf{nat}$ is total if and only if the underlying number-theoretic function is total. Similarly for any ground arenas.

3.5. **Definition.** $\mathcal{CBV}_{\mathsf{t}}$ denotes the category of arenas and total strategies, which again CPO-enriches.

The relationship between total maps and usual (often called *partial*) maps is clarified by the notion of *lifting*. Write $A_\bot$ for the arena given by adding two sorts to A, one initial which justifies the other one, the latter justifying all $x \in \mathrm{init}(A)$, the rest as in A (see below).

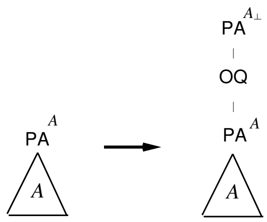

Figure 2

Then we can see the set of *total* arrows from $\mathbf{1}$ to $A_\bot$ is in bijection to the set of *partial* arrows from $\mathbf{1}$ to A, indeed they are order-isomorphic. These two are mediated by two copy-cat like strategies, $\mathsf{up} : A \to A_\bot$ and $\mathsf{dn} : A_\bot \to A$, with obvious behaviours (up reacts to an initial action at A of the domain by going through two added sorts of $A_\bot$, does the dual action of the initial action at the top of A in the co-domain, then go into the copy-cat behaviour between the non-initial actions in the domain and the corresponding actions in the co-domain, specified just as in 2.10 (v) for identity strategies: dn then just does the dual). In a familiar way this induces the adjoint situation and the associated monad. We now present basic facts about $\mathcal{CBV}_{\mathsf{t}}$ and the monad on it, including the above mentioned bijection. We use the following operations on arenas.

3.6. **Definition.** (operations on arenas) Below let A and B denote arenas.

(i) $A_\bot$, which is already discussed, is formally given as: first take the prearena $\mathbf{1} \uplus \overline{\mathbf{1}} \uplus A$, then add one edge from the initial sort of $\mathbf{1}$ to that of $\overline{\mathbf{1}}$, and one from the initial sort of $\overline{\mathbf{1}}$ to each initial sort of A.

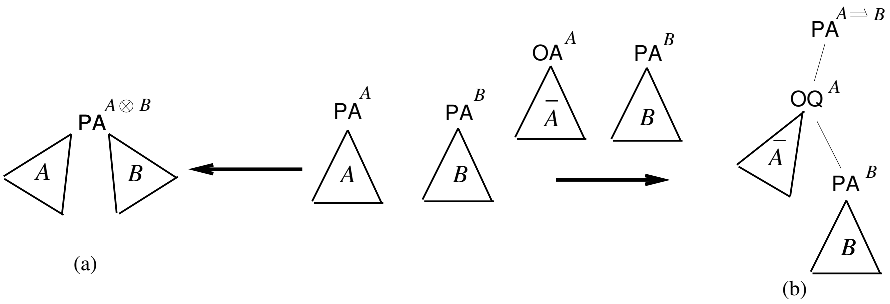

Figure 3

(ii) $A \otimes B$ is an arena which is constructed thus: for each component tree A_i in A and B_j in B, let x and x' be their respective initial sorts, and A'_i and B'_j be the result of taking off these initial sorts from A_i and B_j, respectively. Take the prearena $\{\langle x, x'\rangle\} \uplus A'_i \uplus B'_j$, and add an edge from $\langle x, x'\rangle$ to each initial sort of A'_i and B'_j. Let the resulting arena be $C_{i,j}$. Then $A \otimes B \stackrel{\text{def}}{=} \uplus_{i,j} C_{i,j}$.

(iii) $A \Rightarrow B$ is an arena which is constructed thus: for each maximal tree A_i in A, we first form $\overline{A_i} \uplus B$, relabel the initial sort of $\overline{A_i}$ as (Opponent) Question, then add an edge from it to each initial sort of B (notice this is indeed a prearena). Let the result be denoted C_i. We then take $\mathbf{1} \uplus (\uplus_i C_i)$, and finally add an edge from $\mathbf{1}$ to each initial sort of C_i. The resulting arena is $A \Rightarrow B$.

In Figure 3 we depict the formation of $A \otimes B$ and $A \Rightarrow B$ from A and B each of which is a tree, i.e. has a single initial sort. Notice, in $A \Rightarrow B$, we first take the dual of A, then replace the label of the initial sort to the opponent question. The behavioural idea underlying the construction is: (1) Player first tells "I am here" at the initial sort, then (2) it receives a question from Opponent, and (3) after interaction at the A component (if any), it answers at the initial sort of B, possibly continuing interaction at B (and back to A when necessary). Notice how the construction closely corresponds to the sorting $\overline{A} \uplus B$ where strategies inhabit, except we have an additional initial sort. Moreover we can verify that the switching between $\overline{A}$ and B occurs only by Player except at an initial sort of $\overline{A}$, the latter being always by Opponent by definition. This indicates that we could have defined cbv-strategies from A to B based on the arena $A \Rightarrow B$, rather than based on prearenas. However in the definition of composition using this variant form of strategies, we first need to take off the initial action, then need to compose a question with an answer, so that there is a non-trivial manipulation of strategies before composition. This is because the original structure of information flow can only be preserved by adding an additional action to make it into a "value" (in contrast to the situation in call-by-name games, cf. [27]). The simplicity of composition in Definition 2.12 may indicate the naturalness of the present formulation, though formally two forms induce the isomorphic category.

3.7. **Proposition.**

(i) $\mathcal{CBV}_{\mathtt{t}}$ has the initial object $\mathbf{0}$, and has finite products with $\otimes$ giving a binary product (which in $\mathcal{CBV}_{\mathtt{t}}$ we often denote $\times$) which induces a bi-functor which CPO-enriches, and $\mathbf{1}$ giving a terminal object.

(ii) The inclusion functor F from $\mathcal{CBV}_{\mathtt{t}}$ to $\mathcal{CBV}$ has the right adjoint T, with $T(A) = A_\perp$, the unit $\eta_A = \mathsf{up}$, and the co-unit $\epsilon = \mathsf{dn}$, which CPO-enriches. The monad $\langle T, \eta_A, T(\mathsf{dn})\rangle$ is denoted $\mathbf{T}$. Then $\mathbf{T}$ has a tensorial strength $\mathtt{st}_{A,B} : A \times TB \to T(A \times B)$ and a co-strength (in the sense of [50]) $\mathtt{st}'_{A,B} : TA \times B \to T(A \times B)$.

(iii) The Kleisli category of $\mathbf{T}$ on $\mathcal{CBV}_{\mathtt{t}}$ is isomorphic to $\mathcal{CBV}$. The representation of $\sigma : A \to B$ in the Kleisli category, given as $\mathsf{up}; T(\sigma) : A \to B_\perp$, is written $\sigma^\dagger$, while the inverse representation of a total map $\sigma : A \to B_\perp$ in $\mathcal{CBV}$, given as $\sigma; \mathsf{dn} : A \to B$, is denoted $\sigma_\dagger$.

Proof: **(i)** We already know **0** is initial and **1** is terminal. As to products, projections are given as the following copy-cat: $\pi_1 : A \otimes B \to A$ acts, when receiving an initial O-signal, which is a tuple of an initial signal at A and that at B, extracts the A-component part of that tuple and does the dual action at A in the co-domain: after that it just does the copy-cat between A in the co-domain and A in the domain. $\pi_2 : A \otimes B \to B$ behaves in the symmetric way. Before describing a pairing, we note a basic fact about legal sequences from $C \to A \otimes B$: if $s^{C \to A \otimes B}$ is legal, then the projection onto $A \otimes B$ has the property that it is always Opponent who switches between A and B components. See Appendix A.16 for the proof. Notice this means that a P-view is always — except the P-signal at $A \otimes B$ — either that from C to A or that from C to B. Now the (total) pairing $\langle \sigma, \tau \rangle : C \to A \otimes B$ has the following behaviour: for each signal at C, it reacts at the co-domain by combining the reactions by σ and τ to the signal at C (note that the totality is essential here); then, for each P-view ending with an O-action from A (thus essentially a P-view from C to A), it reacts as σ, similarly it reacts as τ on P-views ending with an O-action from B. The commutativity is now clear (see Appendix B.10 (i)). For universality, by the above fact on P-views, the behaviour of a strategy from C to $A \otimes B$ is completely determined by how it reacts to the C-A component and the C-B component of its past interactions. Thus the equations $\delta; \pi_1 = \sigma$ and $\delta; \pi_2 = \tau$ uniquely determine δ, as required. Finally the construction of pairing is clearly continuous at both variables, from which CPO-enrichment follows.

(ii) Given $\sigma : A \to B$, define $T(\sigma) : TA \to TB$ as simply that which reacts to the initial signal of TA by going to the initial signal of TB, then reacts to the question at the second sort of TB by doing the answer at the second sort of TA, then behaves precisely as σ (notice the construction is obviously continuous w.r.t. $\sqsubseteq$). We can now easily show the adjointness. For each total $\sigma : A \to TB$, take $\sigma' \stackrel{\text{def}}{=} \sigma; \mathsf{dn} : A \to B$, then by construction $\mathsf{up}; T(\sigma') = \sigma$ (because σ is total), while the universality holds because, if for $\tau : A \to B$ such that $\mathsf{up}; T(\tau) = \sigma$, then $\tau = \mathsf{up}; T(\tau); \mathsf{dn} = \sigma; \mathsf{dn}$ (the first equation is by construction), hence done. We next consider strength. We define $\mathtt{st}_{A,B} : A \times TB \to T(A \times B)$ as a total strategy which, after the initial O-signal at $A \times TB$, does two (uniquely determined) consecutive actions at $T(A \times B)$, then returns to TB (the right component of the domain) to act twice, and finally becomes a copy-cat between A and B component. $\mathtt{st}'_{A,B} : TA \times B \to T(A \times B)$ is its dual. That these satisfy defining equations can again become evident when one uses process representation, see B.10.

(iii) This is a classical result by Kleisli, noticing F is identity on objects. We observe that we could have used the Kleisli triple is $\langle T, \mathsf{up}, (\cdot)^{\dagger} \rangle$ for proving (ii) above. ☐

We note that $\mathcal{CBV}_{\mathtt{t}}$, hence $\mathcal{CBV}$, also has a co-product denoted $A \oplus B$ which is simply $A \uplus B$ (the injections are simple copy-cats, and that the construction of $A \otimes B$ and $A \oplus B$ can be easily extended to arbitrary products and co-products), though we do not use them. By Proposition 3.7 we obtain:

3.8. **Definition and Proposition.**

(i) (partial pairing [42]) Given $\sigma_1 : C \to A$ and $\sigma_2 : C \to B$, their *left pairing*, $\langle\!\langle \sigma_1, \sigma_2 \rangle\!\rangle_l : C \to A \otimes B$, and the *right paring*, $\langle\!\langle \sigma_1, \sigma_2 \rangle\!\rangle_r : C \to A \otimes B$ are given as: $\langle\!\langle \sigma_1, \sigma_2 \rangle\!\rangle_l \stackrel{\text{def}}{=} (\langle \sigma_1^{\dagger}, \sigma_2^{\dagger} \rangle; \psi_{A,B})_{\dagger}$ and $\langle\!\langle \sigma_1, \sigma_2 \rangle\!\rangle_r \stackrel{\text{def}}{=} (\langle \sigma_1^{\dagger}, \sigma_2^{\dagger} \rangle; \tilde{\psi}_{A,B})_{\dagger}$ where $\psi_{A,B} = \mathtt{st}'_{TA,B}; T(\mathtt{st}_{A,B}; \mathsf{dn})$ and $\tilde{\psi}_{A,B} = \mathtt{st}_{A,TB}; T(\mathtt{st}'_{A,B}; \mathsf{dn})$.

(ii) (premonoidal tensor [50]) Given A, we define $A\otimes$ and $\otimes A$ by: (i) $A \otimes \cdot B = A \otimes B$ and $B \cdot \otimes A = B \otimes A$, and (ii) $A \otimes \sigma \stackrel{\text{def}}{=} \langle\!\langle \pi_1, \pi_2; \sigma \rangle\!\rangle_l$, and $\sigma \otimes A \stackrel{\text{def}}{=} \langle\!\langle \pi_1; \sigma, \pi_2 \rangle\!\rangle_r$ where π_i denote projections. Then $A\otimes$ and $\otimes A$ both define functors on $\mathcal{CBV}$ which CPO-enrich. We then define, for $\sigma : A \to B$ and $\tau : C \to D$: (i) $\sigma \otimes_l \tau = (\sigma \otimes C); (B \otimes \tau)$ and (ii) $\sigma \otimes_r \tau = (A \otimes \tau); (\sigma \otimes D)$.

(iii) (partial exponential [31]) The functor $\_ \otimes A$ viewed as going from $\mathcal{CBV}_{\mathtt{t}}$ to $\mathcal{CBV}$ has the right adjoint $A \Rightarrow \_ : \mathcal{CBV} \to \mathcal{CBV}_{\mathtt{t}}$, which CPO-enriches. Equivalently, there exists an arrow

$\mathsf{ev} : (A \rightleftharpoons B) \otimes A \to B$ such that, for any $\sigma : C \otimes A \to B$, there is a unique total arrow $p\lambda(\sigma) : C \to A \rightleftharpoons B$ satisfying $(p\lambda(\sigma) \otimes \mathsf{id}); \mathsf{ev} = \sigma$, and $p\lambda$ is a continuous operator on each homset.

PROOF: (ii) is from Corollary 4.2 of [50]. The CPO-enrichment comes from that of pairing in $\mathcal{CBV}_{\mathtt{t}}$. For (iii), $\mathsf{ev} : (A \rightleftharpoons B) \otimes A \to B$ first receives the initial signal at the domain, which is essentially at the initial sort of A, then asks, by the dual action, at $\overline{A}$, then starts the copy-cat between the rest of A and the rest of $\overline{A}$; and when Opponent answers at B (of the domain), then does the dual action at B of the co-domain, and starts another copy-cat between two B's again. Next, given $\sigma : C \times A \to B$, $p\lambda(\sigma) : C \to A \rightleftharpoons B$ has the following behaviour: after the initial signal at C, it answers immediately at $A \rightleftharpoons B$ by the unique Player signal. Opponent then asks, at some initial sort of A (which is the only possibility by the switching condition), which now gives $p\lambda(\sigma)$ two data, one from C and one from A. This is precisely what σ would have gotten when it receives the initial signal. Now $p\lambda(\sigma)$ can act precisely as in σ, using the $\overline{A}$ at the co-domain instead of A in the domain (note the latter is $\overline{A}$ in the whole prearena). The defining equations are easily established (we list calculation using process representation in Appendix B.11). The universality follows because a P-view from C to $A \rightleftharpoons B$ whose second action is at the co-domain (i.e. the kind which is in a total arrow) is essentially that from $C \otimes A$ to B, which in turn is because a legal action sequence from C to $A \rightleftharpoons B$ switches between A and B only at a Player's turn, just as in Proposition 2.8. $\square$

Note that (iii) above means the homset $\mathcal{CBV}(A, B)$ is represented in $\mathcal{CBV}_{\mathtt{t}}(\mathbf{1}, A \rightleftharpoons B)$, i.e. all maps in $\mathcal{CBV}(\mathbf{1}, A \rightleftharpoons B)$ except $\perp$ correspond to $\hom(A, B)$ order-isomorphically. We now give a few basic properties of partial pairings (which, from now on, we simply call *pairings*). In particular (i) shows a strongly intensional character of $\mathcal{CBV}$.

3.9. **Proposition.** (partial pairings)

(i) There exist σ_1 and σ_2 such that $\langle\!\langle \sigma_1, \sigma_2 \rangle\!\rangle_l \neq \langle\!\langle \sigma_1, \sigma_2 \rangle\!\rangle_r$.

(ii) Neither of $\otimes_l$ or $\otimes_r$ in $\mathcal{CBV}$ gives a bifunctor, i.e. for some $\tau_{1,2}, \sigma_{1,2}$ with appropriate types, we have $(\tau_1; \sigma_1) \otimes_{l,r} (\tau_2; \sigma_2) \neq (\tau_1 \otimes_{l,r} \tau_2); (\sigma_1 \otimes_{l,r} \sigma_2)$.

(iii) Right/left pairings and tensors are right/left strict w.r.t. $\perp$, respectively, and preserve and reflect totality (of both variables). Right/left pairings coincide if either is total or when it is a diagonal, i.e. if two arrows are identical, so in particular if both are from $\mathbf{1}$. Right/left tensors coincide iff either is total or both are bottoms (so again when both are from $\mathbf{1}$). We write $\langle\!\langle \sigma_1, \sigma_2 \rangle\!\rangle$, resp. $\sigma_1 \otimes \sigma_2$, when left/right pairings, resp. tensors, coincide.

(iv) $\langle\!\langle \sigma_1, \sigma_2 \rangle\!\rangle_l; \pi_1 = \sigma_1$ when σ_2 is total, similarly $\langle\!\langle \sigma_1, \sigma_2 \rangle\!\rangle_r; \pi_2 = \sigma_2$ when σ_1 is total. Also, $\tau; \langle\!\langle \sigma_1, \sigma_2 \rangle\!\rangle_{l,r} = \langle\!\langle \tau; \sigma_1, \tau; \sigma_2 \rangle\!\rangle_{l,r}$ when τ is total.

(v) $\langle\!\langle \sigma_1, \sigma_2 \rangle\!\rangle_l; \tau_1 \otimes \tau_2 = \langle\!\langle \sigma_1; \tau_1, \sigma_2; \tau_2 \rangle\!\rangle_l$ if τ_1 is total, similarly $\langle\!\langle \sigma_1, \sigma_2 \rangle\!\rangle_r; \tau_1 \otimes \tau_2 = \langle\!\langle \sigma_1; \tau_1, \sigma_2; \tau_2 \rangle\!\rangle_r$ if τ_2 is total.

(vi) $\sigma \otimes_{l,r} \tau = \langle\!\langle \pi_1; \sigma, \pi_2; \tau \rangle\!\rangle_{l,r}$.

(vii) $\langle\!\langle p\lambda(\sigma_1), \sigma_2 \rangle\!\rangle; \mathsf{ev} = \langle\!\langle \mathsf{id}_C, \sigma_2 \rangle\!\rangle; \sigma_1$ for $\sigma_1 : C \otimes A \to B$ and $\sigma_2 : C \otimes A \to B$.

Remark. It is useful to have an intuitive understanding of the behaviour of partial pairings before embarking on the proof. Take, say, $\langle\!\langle \sigma, \tau \rangle\!\rangle_l : C \to A \otimes B$ (the right pairing is just dual). After the initial signal at C, it has the same behaviour as σ assuming the latter is given the same initial signal, until σ has a P-signal at the co-domain A (if ever). At this point (note the next action is a P-action), the behaviour becomes that of τ, starting just after the same initial O-signal at C (so indeed we have a P-action), until it has a P-signal at the co-domain B (if ever). Now we have a pair of actions, one from σ (at A) and one from τ (at B), so we can combine them to have a P-signal at $A \otimes B$. In the rest, the strategy behaves just as the disjoint union of the rest of respective strategies. Thus, as was already noted by Moggi [42], $\langle\!\langle \sigma_1, \sigma_2 \rangle\!\rangle_l$ and $\langle\!\langle \sigma_1, \sigma_2 \rangle\!\rangle_r$ reflect the "order of evaluation." This shows that, while the algebraic situation of $\mathcal{CBV}$ as described in 3.7

is quite analogous to pCPO (for the detailed study of the latter and related universes see [16]), it has a strong intensional character where some algebraic structures which have only been implicit in CPO come out explicitly.

PROOF: For (i), take, for example, $\sigma : \mathsf{nat} \rightleftharpoons \mathsf{nat} \to \mathsf{nat}$ of Example 2.10 (iv), together with $\sigma' : \mathsf{nat} \rightleftharpoons \mathsf{nat} \to \mathsf{nat}$ which has the same behaviour as σ except that it gives, say, 2, instead of 3, at the domain. Then obviously $\langle\!\langle \sigma, \sigma' \rangle\!\rangle_l \neq \langle\!\langle \sigma, \sigma' \rangle\!\rangle_r$. Then (ii) follows from Corollary 4.3 of [50]. For (iii) all statements are immediate from the definition. For example, $\langle\!\langle \bot, \sigma \rangle\!\rangle_l \stackrel{\text{def}}{=} \langle\!\langle \bot^\dagger, \sigma^\dagger \rangle\!\rangle; \mathtt{st}'_{A,TB}; T(\mathtt{st}_{A,B}; \mathsf{dn})$, in which, after each initial action at the domain, there is an immediate signal at $TA \times TA$, then $\mathtt{st}'_{A,TB}$, after going through two actions in its co-domain, asks at the left-component TA of its domain, but then there is no further action because $\bot^\dagger$ has only the unique P-signal at TA. Notice in terms of the behavioural description in the above Remark, we can rephrase the reasoning simply as: after the initial O-signal, the behaviour of $\langle\!\langle \bot, \sigma \rangle\!\rangle_l$ should start from that of $\bot$, but $\bot$ never does anything, hence the whole arrow can do nothing. Similarly, the statements in (iv) are obvious, e.g. in $\langle\!\langle \sigma_1, \sigma_2 \rangle\!\rangle_l; \pi_1$, if σ_2 is total, the behaviour of $\langle\!\langle \sigma_1, \sigma_2 \rangle\!\rangle_l$ until it acts in the co-domain is that of σ_1, but once it reacts at the co-domain, π_1 interacts only at the left component by copy-cat, hence the whole behaviour is that of σ_1. For (v) if τ_1 is total $\langle\!\langle \sigma_1, \sigma_2 \rangle\!\rangle_l; \tau_1 \otimes \tau_2$ first behaves at the domain as σ_1, then as $\sigma_2; \tau_2$, until reaching the co-domain eventually, which is the same as $\langle\!\langle \sigma_1; \tau_1, \sigma_2; \tau_2 \rangle\!\rangle_l$. For (vi), we calculate, using (v) and the definition of tensor: $\sigma_1 \otimes_l \sigma_2 \stackrel{\text{def}}{=} \langle\!\langle \pi_1; \sigma_1, \pi_2 \rangle\!\rangle_l; \mathsf{id} \otimes \sigma_2 = \langle\!\langle \pi_1; \sigma_1, \pi_2; \sigma_2 \rangle\!\rangle_l$, as required. (vii) is derived by: $\langle\!\langle p\lambda(\sigma_1), \sigma_2 \rangle\!\rangle; \mathsf{ev} = \langle\!\langle \mathsf{id}_C, \sigma_2 \rangle\!\rangle; p\lambda(\sigma_1) \otimes \mathsf{id}_A; \mathsf{ev} = \langle\!\langle \mathsf{id}_C, \sigma_2 \rangle\!\rangle; \sigma_1$, where (v) above is used in the first equation. □

Finally we discuss how recursion arises in the present setting.

3.10. **Definition.**

(i) (pointed arenas) A is *pointed* when it has a unique initial sort, or, equivalently, when the homset $(\mathbf{1}, A)$ in $\mathcal{CBV}_\mathtt{t}$ is a pointed cpo. We write $\mathsf{dn}'_A : TA \to A$ for the unique total map such that $\mathsf{up}_A; \mathsf{dn}'_A = \mathsf{id}_A$.

(ii) (call-by-value fixed point combinator, cf.[16]) Let A be pointed. Then $\mathsf{fix}_A : A \rightleftharpoons A \to A$ is the strategy with the following behaviour: after the unique initial signal at the domain, it reacts by asking at the initial sort of $\overline{A}$ in the domain, to which, *if* Opponent asks at a question in $\overline{A}$, it has no subsequent behaviour: *else*, i.e. if Opponent answers at the initial sort of A in the domain, then it visits the initial sort of A in the co-domain (say by x), after which, if there is any O-action (say y) at A of the domain, it visits the initial sort of $\overline{A}$ in the domain, and, as before, only when the Opponent answers at A in the domain (say by z), copies y at A in the domain, say by $\overline{y}$, justified by z, after which, continuing the copy-cat between A in the domain for those actions justified by $\overline{y}$ and $\overline{A}$ in the co-domain for those actions justified by y; In the same way, whenever an O-action at A in the co-domain which is immediately justified by the initial action at A is done, the strategy visits the initial action of $\overline{A}$, and proceeds to the copy-cat. On the other hand, if Opponent asks at $\overline{A}$ in the domain by an action (say x') immediately justified by x, then it again visits the initial sort of $\overline{A}$ in the domain, waits for the O-action (say w) at the initial sort of A in the domain, then does the copy-cat between the actions justified by x' and those justified by w. Similarly when there is an O-action by later instances of x.

(iii) (recursion) For a pointed A and $\sigma : C \times A \to A$, we define the *recursion of* σ, denoted $\mathbf{rec}(\sigma)$, as: $\mathbf{rec}(\sigma) : C \to A \stackrel{\text{def}}{=} p\lambda(\sigma); \mathsf{fix}_A$.

Remark.

(i) A basic example of pointed arena is the higher-order arena $A \Rightarrow B$ for some A and B. It is notable that pointed arenas are precisely objects in the category of Eilenberg-Moore algebra of the monad $\mathbf{T}$ (then dn' is the arrow of that type used in it).

(ii) The behaviour of fix, the call-by-value fixed-point combinator, is among the most complex ones which use the copy-cat. The essential idea is that, whenever a new dialogue is opened by an O-action at the second-level sort of the positive types ($\overline{A}$ in the domain and A in the co-domain), then two "view-clearing" moves are done at the initial sorts of $\overline{A}$ and A in the domain, after which the copy-cat starts by copying the O-action to the P-action immediately justified by the second view-clearing move. This form is essential not only for the legality of actions but also for realising the structure of nested feedback (which is done from $\overline{A}$ in the domain to A in the domain) in recursion, as described in the the proof of the next Proposition. Note fix is innocent because two views for its two copy-cat behaviours are independent (by the same reasoning as for identity strategy, cf. Example 2.10(v)). We observe that the existence of the fixed point combinator can also be derived from the recursive types in $\mathcal{CBV}$, cf.[18], or from the recursion in the underlying total category.

The basic properties of recursion follow.

3.11. **Proposition.** For each homset of form $hom(C \otimes A, A)$ with A pointed, $\mathbf{rec}(\cdot)$ is a continuous operator. Moreover, letting A be pointed and $\sigma : C \otimes A \to A$, we have:

(i) $\mathbf{rec}(\tau \otimes \mathsf{id}_A; \sigma) = \tau; \mathbf{rec}(\sigma)$ for each $\tau : B \to C$.

(ii) $\tau; \mathbf{rec}(\sigma) = \tau; \langle\!\langle \mathsf{id}_C, \mathbf{rec}(\sigma) \rangle\!\rangle; \sigma$ for each $\tau : \mathbf{1} \to C$. Moreover $\mathbf{rec}(\sigma) = \langle\!\langle \mathsf{id}_C, \mathbf{rec}(\sigma) \rangle\!\rangle; \sigma$ when σ is total.

(iii) Given $\tau : \mathbf{1} \to C$, define $\rho_i : \mathbf{1} \to A$, $i \in \omega$, as: (1) $\rho_0 = \bot$, (2) $\rho_{i+1} = \langle\!\langle \tau, \rho_i^{\dagger}; \mathsf{dn}' \rangle\!\rangle; \sigma$. Then $\{\rho_i\}$ is an increasing ω-chain such that $\sqcup \rho_i = \tau; \mathbf{rec}(\sigma)$.

PROOF: See Appendix A.15. □

We give the diagrammatic representation of $\mathbf{rec}(\sigma)$, as well as the first two equations, in Figures 4–6.

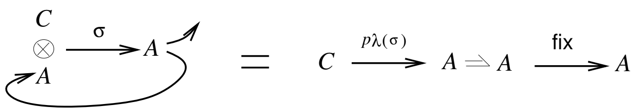

Figure 4

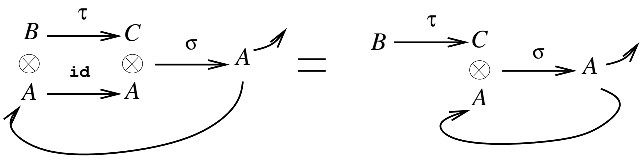

Figure 5

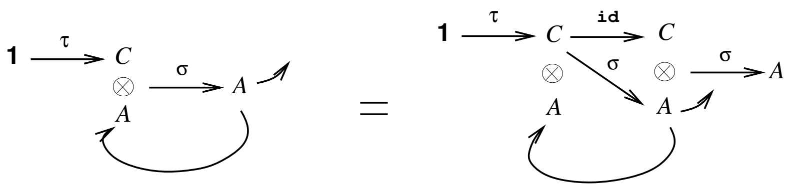

Figure 6

The next section studies the extensional universe derived from $\mathcal{CBV}$.

4. Extensional Universe

$\mathcal{CBV}$ represents an abstract notion of execution of call-by-value computation. For the interpretation of programming languages at the same abstraction level as in the standard semantic universe like the category of domains, we may need a more abstract universe, which we construct from $\mathcal{CBV}$ by a simple quotient construction. The universe is also useful for understanding the behaviour of arrows in $\mathcal{CBV}$ in an abstract way. We first define the following ordering, cf. [48, 16].

4.1. **Definition.** (standard ordering) Define $\precapprox$ on each homset $\mathrm{hom}(A, B)$ by:

$$\sigma_1 \precapprox \sigma_2 \quad \stackrel{\mathrm{def}}{\Leftrightarrow} \quad \forall C, C', \tau : C \to A, \tau' : B \to C'. \ \tau;\sigma_1;\tau' \Downarrow \quad \Rightarrow \quad \tau;\sigma_2;\tau' \Downarrow .$$

4.2. **Proposition.**

(i) $\precapprox$ is a preorder such that: (1) $\precapprox \supset \sqsubseteq$ at each homset, and (2) composition ; is monotone with respect to $\precapprox$.

(ii) If $\sigma : A \to B$ is insensitive, $\sigma \precapprox \sigma'$ for all $\sigma' : A \to B$. If $\sigma : A \to B$ is total, $\sigma \precapprox \sigma'$ implies σ' is total. Finally, letting $B \neq \mathbf{0}$, $\sigma : A \to B \not\precapprox \bot$ iff $\exists \tau : \mathbf{1} \to A.\tau;\sigma \Downarrow$.

(iii) $\sigma_1 \precapprox \sigma_2$ iff for any $\tau : \mathbf{1} \to A$ and $\tau' : B \to \mathbf{1}$, we have $\tau;\sigma_1;\tau' \Downarrow \quad \Rightarrow \quad \tau;\sigma_2\tau' \Downarrow$.

Proof: (i) is immediate. For (ii), the first statement, if σ_1 is insensitive then so is $\tau;\sigma_1;\tau'$ but insensitive arrows are never total (except on $\mathbf{0}$ in which case the statement is vacuous). The second statement is immediate by taking id on both sides of Definition 4.1. For the last statement, the "if" direction is immediate from the definition and $\tau;\bot = \bot$ always. For the "only if" direction, suppose $\sigma \Downarrow$. If $\tau;\sigma \not\Downarrow$ for $\tau : \mathbf{1} \to A$, immediately $\tau;\sigma = \bot$, that is $\tau = \bot$, as required. For (iii) let the right hand side hold and $\tau;\sigma_1;\tau' \Downarrow$ for $\tau : C \to A$ and $\tau' : B \to C$. Take $\{\sigma_i \mid \mathbf{1} \to C\}$ where i ranges over initial actions of C, and where σ_i just does (after the unique O-signal at $\mathbf{1}$) the initial P-action i at C and then ceases to react. Clearly $\sigma_i;\tau;\sigma_1;\tau';!_{C'} \Downarrow$ for each initial action i, so by assumption $\sigma_i;\tau;\sigma_2;\tau';!_{C'} \Downarrow$ for each i, which immediately shows (via the behavioural characterisation of totality in Proposition 3.2) $\tau;\sigma_2;\tau' \Downarrow$, as required. □

Remark. For (ii), the converse of the first statement does *not* hold, so in particular the bottom map in $\widehat{\mathcal{CBV}}$ (see 4.3 below) may include non-insensitive strategies. For explication of the situation, see Appendix A.17. We also note the property (iii) is a general phenomenon, see 3.3.2 of [16].

4.3. **Definition and Proposition.** $\widehat{\mathcal{CBV}}$ is given by the following data:

- Objects: those of $\mathcal{CBV}$.
- Arrows: $\precapprox$-equivalence classes of arrows in $\mathcal{CBV}$, denoted $f, g, \ldots$, which are ordered by the induced partial order which we also denote $\precapprox$.

Then $\widehat{\mathcal{CBV}}$ is enriched over Poset (the category of posets with monotone maps) with each homset having a bottom (denoted $\bot$ again) with strict composition at both sides.

4.4. **Proposition.**

(i) $\mathbf{0}$ is the zero object in $\widehat{\mathcal{CBV}}$, that is, is both terminal and initial. Also, $f = \bot : A \to B$ iff $f : A \to \mathbf{0} \to B$ iff $\exists \rho \in f.\rho : A \to \mathbf{0} \to B$ iff $\exists \rho \in f.s \in \rho : A \ \Rightarrow \ s \restriction B = \varepsilon$.

(ii) We say $f : A \to B$ is *total*, written $f \Downarrow$, when $g;f = \bot$ implies $g = \bot$, equivalently when the square $\langle \mathbf{0} \to A \xrightarrow{f} B,\ \mathbf{0} \to \mathbf{0} \to B \rangle$ is a pullback. Then f is total iff $g;f = \bot$ for $g : \mathbf{1} \to A$ implies $g = \bot$, iff $\forall \sigma \in f.\sigma \Downarrow$, iff $\exists \sigma \in f.\sigma \Downarrow$.

(iii) $f_1 \precapprox f_2 : A \to B$ iff $\forall x : \mathbf{1} \to A.\ x;f_1 \precapprox x;f_2$.

(iv) $f \neq \bot : A \to B$ iff for some $g : \mathbf{1} \to A$ we have $g;f \Downarrow$.

(v) If $\{f_i\}$ is a ω-chain in $\mathcal{CBV}_{\mathsf{t}}$ and $\{\rho_i\}$ is also such in $\mathcal{CBV}$ with $\rho_i \in f_i$ for each i, $\sqcup_i\{f_i\}$ exists and is given as $[\sqcup_i \rho_i]_{\precapprox}$.

Proof: **(i)** $\mathbf{0}$ is initial because it is so in $\mathcal{CBV}$. Before showing $\mathbf{1}$ is terminal, we observe that the second part is immediate from Proposition 4.2 (ii), which says insensitive arrows are extensionally the least elements (notice the last two statements are just different characterisations of insensitive arrows in $\mathcal{CBV}$, cf.3.3). But all arrows to $\mathbf{0}$ in $\mathcal{CBV}$ are by definition insensitive, thus we now know $\mathbf{0}$ is terminal.

(ii) The equivalence of definitions is by (i) above. That $f \Downarrow$ implies the first statement is by definition. Suppose the first statement holds, i.e. $g; f = \bot$ for $g : \mathbf{1} \to A$ implies $g = \bot$, equivalently for each $\sigma \in f$, $\tau; \sigma \underset{\approx}{\precsim} \bot$ for $\tau : \mathbf{1} \to A$ implies $\tau \underset{\approx}{\precsim} \bot$. But $\tau : \mathbf{1} \to A \underset{\approx}{\precsim} \bot$ implies $\tau = \bot$, we know $\sigma \Downarrow$. Next suppose $\sigma \in f$ implies $\sigma \Downarrow$. Suppose $\tau; \sigma \underset{\approx}{\precsim} \bot$ for $\tau : C \to A$. We show $\tau \underset{\approx}{\precsim} \bot$. If $A = \mathbf{0}$ the statement is vacuous so suppose not. Now if $\tau \underset{\approx}{\not\precsim} \bot$, then for some $\tau' : \mathbf{1} \to C$ we have $\tau'; \tau \Downarrow$ by Proposition 4.2 (ii), the last statement. But then we should have $\tau'; \tau; \sigma \Downarrow$, which cannot be the case, hence we now know $\tau \underset{\approx}{\precsim} \bot$, as required. The last two statements are equivalent by Proposition 4.2 (ii), the second statement.

(iii) The "only if" direction holds by monotonicity of composition. For "if" direction, $\forall x.\, \mathbf{1} \to A.\ x; f_1 \underset{\approx}{\precsim} x; f_2$ implies $\forall x : \mathbf{1} \to A,\ y : B \to \mathbf{1}.\ \ x; f_1; y \Downarrow \ \Rightarrow\ x; f_2; y \Downarrow$, as required.

(iv) The contrapositive is $f = \bot$ iff for all $g : \mathbf{1} \to A$ we have $g; f \Downarrow$, but this is immediate from (iii) above, rephrasing $f = \bot$ by $f \underset{\approx}{\precsim} \bot$.

(v) Clearly $\forall i. f_i \ \underset{\approx}{\precsim}\ [\sqcup_i \rho_i]_{\underset{\approx}{\precsim}}$. If $\forall i.\, g \underset{\approx}{\succsim} f_i$, then $\tau \in g$ and $\forall i.\, \tau \underset{\approx}{\succsim} \rho_i$, thus $\tau \underset{\approx}{\succsim} \sqcup_i \rho_i$, so that $g \underset{\approx}{\succsim} [\sqcup_i \rho_i]_{\underset{\approx}{\precsim}}$. □

By (ii), all properties of total maps in $\mathcal{CBV}$ carry over to their corresponding arrows in $\widehat{\mathcal{CBV}}$. (iii) and (iv) show the extensional nature of arrows in $\widehat{\mathcal{CBV}}$. We write $\widehat{\mathcal{CBV}}_{\mathtt{t}}$ for the subcategory of total maps in the extensional universe. The following gives what corresponds to Proposition 3.7 for the present setting.

4.5. **Proposition.**

(i) $\widehat{\mathcal{CBV}}_{\mathtt{t}}$ is well-pointed, with the initial object, and has finite products which give a bifunctor which Poset-enriches, all inheriting from $\mathcal{CBV}_{\mathtt{t}}$.

(ii) The inclusion functor from $\widehat{\mathcal{CBV}}_{\mathtt{t}}$ to $\widehat{\mathcal{CBV}}$ has the right adjoint inheriting all constructions from T (which we write again T), which Poset-enriches. The corresponding monad, again denoted $\mathbf{T}$, has strengths and is commutative, that is $\psi_{A,B} = \tilde{\psi}_{A,B}$ where $\psi_{A,B}$ and $\tilde{\psi}_{A,B}$ are morphisms given in 3.8 (i). Again the Kleisli category of $\mathbf{T}$ on $\widehat{\mathcal{CBV}}_{\mathtt{t}}$ is isomorphic to $\widehat{\mathcal{CBV}}$.

Proof: **(i)** Well-pointedness is the contrapositive of Proposition 4.4 (iii). $\mathbf{0}$ is initial, $\mathbf{1}$ is terminal, and products come from $\mathcal{CBV}$. We show the paring is monotone. Given $\sigma_1, \sigma_1' : C \to A$ and $\sigma_2 : C \to B$, suppose $\sigma_1 \underset{\approx}{\precsim} \sigma_1'$. By Proposition 4.2, we set $C = \mathbf{1}$. Fix $\tau : A \times B \to \mathbf{1}$ and define $\overline{\sigma_2} : A \to A \times B$ as $\mathsf{id}_A \times \sigma_2$. Immediately $\sigma; \overline{\sigma_2}; \tau = \langle \sigma, \sigma_2 \rangle; \tau$ for each $\sigma : \mathbf{1} \to A$. Thus $\langle \sigma_1, \sigma_2 \rangle; \tau \Downarrow$ implies $\langle \sigma_1', \sigma_2 \rangle; \tau \Downarrow$ (by the monotonicity of ;), as required. Notice that, by a trivial reasoning, the pairing is also order-reflecting.

(ii) For the adjointness, let $\eta_A : A \to TA$ be given as before, and let $f : A \to TB$ be total. Then $\rho \in f$ is total and $\rho \underset{\approx}{\precsim} \rho_i' : A \to TB$ implies $\rho_\dagger = \rho; \mathsf{dn} \underset{\approx}{\precsim} \rho'; \mathsf{dn} = \rho_\dagger'$. Write $f_\dagger$ for $[\rho_\dagger \mid \rho \in f]_{\underset{\approx}{\precsim}}$ which is now well-defined. By definition $\eta; T(f_\dagger) = f$. Suppose $\eta; T(g) = f$ for $g : A \to B$, i.e. $\eta; T(\tau) \cong \rho \in f$ for some $\tau \in g$. By putting $\epsilon\,(= \mathsf{dn}) : TB \to B$, we have $\eta; T(\tau); \epsilon = \tau \cong \rho_\dagger$, as required. For the rest, that $\mathtt{st}$ and $\mathtt{st}'$ again give the strength come from $\mathcal{CBV}$. The commutativity, that is $\psi_{A,B} = \tilde{\psi}_{A,B}$ (both regarded as arrows in $\widehat{\mathcal{CBV}}$ and which again become singletons), holds because $\psi_{A,B}$ and $\tilde{\psi}_{A,B}$ only differ in the order of visiting the initial two sorts of A-component, OA^{TA} and PA^{A}, and those of B-component, OQ^{TB} and PA^{B}, at $TA \times TB$. Because they both do this consecutively and exactly once and, moreover, each sequence actions own independent views both for Player and Opponent, so that no visible difference can come about at the codomain $T(A \times B)$ even if the order is reversed. This is easily established by the case analysis of the first few actions of $\tau : \mathbf{1} \to TA \otimes TB$, tracing interactions afterwards a few steps further: after that two strategies converge into exactly the same behaviour. $\square$

We can now state the basic facts about type structures of $\widehat{\mathcal{CBV}}$. Below we write $\cong$ for $\precsim \cap \succsim$.

4.6. **Proposition.** $\widehat{\mathcal{CBV}}$ is a symmetric monoidal category with the tensor given by $\otimes$ enriched over Poset, with pairings given by $\langle\!\langle f, g \rangle\!\rangle = \{\langle\!\langle \sigma, \tau \rangle\!\rangle_l \cong \langle\!\langle \sigma, \tau \rangle\!\rangle_r \mid \sigma \in f, \tau \in g\}$, which satisfy $f_1 \otimes f_2 = \langle\!\langle \pi_1; f_1, \pi_2; f_2 \rangle\!\rangle$ and $f; \langle\!\langle g_1, g_2 \rangle\!\rangle = \langle\!\langle f; g_1, f; g_2 \rangle\!\rangle$, as well as $\langle\!\langle f, f \rangle\!\rangle; \pi_i = f$. Then $\_ \otimes A$, viewed as going from $\widehat{\mathcal{CBV}}$ to $\widehat{\mathcal{CBV}}_{\mathtt{t}}$, has the right adjoint $A \Rightarrow \cdot$, with the same counit ev in $\mathcal{CBV}$, which again Poset-enriches.

Proof: By Corollary 4.3 in [50] and Proposition 4.5, immediately $\widehat{\mathcal{CBV}}$ is symmetric monoidal, from which all properties of pairings also follow. For the final clause let $\sigma_1 \precsim \sigma_2 : C \times A \to B$, and assume $\tau; p\lambda(\sigma_1); \tau' \Downarrow$ for $\tau : \mathbf{1} \to C$ and $\tau' : (A \Rightarrow B) \to \mathbf{1}$. Then there is a decomposition of τ' to its two components, say $\tau'_A : \mathbf{1} \to A$ and $\tau'_B : B \to \mathbf{1}$, such that $\tau; p\lambda(\sigma_1); \tau' = \langle\!\langle \tau, \tau'_A \rangle\!\rangle; \sigma_1; \tau_B$, because the views of τ' are either entirely in $\overline{A}$ or in B. Thus $\langle\!\langle \tau, \tau'_A \rangle\!\rangle; \sigma_2; \tau'_B = \tau; p\lambda(\sigma_2); \tau' \Downarrow$, as required. $\square$

We note that the functor $\Rightarrow$ easily extends to a Poset-enriched bifunctor of type $\widehat{\mathcal{CBV}}^{op} \times \widehat{\mathcal{CBV}} \to \widehat{\mathcal{CBV}}$. Finally we check that the recursion operator is well defined, using the finite approximation result in $\mathcal{CBV}$.

4.7. **Proposition.** Let A be pointed and $f : C \otimes A \to A$. Define $\mathbf{rec}(f) \stackrel{\text{def}}{=} [\mathbf{rec}(\sigma)]_{\precsim}$ for any $\sigma \in f$. Then $\mathbf{rec}(\cdot)$ is a well-defined monotone operator of type $\hom(C \otimes A, A) \to \hom(A, A)$ (regardless of the choice of σ). Moreover we have:

- (i) $\mathbf{rec}(g \otimes \mathsf{id}_A; f) = g; \mathbf{rec}(f)$ for any $\tau : B \to C$.
- (ii) $\mathbf{rec}(f) = \langle\!\langle \mathsf{id}_A, \mathbf{rec}(f) \rangle\!\rangle; f$ always.
- (iii) Let $g : \mathbf{1} \to C$ and define $e_i : C \to A$, $i \in \omega$ as: (1) $e_0 = \bot$. (2) $e_{i+1} = \langle\!\langle g, e_i^{\dagger}; \mathsf{dn}' \rangle\!\rangle; f$. Then $\{e_i\}$ is an increasing ω chain for which $\sqcup\{e_i\}$ exists given as $g; \mathbf{rec}(f)$.

Proof: We first show $\sigma \precsim \sigma' : C \otimes A \to A$ implies $\mathbf{rec}(\sigma_1) \precsim \mathbf{rec}(\sigma_2)$, from which well-definedness is immediate. Indeed, suppose $\sigma \precsim \sigma' : C \times A \to A$. Construct $\{\rho_i\}$ (approximating $\mathbf{rec}(\sigma)$) and $\{\rho'_i\}$ (approximating $\mathbf{rec}(\sigma')$) following Proposition 3.11 (iii). By the monotonicity of $\langle\!\langle \cdot, \cdot \rangle\!\rangle$ and ; w.r.t $\precsim$, we immediately know $\rho_i \precsim \rho'_i$ for each i. Since the behaviours of arrows entirely depend on their approximations, we readily conclude $\tau; \mathbf{rec}(\sigma) \precsim \tau; \mathbf{rec}(\sigma')$, but by Proposition 4.4 (iii) this implies $\mathbf{rec}(\sigma) \precsim \mathbf{rec}(\sigma')$, so we now know $\mathbf{rec}(\cdot)$ is well-defined and monotone in $\widehat{\mathcal{CBV}}$. We now show (i)–(iii). (i) is immediate from Proposition 3.11 (i). For (ii), again from Proposition 3.11 (ii), we know $g; \mathbf{rec}(f) = g; \langle\!\langle \mathsf{id}_C, \mathbf{rec}(f) \rangle\!\rangle; f$ for each g. Again by Proposition 4.4 (iii) we know $\mathbf{rec}(f) = \langle\!\langle \mathsf{id}_C, \mathbf{rec}(f) \rangle\!\rangle; f$, as required. For (iii), given $f : C \otimes A \to A$ and $g : \mathbf{1} \to C$ take $\sigma \in f$ and $\tau \in g$. By definition each e_i of $\{e_i\}$ above (which is obviously an ω-chain) is associated with $\{\rho_i\}$ which is constructed following 3.11 (iii), so that, by 4.4 (v), we know its limit is $[\tau; \mathbf{rec}(\sigma)]_{\precsim}$, that is $g; \mathbf{rec}(f)$, as required. $\square$

Further $\widehat{\mathcal{CBV}}$ has arbitrary products and co-products and allows the treatment of recursive types (see [18]), but the constructions so far are all we need for our current purpose. In the next section we use these type structures to interpret call-by-value PCF.

5. Interpretation of $\mathrm{PCF_V}$

$\mathrm{PCF_V}$ [47] is a typed programming language based on call-by-value evaluation. The syntax and evaluation rules can be found in the standard literature, cf.[23, 55, 53], which are briefly reviewed in Appendix C (basically following [23] except the recursion is only defined for function types, cf.[55, 53]). $\mathcal{CBV}$ and its extensional quotient are conceived to represent call-by-value, or partial, higher-order functional computation. Moreover it has a type structure which does include that of

$\mathrm{PCF_V}$. Thus we may seek to represent $\mathrm{PCF_V}$-terms and its computation in these universes. We primarily consider the interpretation in $\mathcal{CBV}$, and only at the last step move to $\widehat{\mathcal{CBV}}$. We start from the interpretation of types.

5.1. **Types.** Types in $\mathrm{PCF_V}$ are mapped to objects in $\mathcal{CBV}$ as $[\![\iota]\!] \stackrel{\text{def}}{=} \mathsf{nat}$, $[\![o]\!] \stackrel{\text{def}}{=} \mathsf{bool}$, and $[\![\alpha \Rightarrow \beta]\!] \stackrel{\text{def}}{=} [\![\alpha]\!] \Rightarrow [\![\beta]\!]$. We also define the mapping on environments as $[\![\varepsilon]\!] \stackrel{\text{def}}{=} \mathbf{1}$ and $[\![\Gamma, x : \alpha]\!] \stackrel{\text{def}}{=} [\![\Gamma]\!] \otimes [\![\alpha]\!]$.

We next give the interpretation of terms.

5.2. **Definition.** Given a $\mathrm{PCF_V}$-term $\Gamma \rhd M : \alpha$, we define $[\![\Gamma \rhd M : \alpha]\!]$ as an arrow in $\mathcal{CBV}$ of type $[\![\Gamma]\!] \to [\![\alpha]\!]$ inductively as follows, assuming either left or right pairings and tensors are selected uniformly.

(i) $[\![\Gamma, x : \alpha \rhd x : \alpha]\!] \stackrel{\text{def}}{=} \pi : [\![\Gamma]\!] \otimes [\![\alpha]\!]$, where π is an appropriate projection.

(ii) $[\![\Gamma \rhd \lambda x^\alpha.M : \alpha \Rightarrow \beta]\!] \stackrel{\text{def}}{=} p\lambda(\sigma) : [\![\Gamma]\!] \to [\![\beta]\!]$, where $[\![\Gamma, x : \alpha \rhd M : \beta]\!] = \sigma : [\![\Gamma]\!] \otimes [\![\alpha]\!] \to [\![\beta]\!]$.

(iii) $[\![\Gamma \rhd MN : \beta]\!] \stackrel{\text{def}}{=} \langle\!\langle \sigma_1,\, \sigma_2 \rangle\!\rangle; \mathsf{ev} : [\![\Gamma]\!] \to [\![\beta]\!]$, where $[\![\Gamma \rhd M : \alpha \Rightarrow \beta]\!] = \sigma_1 : [\![\Gamma]\!] \to ([\![\alpha]\!] \Rightarrow [\![\beta]\!])$ and $[\![\Gamma \rhd N : \alpha]\!] = \sigma_2 : [\![\Gamma]\!] \to [\![\alpha]\!]$.

(iv) $[\![\Gamma \rhd \boldsymbol{\mu} x^\alpha.M : \alpha]\!] \stackrel{\text{def}}{=} \mathbf{rec}(\sigma) : [\![\Gamma]\!] \to [\![\alpha]\!]$, where $[\![\Gamma, x : \alpha \rhd M : \alpha]\!] = \sigma : [\![\Gamma]\!] \otimes [\![\alpha]\!] \to [\![\alpha]\!]$ (notice any function type is pointed).

(v) $[\![\Gamma \rhd \mathsf{cond}\ L\ M_1\ M_2 : \alpha]\!] \stackrel{\text{def}}{=} (\langle\!\langle \tau,\, \langle\!\langle \sigma_1^\dagger,\, \sigma_2^\dagger \rangle\!\rangle \rangle\!\rangle; \gamma_{T([\![\alpha]\!])})_\dagger : [\![\Gamma]\!] \to [\![\alpha]\!]$ where $[\![\Gamma \rhd L : o]\!] = \tau : [\![\Gamma]\!] \to \mathsf{bool}$, $[\![\Gamma \rhd M_1 : \alpha]\!] = \sigma_1 : [\![\Gamma]\!] \to [\![\alpha]\!]$, $[\![\Gamma \rhd M_2 : \alpha]\!] = \sigma_2 : [\![\Gamma]\!] \to [\![\alpha]\!]$, and $\gamma_A : \mathsf{bool} \otimes A \otimes A \to A$ is given in Appendix B.12.

(vi) For a constant c of type α, we set: $[\![\Gamma \rhd c : \alpha]\!] \stackrel{\text{def}}{=} !_{[\![\Gamma]\!]}; \overline{c} : [\![\Gamma]\!] \to \mathbf{1} \to [\![\alpha]\!]$ where $\overline{c} : \mathbf{1} \to [\![\alpha]\!]$ is given as a strategy with appropriate behaviour (see Appendix B.12 for concrete constructions).

The first key observations on the interpretation follow. Below V denotes a value in $\mathrm{PCF_V}$, i.e. either a constant which is not Ω, or an abstraction.

5.3. **Proposition.**

(i) (totality) $[\![\Gamma \rhd V : \alpha]\!]$ is always total.

(ii) (substitution lemma) $[\![\Gamma \rhd M\{V/x\} : \beta]\!] = \langle\!\langle \mathsf{id}_{[\![\Gamma]\!]},\, \tau \rangle\!\rangle; \sigma : [\![\Gamma]\!] \to [\![\beta]\!]$ for any $\tau = [\![\Gamma \rhd V : \alpha]\!]$ and $\sigma = [\![\Gamma, x : \alpha \rhd M : \beta]\!]$.

Proof: (i) is immediate from the definition. (ii) is by induction on the structure of M, throughout using the totality of τ which is from (i) above. The base cases and the conditional are easy. For other cases:

Case $M = M_1 M_2$. Let $[\![\Gamma, x : \alpha \rhd M_1 : \alpha \Rightarrow \beta]\!] = \sigma_1$ and $[\![\Gamma, x : \alpha \rhd M_2 : \alpha]\!] = \sigma_2$. By induction we know $\langle\!\langle \mathsf{id},\, \tau \rangle\!\rangle; \sigma_1$ and $\langle\!\langle \mathsf{id},\, \tau \rangle\!\rangle; \sigma_2$ give the interpretation of $M_1\{V/x\}$ and $M_2\{V/x\}$. Using $\langle\!\langle \mathsf{id},\, \tau \rangle\!\rangle$ is total, we calculate: $[\![\Gamma \rhd M_1\{V/x\} M_2\{V/x\}]\!] = \langle\!\langle \langle\!\langle \mathsf{id},\, \tau \rangle\!\rangle; \sigma_1,\, \langle\!\langle \mathsf{id},\, \tau \rangle\!\rangle; \sigma_2 \rangle\!\rangle; \mathsf{ev} = \langle\!\langle \mathsf{id},\, \tau \rangle\!\rangle; \langle\!\langle \sigma_1,\, \sigma_2 \rangle\!\rangle; \mathsf{ev}$, as required.

Case $M = \lambda y^\alpha.M'$. Let $\sigma_0 = [\![\Gamma, y : \alpha, x : \beta \rhd M' : \gamma]\!]$ and $\sigma_0' = [\![\Gamma, x : \beta, y : \alpha \rhd M' : \gamma]\!]$. Then $\sigma_0' = \langle\!\langle \langle\!\langle \pi_1; \pi_1',\, \pi_2 \rangle\!\rangle,\, \pi_1; \pi_2' \rangle\!\rangle; \sigma_0$. By assumption $[\![\Gamma, x : \beta \rhd M\{V/y\} : \gamma]\!] = \langle\!\langle \mathsf{id},\, \tau' \rangle\!\rangle; \sigma_0'$ where $\tau' = \pi_1'; \tau$. We calculate this, using the totality of id, τ' and π_1:

$$
\begin{array}{rcl}
\langle\!\langle \mathsf{id},\, \tau' \rangle\!\rangle; \langle\!\langle \langle\!\langle \pi_1; \pi_1',\, \pi_2 \rangle\!\rangle,\, \pi_1; \pi_2' \rangle\!\rangle; \sigma_0 & = & \langle\!\langle \langle\!\langle \mathsf{id},\, \tau' \rangle\!\rangle; \langle\!\langle \pi_1; \pi_1',\, \pi_2 \rangle\!\rangle,\, \langle\!\langle \mathsf{id},\, \tau' \rangle\!\rangle; \pi_1'; \pi_2' \rangle\!\rangle; \sigma_0 \\
& = & \langle\!\langle \langle\!\langle \pi_1',\, \pi_1'; \tau \rangle\!\rangle,\, \pi_2' \rangle\!\rangle; \sigma_0 \\
& = & \langle\!\langle \pi_1'; \langle\!\langle \mathsf{id}_\Gamma,\, \tau \rangle\!\rangle,\, \pi_2' \rangle\!\rangle; \sigma_0 \\
& = & \langle\!\langle \mathsf{id}_\Gamma,\, \tau \rangle\!\rangle \otimes \mathsf{id}_\beta \ ; \ \sigma_0
\end{array}
$$

Observing $p\lambda(\rho \otimes \mathsf{id}; \sigma) = \rho; p\lambda(\sigma)$ when ρ is total, we now get $p\lambda(\langle\!\langle \mathsf{id}_\Gamma,\, \tau \rangle\!\rangle \otimes \mathsf{id}_\beta; \sigma_0) = \langle\!\langle \mathsf{id}_\Gamma,\, \tau \rangle\!\rangle; p\lambda(\sigma_0)$, as required.

Case $M = \boldsymbol{\mu} x^{\beta}.M_0$. Let σ_0 be as above, with $\gamma = \beta$. With the same calculation we obtain $\langle\!\langle \mathsf{id}_\Gamma, \tau \rangle\!\rangle \otimes \mathsf{id}_\beta \ ; \ \sigma_0$, to which we apply Proposition 3.11 (ii) to get: $\mathbf{rec}(\langle\!\langle \mathsf{id}, \tau \rangle\!\rangle \otimes \mathsf{id} \ ; \ \sigma_0) = \langle\!\langle \mathsf{id}, \tau \rangle\!\rangle ; \mathbf{rec}(\sigma_0) = \langle\!\langle \mathsf{id}, \tau \rangle\!\rangle ; \sigma$, thus concluding the proof. ☐

We observe the following results before proceeding.

5.4. **Lemma.** If $\Gamma \rhd M\{\boldsymbol{\mu} x^{\alpha}.N/y\} \Downarrow V$ and we do not have $\boldsymbol{\mu} x.N \Downarrow U$ in its proof, then $\Gamma, y : \alpha \rhd M \Downarrow V_0$ with $V_0\{\boldsymbol{\mu} x^{\alpha}.N/y\} = V$ and its proof is no bigger than that for $M\{\boldsymbol{\mu} x^{\alpha}.N/y\} \Downarrow V$.

Proof: Write σ for the substitution. Then when $V\sigma \Downarrow V\sigma$ the result is obvious. When $(MN)\sigma \Downarrow U$, then $M\sigma \Downarrow \lambda x.M'$, $N\sigma \Downarrow V$, and $M'\{V/x\} \Downarrow U$, and their proofs do not contain $\boldsymbol{\mu} x.N \Downarrow U$, thus by induction we know $M \Downarrow \lambda x.M_0$, $N \Downarrow V_0$, and (noticing $M'\{V/x\} = (M_0\{V_0/x\})\sigma$) we have $M'\{V_0/x\} \Downarrow U_0$ with appropriate M_0, V_0, U_0 and with no bigger proofs. Other cases are immediate. ☐

We can now prove the basic adequacy results on the interpretation.

5.5. **Proposition.** $\Gamma \rhd M \Downarrow V$ implies $[\![M]\!] = [\![V]\!]$.

Proof: We use the induction on the size of the proof of $M \Downarrow V$, using the rules in C.3. For rules involving constants we can easily reason using B.12. For β_v-rule, we use 5.3 (ii) and Proposition 3.9 (vii). For recursion, we should show if $\Gamma \rhd M\{\boldsymbol{\mu} x.M/x\} : \alpha \Downarrow V$ and $[\![\Gamma \rhd M\{\boldsymbol{\mu} x.M/x\} : \alpha]\!] = [\![V]\!]$, we have $[\![\Gamma \rhd \boldsymbol{\mu} x.M : \alpha]\!] = [\![V]\!]$. By Lemma 5.4, we know $M \Downarrow V_0$ with a proof length less than $\boldsymbol{\mu} x.M \Downarrow V$. This shows in particular $[\![\Gamma, x : \alpha \rhd M : \alpha]\!] \stackrel{\text{def}}{=} \sigma$ is total, which immediately implies $\mathbf{rec}(\sigma)$ is total. Therefore:

$$\begin{array}{rcll} [\![\Gamma \rhd \boldsymbol{\mu} x.M : \alpha]\!] & = & \mathbf{rec}(\sigma) & \\ & = & \langle\!\langle \mathsf{id}, \mathbf{rec}(\sigma) \rangle\!\rangle ; \sigma & \text{(Proposition 3.11 (i))} \\ & = & [\![\Gamma \rhd M\{\boldsymbol{\mu} x.M/x\} : \alpha]\!] & \text{(Proposition 5.3 (ii))} \\ & = & [\![V]\!], & \end{array}$$

as required. ☐

5.6. **Proposition.** (computational adequacy) If $M : \alpha$ is a closed term, $[\![M]\!] \neq \bot : \mathbf{1} \rightarrow [\![\alpha]\!]$ if and only if $M \Downarrow V$ for some value V.

Proof: By the standard reasoning based on the ω-chain complete relation between terms and elements, where we use Proposition 3.11 (iii) to reason about recursion. ☐

Remark. In spite of the above proposition, the interpretation of $\mathrm{PCF_V}$-terms in $\mathcal{CBV}$ does not allow many "obvious" equations. As an example, $\lambda z.((\lambda y.My)N)$ and $\lambda z.(MN)$ are in general given distinct interpretation in $\mathcal{CBV}$ (when we use the left pairing), even though they clearly have the same extensional behaviour.

5.7. **Corollary.** (adequacy) For closed terms M and N, $[\![M]\!] \precapprox [\![N]\!]$ implies $M \preceq_{obs} N$.

Proof: Below we write $M \Downarrow$ for $\exists V.\, M \Downarrow V$. We first note if $M : \alpha$ is closed and $\neg M \Downarrow$, then for any $x : \alpha \rhd L : \beta$, $L\{M/x\} \Downarrow \ \Rightarrow \ \forall N : \alpha.\, L\{N/x\} \Downarrow$, by easy structural induction. So suppose $M \Downarrow$. Then we have, assuming $L\{M/x\} \Downarrow n$ for some n, $[\![L\{M/x\}]\!] = [\![M]\!] ; [\![L]\!] \precapprox [\![N]\!] ; [\![L]\!] = \overline{n}$, that is, $L\{N/x\} \Downarrow n$, as required. ☐

Given the adequacy result, the remaining task is to show that this interpretation, when considered modulo $\precapprox$, exactly captures the observational congruence $\preceq_{obs}$ of $\mathrm{PCF_V}$. The main tool we utilise is a subset of $\mathrm{PCF_V}$-terms called *finite canonical forms*. Finite canonical forms faithfully capture the behaviour of compact strategies of PCF-types in $\mathcal{CBV}$. We use a specific syntax for them, even though we later show that the new syntax is just an abbreviation for the old one. In the following we assume we only have one program type, ι. The incorporation of o type is straightforward.

5.8. **Definition.** (finite canonical forms) We define *finite canonical forms (FCFs for short) of type* α, by the following syntax.

(i) $\Gamma \rhd \Omega : \alpha$ and $\Gamma \rhd n : \iota$ are FCF's.

(ii) $\Gamma \rhd \lambda y^{\alpha}.M : \alpha \Rightarrow \beta$ is a FCF if $\Gamma, y : \alpha \rhd M : \beta$ is.

(iii) $\Gamma \rhd \mathbf{let}\ y : \alpha = zM\ \mathbf{in}\ N : \gamma$ is a FCF if (1) $\Gamma, y : \alpha \rhd N : \gamma$ is a FCF, (2) z has a type $\beta \Rightarrow \alpha$ in Γ, and (3) $\Gamma \rhd M : \beta$ is a FCF.

(iv) $\Gamma \rhd (\mathbf{case}\ x\ \mathbf{of}\ \ n_1 : M_1\ []\ n_2 : M_2\ []..[]\ n_k\ : M_k) : \alpha$ is a FCF if, for each i, $x_i : \iota \in \Gamma$, and, again for each i, $\Gamma \rhd M_i : \alpha$ is a FCF.

We note that (iii) and (iv) are just abbreviations for $\mathrm{PCF_V}$-terms, in the following way:

(i) $\Gamma \rhd \mathbf{let}\ y : \alpha = zM\ \mathbf{in}\ N : \beta$ stands for $\Gamma \rhd (\lambda y^{\alpha}.N : \beta)(zM)$.

(ii) $\Gamma \rhd \mathbf{case}\ y\ \mathbf{of}\ n_1 : M_1[]..[]n_k : M_k\ : \alpha$ stands for:

$$\mathsf{cond}\ (y = n_1)\ M_1(...(\mathsf{cond}\ (y = n_k)\ M_k\ \Omega)..)$$

where we assume an encoding of equality check of natural numbers in $\mathrm{PCF_V}$.

Remark. Before the proof of the definability, we give some intuition on the correspondence between strategies and FCFs, except the obvious ones. Essentially the use of finite canonical forms lies in making the order of evaluation (or, more exactly, the order of interaction) explicit.

(i) $\lambda x^{\alpha}.M : \alpha \Rightarrow \beta$ represents, after an initial O-signal at the domain, an action sequence of $\mathrm{PA}^{\alpha \Rightarrow \beta}\ \mathrm{OQ}^{\alpha}$, where OQ^{α} is justified by the first $\mathrm{PA}^{\alpha \Rightarrow \beta}$, then behaves as M.

(ii) $\Gamma, x_i : \gamma_1 \Rightarrow \gamma_2, \Delta \rhd \mathbf{let}\ y : \alpha = x_i M\ \mathbf{in}\ N : \beta$ first interacts at x_i by PQ^{γ_i}, then Opponent may ask at M (when γ_1 is a higher-order type) which, after some interaction, will be answered by Player, and then there is an Opponent Answer by OA^{γ_2}, and finally the actions move to N.

(iii) The case statement corresponds to the situation when, after some interactions, it acts according to the (view containing the) received ground value (here natural numbers). We only use one variable since an n-tuple of natural numbers can be reduced to one up to isomorphisms.

Below we establish our main result in this section. The proof shows how innocence, visibility and bracketing allow us to translate strategies into $\mathrm{PCF_V}$-terms. While the basic reasoning method follows Hyland and Ong [27], we give the proof in detail since it illustrates the basic correspondence between FCF's and compact strategies.

5.9. **Theorem.** (definability) For each compact element $\sigma : \mathbf{1} \rightarrow [\![\alpha]\!]$ for any $\mathrm{PCF_V}$-type α in $\mathcal{CBV}$, there is a FCF $F : \alpha$ such that $[\![F : \alpha]\!] = \sigma$. Conversely, the interpretation of any FCF is a compact element in the respective type.

Proof: The second half is by easy induction on the structure of terms. The first half is by induction on the cardinality of f_σ for a compact strategy σ of form $\otimes[\![\alpha_i]\!] \rightarrow [\![\beta]\!]$ where α_i, β are PCF-types, constructing a FCF σ°. Let $\Gamma = \{x_i : \alpha_i\}_{0 \le i \le n-1}$ (the choice of variables does not matter) and $\sigma : [\![\Gamma]\!] \rightarrow [\![\beta]\!]$ be compact. If $|f_\sigma| = 0$, let $\sigma^\circ = \Gamma \rhd \Omega : \beta$, which obviously satisfies the condition. Assume the statement holds up to m and let $|f_\sigma| = m+1$. Assume also, without loss of generality, that the first k variables in Γ are ι while others are function types. Notice σ is defined only for a finite subset of these tuples, since σ is compact. Let it be defined for l tuples, say $\{\langle n_{11}, .., n_{1k}\rangle, .., \langle n_{l1}, n_{l2}, ..., n_{lk}\rangle\}$. Writing

$$\mathbf{case}\ \langle x_1, x_2, .., x_m\rangle\ \mathbf{of}\ \langle n_{11}, n_{12}, ..., n_{1k}\rangle : M_1[]..[]\langle n_{l1}, n_{l2}, ..., n_{lk}\rangle : M_l$$

for the nested case statement $\mathbf{case}\ x_1\ \mathbf{of}\ n_{11} : \mathbf{case}\ x_2\ \mathbf{of}\ ...\ []n_{l1} : \mathbf{case}\ x_2\ \mathbf{of}\ ...$, we now set:

$$\sigma^\circ\ =\ \Gamma \rhd \mathbf{case}\ \langle x_1, .., x_k\rangle\ \mathbf{of}\ \langle n_{11}, .., x_{1m}\rangle\ : M_1[]..[]\langle n_{l1}, n_{l2}, ..., n_{lm}\rangle : M_l$$

where we give $\Gamma \triangleright M_j : \beta$ ($l \leq j \leq l$) in the following, according to how σ reacts to each specified k-tuple of natural numbers). In the subsequent reasoning we shall often confuse PCF-types and their interpretation.

Case (i): σ reacts by PA^{β}. I.e. by an initial P-signal at β.

(1) If $\beta = \iota$, the action has a form $m' \in \omega$, and is the final action by the switching condition (or alternatively by the bracketing condition), so we simply put, according to the corresponding integer, $M_j = m'$.

(2) If $\beta = \gamma_1 \Rightarrow \gamma_2$, the action at the initial sort of β is uniquely determined (since the only possible next O-action is a question at γ_1). By the construction of P-views, this question remains in the P-view until another question at γ_1 is asked. Thus the P-view of any action sequence after a question at γ_1 has a form:

$$\mathrm{OA}_1^{\otimes \alpha_i}\ \mathrm{PA}_2^{\beta}\ \mathrm{OQ}_3^{\gamma_1}\ s$$

where e.g. OQ^{γ_1} denotes an action at the initial sort of γ_1 (which is labelled OQ), with actions in s hereditarily justified by $\mathrm{OA}^{\otimes \alpha_i}$ or OQ^{β_1}. Because PA^{β} is uniquely determined, the effect of f_σ on such sequences can be considered as an innocent function on P-views of form $\mathrm{OA}_1^{(\otimes \alpha_i)\otimes\gamma_1} s$ which inherits the justification from the original P-view (actions originally justified by PA_2 is now justified by the initial O-signal) except that the original answer to OQ_3, given at γ_2, now becomes unjustified P-signal. Thus we have another innocent strategy of type $[\![\Gamma, z_2 : \gamma_1]\!] \rightarrow [\![\gamma_2]\!]$, say τ, and clearly we have $|\tau| \leq m$ (since at least one P-action, PA_2, is taken away). By induction there is a FCF $\Gamma, z_2 : \gamma_1 \triangleright N : \gamma_2$ which defines τ, using which we now set:

$$M_j \stackrel{\mathrm{def}}{=} \Gamma \triangleright \lambda z_2^{\gamma_1}.N : \beta,$$

which does realise the behaviour of σ after the specified initial O-signal at Γ, via the interpretation $[\![\cdot]\!]$.

Case (ii): σ reacts by PQ. I.e. σ reacts by a P-question at Γ, justified by the initial O-signal. Such PQ belongs to the k-th component of the tensor for some k, say α_k, and α_k should have a form $\gamma_1 \Rightarrow \gamma_2$. We now form a FCF for this behaviour, according to the form of γ_1.

(1) If $\gamma_1 = \iota$, then the strategy must have given some $m' \in \omega$ at PQ^{γ_1}, and Opponent can only answer at γ_2, say by OA_3. We now show the P-views after OA_3 always take a form:

$$\mathrm{OA}_1^{\otimes \alpha_i}\ \mathrm{PQ}_2^{\iota}\ \mathrm{OA}_3^{\gamma_2}\ s.$$

This is easy by induction: if only these three actions are done this is clear. If not, then we start from an O-action, say x_h, which is justified by some P-action, say x_j, and x_j cannot be PQ_2 (notice it can justify nothing), so it is in s. So the P-view is the view of x_j postfixed by $x_j x_h$, but by induction the former contains the initial three actions, hence done. We can now transform the part of f_σ on these P-views to an innocent function which acts on the P-views of form:

$$\mathrm{OA}_1^{(\otimes \alpha_i)\otimes\gamma_2}\ s$$

where the actions in s originally justified by OA_3 is now justified by (the new) OA_1, with the associated innocent strategy of type $\Gamma, y' : \gamma_2 \Rightarrow \beta$. Notice we simply omit the original action PQ_2^{ι} since it is fixed. Also notice, by construction, the innocent function is only defined for the specified k-tuple for the initial O-signal. Since the original PQ_2 is deleted, the innocent function has strictly less cardinality than f_σ, so by induction we can form a FCF $\Gamma, y' : \gamma_2 \triangleright N_2 : \beta$. Then let M_j be:

$$\Gamma \triangleright \mathbf{let}\ y' = x_k m'\ \mathbf{in}\ N_2 : \beta,$$

which, when interpreted, does act as σ, once given the same initial k-tuple at Γ.

(2) If $\gamma_1 = \theta_1 \Rightarrow \theta_2$, then immediately after PQ^{γ_1}, Opponent can either ask back a question at θ_1 or answer at γ_2. Indeed the P-views after the initial three actions always take form:

$$\mathrm{OA}_1^{\otimes \alpha_i}\ \mathrm{PQ}_2^{\gamma_1}\ \mathrm{OQ}_3^{\theta_1}\ s \tag{5.1}$$

or

$$\mathrm{OA}_1^{\otimes\alpha_i}\ \mathrm{PQ}_2^{\gamma_1}\ \mathrm{OA}_3^{\gamma_2}\ s', \tag{5.2}$$

as can be simply verified, just like the case (1) above. We now consider these two cases separately, form a FCF for each behaviour, and then combine them to form the required FCF.

Case (5.1). I.e. when an O-question at θ_1 is the third action in the P-view. We first note that, in (5.1), s cannot contain an action OA_2^{γ} justified by PQ_2 since any P-view has a form $\mathrm{O}_0\mathrm{P}_1\mathrm{O}_1$ ... $\mathrm{P}_n\mathrm{O}_n$ where each P_i justifies O_i. Further s can neither contain a free (initial) P-signal at β, say PA_j^{β}, since, in that case, PQ_2 should be answered before PA_j (see Proposition A.1 in Appendix A), which implies OQ_3 cannot occur in the P-view by the structure of P-views we noted above. We now transform the part of f_σ acting on (5.1) into an innocent function action on the P-views of the following form:

$$\mathrm{OA}_1^{(\otimes\alpha_i)\otimes\theta_1}\ s.$$

where the P-answer in s at θ_2 originally justified by $\mathrm{OQ}_3^{\theta_1}$ now becomes free. This gives us an innocent strategy of type $[\![\Gamma, w : \theta_1]\!] \to [\![\theta_2]\!]$. The innocent function has again obviously a smaller cardinality than f_σ, by induction we have the corresponding FCF, say, $\Gamma, w : \theta_1 \rhd N_1 : \theta_2$.

Case (5.2). I.e. when an O-answer at γ_2 is the third action in the P-view. We first notice, in (5.2), $\mathrm{PQ}_2^{\gamma_1}$ only (directly) justifies OA_3 in s', i.e. no other answers and questions in s' are justified by PQ_2, which is immediate from the form of P-views. Using this we can now transform that part of f_σ which maps the P-views of form (5.2) into an innocent function mapping P-view of form:

$$\mathrm{OA}_1^{(\otimes\alpha_i)\otimes\gamma_2}\ s'$$

corresponding to (5.2). Notice we can delete γ_2 since the original PQ_2 justifies only OA_3. The justification to actions originally justified by OA_1 and OA_3 are now given from (the new) OA_1. We thus have an innocent strategy of type $[\![\Gamma, y' : \gamma_2]\!] \to [\![\beta]\!]$ with a smaller cardinality than σ, corresponding to which we form a FCF $\Gamma,\ y' : \gamma_2 \rhd N_2 : \beta$.

Now putting N_1 and N_2 together, we construct M_j as:

$$\Gamma \rhd \mathbf{let}\ y' = x_i\lambda w^{\theta_1}.N_1\ \mathbf{in}\ N_2 : \beta,$$

which, when interpreted, asks at α_i after an initial O-signal, then, if Opponent questions back (at θ_1 corresponding to w), then interacts as N_1, and, after the answer comes (as γ_2 corresponding to y'), starts interacting as N_2, which does behave as σ when it receives the specified initial O-signal. By combining these M_j by the case statement, as we discussed at the beginning, we have now constructed the FCF corresponding to σ, as required. $\Box$

Let us write $[\![\Gamma \rhd M : \alpha]\!]_e$ for the interpretation of $\mathrm{PCF_V}$-terms in $\widehat{\mathcal{CBV}}$ given as $[[\![\Gamma \rhd M : \alpha]\!]]_{\precsim}$. We are now ready to prove:

5.10. **Theorem.** (full abstraction) Assume given closed $\mathrm{PCF_V}$-terms $M : \alpha$ and $N : \alpha$. Then we have $M : \alpha \preceq_{obs} N : \alpha$ iff $[\![M : \alpha]\!]_e \precsim [\![N : \alpha]\!]_e$.

Proof: For "if" direction, we show Proposition 5.6 implies the same result for $\widehat{\mathcal{CBV}}$, from which the result is immediate. By Proposition 4.4 (ii), $\sigma : \alpha$ is total in $\mathcal{CBV}$ iff it(s equivalence class) is so in $\widehat{\mathcal{CBV}}$, thus: $M \Downarrow V$ iff $[\![M]\!] \neq \bot$ in $\mathcal{CBV}$ iff $[\![M]\!] \neq \bot$ in $\widehat{\mathcal{CBV}}$, as required. For the "only if" direction, it suffices to show the left-hand side implies $[\![M : \alpha]\!] \precsim [\![N : \alpha]\!]$ in $\mathcal{CBV}$. We first note that we can easily replace $\mathbf{1}$ in the target by nat in Proposition 4.2, so to measure the observability at ι type suffices. Suppose $M \preceq_{obs} N$ and, moreover, for some $\tau : [\![\alpha]\!] \to \mathsf{nat}$, we have $[\![M]\!]; \tau = \overline{n}$ where $\overline{n} : \mathbf{1} \to \mathsf{nat}$ is an obvious strategy corresponding to n. We can always write $\tau = \sqcup \tau_i$ where τ_i is all compact, and if for all τ_i we have $[\![M]\!]; \tau_i = \bot$ then it should be $[\![M]\!]; \tau = \bot$

by continuity, so for some τ_i we have $[\![M]\!];\tau_i = \overline{n}$. Take the corresponding FCF $x : \alpha \triangleright \tau_i^\circ : \iota$. Let $F \stackrel{\text{def}}{=} \lambda x^\alpha.\tau_i^\circ : \iota$ which is again a FCF. Then $[\![M]\!];\tau_i = \langle\!\langle p\lambda(\tau_i),\, [\![M]\!]\rangle\!\rangle;\mathsf{ev} = [\![M]\!];[\![F]\!] = [\![FM]\!]$, using $\langle\!\langle p\lambda(\sigma_1),\, \sigma_2\rangle\!\rangle;\mathsf{ev} = \langle\!\langle \mathsf{id},\, \sigma_2\rangle\!\rangle;\sigma_1$ [because: $\langle\!\langle \mathsf{id},\, \sigma_2\rangle\!\rangle; p\lambda(\sigma_1)\otimes\mathsf{id};\mathsf{ev} = \langle\!\langle \mathsf{id}; p\lambda(\sigma_1),\, \sigma_2\rangle\!\rangle;\mathsf{ev}$ by totality of $p\lambda(\sigma_1)$ and id_A]. By Proposition 5.6 (computational adequacy), we know $FM \Downarrow n$, therefore $FN \Downarrow n$, which implies $[\![N]\!];[\![F]\!] = [\![N]\!];\tau_i = \overline{n}$ hence $[\![N]\!];\tau = \overline{n}$, as required. ☐

We remark that the above result can be generalised to open terms, using essentially the same argument.

6. Relationship with the category of call-by-name games

It is well-known that, in the context of domain theory, the call-by-name universe (say CPPO, the category of pointed cpo's and continuous functions) and the call-by-value universe (say pCPO, the category of cpo's and partial continuous functions) are mutually embeddable. In the following we discuss how an analogous result holds in the present setting. We start from introducing the universe of call-by-name games, which is a conservative extension of the original category of games by Hyland-Ong [27].

6.1. **Definition.**

(i) (cbn arenas) A *cbn arena* is a pre-arena in which initial sorts are all labelled by OQ.

(ii) (action sequences) Given cbn arenas A and B, an *action sequence from A to B* is defined as in Definition 2.5, reading "arenas" there as "cbn arenas", and changing (3) into: (3') $\underline{x_0}$ is initial in B and if x_j with $j \neq 0$ is not justified, *either* x_j is initial in B, in which case its preceding action is from B, *or* x_j is initial in $\overline{A}$.

(iii) (legal positions) An action sequence from a cbn-arena to a cbn-arena is *well-bracketed* precisely when the same condition as given in Definition 2.7 (i) holds. The *visibility condition* is given just as in Definition 2.7 (ii) except we change **(ov1)** as: $\mathsf{OV}(sx_j) = \{i, j\}$ if x_j is a free P-action, where x_i is the free O-action such that $i \in \mathsf{PV}(sx_j)$. An action sequence from a cbn arena to a cbn arena is *legal* when it is well-bracketed and satisfies the visibility condition.

(iv) (innocent strategy) Given cbn arenas A and B, an *innocent strategy from A to B*, or often simply a *strategy from A to B,* is defined precisely as in Definition 2.9, reading "arenas" as "cbn-arenas". We sometimes call such strategies, *cbn strategies*, to distinguish from strategies in $\mathcal{CBV}$. The set of cbn strategies between fixed cbn arenas is ordered by $\sigma \sqsubseteq \tau \;\Leftrightarrow\; \sigma \subseteq \tau$, just as in Definition 2.9.

(v) (composition) The composition of cbn strategies is defined exactly as in Definition 2.12, reading "arenas" as "cbn arenas".

Remark. In (ii), the extra condition (3') (which concerns the preceding action of an unjustified B-actions) is necessary because of the existence of multiple initial O-actions (which is possible because the initial sort is labelled as Question in cbn arenas), which otherwise can make the switching condition invalid.

6.2. **Definition and Proposition.** The category $\mathcal{CBN}$ is given by the following data.

- Objects: cbn-arenas.
- Arrows: strategies over cbn-arenas, composed by ; and ordered by $\sqsubseteq$.

Then $\mathcal{CBN}$ is cartesian closed and is enriched over CPPO, the category of pointed cpos and continuous functions.

We can then show the following:

6.3. **Proposition.** $\mathbb{CA}$ of [27] is a full subcategory of $\mathcal{CBN}$.

Proof Outline: Call a cbn-arena in which no answers justify questions, a *Hyland-Ong arena*. Then the full subcategory of $\mathcal{CBN}$ of Hyland-Ong arenas is isomorphic to $\mathbb{CA}$. As to objects, we associate each Hyland-Ong arena to an arena of $\mathbb{CA}$ of the corresponding shape. As to arrows, the form of strategies (or action sequences) differs because, in $\mathbb{CA}$, each sequence has only one unjustified action, which is initial; however by taking the representation by innocent function, it is immediate that two notions coincide. Finally we verify that the notion of composition in Definition 2.12 for cbn strategies coincide with the one given in Section 5.1 of [27]. This is best done by showing their "uncovering" coincides with the expanded form of composition given in Appendix A.12, transforming Definition 5.1.1 of [27] into the "board-form" as given in Appendix A.12. We omit the details. □

$\mathcal{CBN}$ is in close relationship with McCusker's category of Hyland-Ong games with linear decomposition [33] (the equivalence may hold but this has not been verified): As such, the extensional quotient of $\mathcal{CBN}$ allows a fully abstract interpretation of call-by-name FPC, and its variant allows interpretation of imperative constructs as in [6].

Our interest in this section, however, is how this universe relates to $\mathcal{CBV}$. First, the following result shows how we can simply embed $\mathcal{CBN}$ into $\mathcal{CBV}$.

6.4. **Theorem.** $\mathcal{CBN}$ is isomorphic to the full subcategory of $\mathcal{CBV}_{\mathrm{t}}$ of pointed arenas.

Proof: In the embedding, each cbn arena A is mapped to the pointed cbv arena A° which is the pre-arena $\mathbf{1} \uplus A$ together with one justification from $\mathbf{1}$ to each initial sort of A. For arrows, given a cbn strategy $\sigma : A \to B$, we define $\sigma^\circ : A^\circ \to B^\circ$ as $\{s^\circ | s \in \sigma\} \cup \{\varepsilon\}$ where we construct s° as follows (apart from injection of each element to adjust types): if $s = \varepsilon$ then $s^\circ = *$ where $*$ is the unique Opponent Answer at A. If $s \neq \varepsilon$ then we add the initial signal of A° followed by the initial signal of B° as the prefix of s, then let the first (resp. second) signal justify all originally free actions from B (resp. from A), following the graph structure of A° and B°. That this gives a functor is mechanical verification. It is full because the initial two actions in a total strategy between pointed cbv arenas are fixed. □

Notice that this is an exact analogue of the situation which arises in the domain-theoretic setting, where CPPO is precisely the subcategory of CPO of pointed types (note CPO is a "total" subcategory of pCPO). Just as the domain-theoretic setting, pointed arenas arise as the objects of Eilenberg-Moore algebra of the lifting monad in $\mathcal{CBV}$, as can be easily verified.

As we observed at the outset, it is a well-known fact in the domain theory that we can also embed pCPO into CPPO. The method is simple, we take the full-on-object sub-category of CPPO whose arrows are restricted to *strict* ones. A similar idea, though not an exact analogue this time, works in the present setting. We need the following notion.

6.5. **Definition.**

(i) A cbn arena is *pointed* when it has a unique initial sort, cf. 3.10. A cbn arena is *sharply pointed* when it is pointed and, moreover, it justifies no sorts labelled by Q.

(ii) Suppose A and B are pointed $\mathcal{CBN}$-arenas. Then a strategy $\sigma : A \to B$ is *strict* when, after the initial question, it immediately asks the question at A, or alternatively $\bot; \sigma = \bot$. It is *linear* if it is strict and never asks the (unique) initial question at A until another (unique) initial question at B is asked.

6.6. **Proposition and Definition.** If $\sigma : A \to B$ and $\tau : B \to C$ are linear cbn strategies then so is $\sigma; \tau$. The category of pointed types and linear strategies is denoted $\mathcal{CBN}_{\mathbf{1}}$.

Proof: Clearly strict strategies compose, while the linearity of $\sigma; \tau$ is because its action at A only comes from σ. $\square$

Remark. The use of linearity with call-by-value computation in mind is already made in the work by Harmer and Malacaria [32] in a different setting of games (using Abramsky-Jagadeesan-Malacaria games).

We can now state the second embedding result.

6.7. **Theorem.** $\mathcal{CBV}$ is isomorphic to the full subcategory of $\mathcal{CBN}_1$ of sharply pointed arenas.

Proof: At the object level, the embedding functor turns a cbv-arena A into a cbn-arena $A^\star$ by adding one sort labelled by OQ, which justifies what have been initial in the original cbv arena A. For strategies, we work with innocent functions. Given a cbv strategy $\sigma : A \to B$ and its innocent function f_σ, we construct a function g as follows: if $f_\sigma(s_1) = s_2$, then we set $g(s_1^\star) = s_2^\star$ where $s_1^\star$ and $s_2^\star$ are the result of adding the same prefix to s_1 and s_2, which consists of the unique initial action $*$ from $B^\star$ (which is an O-question) followed by the unique initial action $*'$ from $A^\star$ (which is a P-question). In the newly formed sequence, the former justifies what have been initial in B and the latter justifies what have been initial in A. In addition, we let $g(*) = **'$. Since it is easy to show $\{s, s' \mid g(s) = s'\}$ is prefix-closed, g does characterise a cbn innocent strategy $\sigma^\star$. Finally it is mechanical to verify that $(\cdot)^\star$ does define a full functor. $\square$

The two embedding results may best be seen as the simple translation of information flows. We also note that the full-on-object subcategory of $\mathcal{CBN}$ of strict maps, where the notion of strictness is generalised to non-pointed types in the obvious way, is isomorphic to $\mathcal{CBV}_{\mathtt{t}}^{\mathbf{T}}$, the latter being the category of Eilenberg-Moore algebras of the monad $\mathbf{T}$.

7. Discussions

7.1. **Related works.** The present work showed that the category of games originally introduced by Hyland and Ong [27] is simply extendable to the one for call-by-value by changing the parameter representing the form of information flow, and that the resulting category has the algebraic structures parallel to those of the foregoing domain-theoretic call-by-value universe such as pCPO, yet with a strong intensional flavour. The faithfulness with which this universe captures call-by-value sequential higher-order computation is exhibited by the full abstraction result for call-by-value PCF. In the following we discuss those works related to the present one.

- (i) After completing the full version of this paper [25], the authors were informed of an independent (and essentially concurrent) work by Riecke and Sandholm [52], in which they obtained a full abstraction for call-by-value FPC (which easily implies that of $\mathrm{PCF_V}$). The construction is based on Kripke logical relations on pCPO, and is thus quite different from the present one. No quotienting is necessary to reach the semantic universe so that, in this sense, their universe is more abstract than $\mathcal{CBV}$, while the construction of the universe itself is substantially more complicated than ours. In a brief comparison, one may say that their approach would give better insights for understanding why some (continuous) function is *not* sequential; while their construction does not directly model the dynamic aspects of sequential call-by-value computation, thus may not lead to the insights in that context. Thus two methods would play different roles in semantic analysis.
- (ii) Following the conference version of the present paper, Abramsky and McCusker [7] presented another categorical universe for call-by-value based on Hyland-Ong games. The universe, suggested by the present construction and also related to McCusker's early idea, is in close relationship with the presentation of $\mathcal{CBV}$ from $\mathcal{CBV}_{\mathtt{t}}$. The basic idea is to represent the initial Opponent/Player answers of total strategies by a certain indexing function. Thus a strategy is such an index function together with an innocent (and, this time, call-by-name, because the initial two actions are gone) strategy. Once the category of total strategies is made, we stipulate the required monadic structures and form the universe

of "partial" arrows by taking the Kleisli category. The construction is originally done as an algebraic clarification of the present construction (personal communication), and is applied to interpretation of certain imperative constructs by changing the base call-by-name category. An advantage of this approach would be, as we just noted, that one can use the foregoing call-by-name universes such as the one McCusker presented in [33] as a base universe (which corresponds to changing parameters of games in the present setting, see 7.2 later). On the other hand, the initial construction of the universe becomes somewhat complex, which may obscure, at a first glance, a couple of intensional elements of the present universe, such as the form of information flow. The construction however does offer a deep categorical analysis of semantic universes of call-by-value games such as $\mathcal{CBV}$, thus would contribute to a deeper understanding on game-based semantics of call-by-value computation.

(iii) In the setting of Abramsky-Jagadeesan-Malacaria games [4], Harmer and Malacaria are working on game semantics for call-by-value computation by extracting "linear" strategies from the original universe or its variant. Such a study would lead to the basic clarification of the notion of call-by-value in games, on the one hand, and to the deeper understanding on the relationship between the AJM games and the HO games, on the other hand. [32] gives a preliminary study in this direction.

7.2. **Extensions of full abstraction results.** We discuss a few extensions of our full abstraction results.

(i) The proofs in Section 5 easily extend to $\mathrm{PCF_V}$ with sums and products, considering strategies of type $\otimes\alpha_i \rightarrow \oplus\beta_j$ and using such isomorphisms as $\otimes_i \oplus_{i,j} \alpha_{i,j} \simeq \oplus_j \otimes_{i,j} \alpha_{i,j}$ and $\oplus_i\alpha_i \Rightarrow \beta \simeq \otimes_i(\alpha_i \Rightarrow \beta)$, as well as extending FCF's with $(M_1,\ M_2,\ \ldots, M_n) : \otimes\alpha_i$, $\mathbf{in}^i_n[M] : \oplus\alpha_i$, and a refined case statement.

(ii) Another immediate extension is the untyped call-by-value λ-calculus. For the arena, take initially a directed graph (not a forest) which has four sorts, respectively labelled by OA, OQ, PQ and PA, with directed edges OA $\mapsto$ PQ, PQ $\mapsto$ OA, OQ $\mapsto$ PA, PA $\mapsto$ OQ, OQ $\mapsto$ PQ, and PQ $\mapsto$ OQ. We then unfold this graph into the corresponding infinite tree in the standard way, setting OA as the root. This is essentially the canonical solution of the equation $D \simeq D \Rightarrow D$ and the full abstraction result easily follows. In the same way, we can directly interpret the lazy λ-calculus in $\mathcal{CBN}$ discussed in Section 6. It is interesting to compare these constructions with those given in [5]. Relatedly, Sieber [53] showed that how fully abstract domain-theoretic models for various extensions of PCF emerge from the one for the call-by-value PCF, in the presence of parallel-or. It is interesting to see whether a similar result may be obtained in the present setting.

(iii) For recursively typed calculi such as FPC [23], the traditional approach to the interpretation of types with free variables is not possible in $\mathcal{CBV}$ because tensors do not give bifunctors. There are a couple of methods to solve this issue. Using one of such approaches, Marcelo Fiore and the first author have successfully interpreted call-by-value FPC in $\mathcal{CBV}$ which is then shown to be fully abstract. In particular we used the axiomatic framework in [19], adapted to the present setting, to gain the computational adequacy. See [18] for details. We note that, while the situation may seem similar to the one in [33], there is a significant difference in that the definability argument for finite types can be carried out at the intensional level. As a related issue, it is interesting to see whether $\widehat{\mathcal{CBV}}$ is algebraically compact or not for general Poset-enriched functors, cf.[20, 21], which is left as a further issue.

(iv) We already discussed in Introduction that a simple change of other parameters may result in the semantic universe of call-by-value computation capturing various language features. In the context of call-by-name computation, there are recent proposals to treat imperative constructs by Abramsky and McCusker [6] (block structures) and by Laird [30] (control constructs). At least at the categorical level, the corresponding changes in the present setting result in coherent algebraic universes with distinct features, just by altering the concerned parameter. Indeed, just by changing the class of strategies following [6], we can simply construct a semantic universe in which an imperative language with first-order cells can be given a fully abstract translation (while its call-by-name version, formed just as $\mathcal{CBN}$ of Section 6, gives the interpretation of Idealised Algol just as in [6]). This shows that the parameter of games pertaining to the form of information flow (in other words that pertaining to the difference between call-by-name and call-by-value) is indeed independent from some of the basic parameters of games. Study in this direction, as well as the study of type structures arising in such a context, as well as the interpretation of call-by-value languages with "impure" constructs, would be an interesting subject for further research. One recent work in this direction (using the presentation of [7]) can be found in [3].

(v) A topic related to (iv) but not restricted to it is the study of refined type structures in the present and related semantic universes. This would lead to a further elucidation of algebraic structures of $\mathcal{CBV}$ and related universes.

(vi) One of the referees suggested that there is a similarity between Lorenzen's original games in the context of proof theory [15] and the present notion of games. While the formal relationship is not yet clear (partly due to the difference in presentation), it is an interesting open issue how the present approach, in which we try to understand call-by-value in terms of the forms of information flow, may relate to approaches from a logical viewpoint. In this context, Danos suggested a possible connection of the present notion of call-by-value games to a proof system in [14], where two systems with dual characters, conceptually corresponding to call-by-name and call-by-value, are presented. It would be worth studying how this distinction between call-by-name and call-by-value from a proof-theoretic perspective may relate to the same distinction in the present context.

7.3. **Intensionality and relationship with process theories.** The strongly intensional character of $\mathcal{CBV}$ is not at the same level of abstraction as, say pCPO. The same can be said about its call-by-name counterpart and other categories of games, in the sense that they reflect some notion of execution, albeit abstractly, cf. [13, 27]. From the viewpoint that the primary purpose of semantic representation of programming languages lies in giving (in)equations over programs as general as possible, this feature may be considered as a drawback. However we can take a different perspective, and ask whether this novel way of representing programs can be put to a significant use, especially once given the full abstraction result as the semantic justification of the representation. As a first such step, one may exploit the representation for the development of abstract theory of execution, including the formal optimisation techniques. Type structures as we studied in Section 4 may be put to an effective use in this context. One interest in this regard is that our interpretation of $\mathrm{PCF_V}$ in $\mathcal{CBV}$ already gives a simple abstract implementation of the language in the form name passing processes [39]. The representation is closely related to Milner's direct encoding in [36] after doing a simple optimisation, performing the β_v-reduction by three name passing interactions (excepting the buffering actions, cf.[36]). Such a "physical" character of the abstract universe suggests that we may study the execution of, say, call-by-value programming languages from a new level of mathematical abstraction (this is in line with Girard's studies on the semantics of cut elimination [22]). Relatedly the induced encodings also suggest the possibility of relating game semantics and process theories at the fundamental level. The study of behavioural types by Milner [37] may suggest possible directions (from which the present study actually started).

References

[1] Abelson, H. and Sussman, G.J., *Structure and Interpretation of Computer Program*, The MIT Press, 1985.

[2] Abramsky, S. and Jagadeesan, R., Games and Full Completeness for Multiplicative Linear Logic, Journal of Symbolic Logic, 59(2), pp. 543–574, 1994.

[3] Abramsky, S., Honda, K. and McCusker, G., A Fully Abstract Game Semantics for General References. *Logic in Computer Science*, July 1998.

[4] Abramsky, S., Jagadeesan, R. and Malacaria, P., Full Abstraction for PCF, 1994. To appear.

[5] Abramsky, S. and McCusker, G., Games and Full Abstraction for the Lazy λ-calculus, *LICS'95*, pp. 234–243, 1995.

[6] Abramsky, S. and McCusker, G., Linearity, Sharing and State: a fully abstract game semantics for Idealized Algol with active expressions, ENTCS, Vol.3, North Holland, 1996.

[7] Abramsky, S. and McCusker, G., Call-by-value games, CSL'97, August, LNCS, Springer-Verlag, 1997.

[8] Barendregt, H., *The Lambda Calculus: Its Syntax and Semantics.* North Holland, 1984.

[9] Berry, G. and Curien, P. L., Sequential algorithms on concrete data structures *Theoretical Computer Science*, 20(3), pp. 265-321, July 1982.

[10] Blass, A., A Game Semantics for Linear Logic, *Annuals of Pure and Applied Logic*, 56:183–220, 1992.

[11] Boreale, M. and Sangiorgi, D., Fully Abstract Semantics for Causality in the π-calculus. LFCS report, ECS-LFCS-94-297, University of Edinburgh, 1994.

[12] Curien, P. L., Sequentiality and full abstraction. In Proc. of *Application of Categories in Computer Science*, LNM 177, pp.86–94, Cambridge Press, 1995.

[13] Danos, V., Herbelin H. and Regnier, L., Games Semantics and abstract machines. *LICS'96*, IEEE, 1996.

[14] Danos, V., Joinet, J-B, and Schellinx, H. Sequent calculi for second order logic. In *Advances in Linear Logic.* Pages 211-224. Cambridge University Press, 1993.

[15] Felshcer, W., Dialogue games as a foundation for intuitionistic logic, *Handbook of Philosophical logic*, Vol.3, pp.341–372, D. Reidel Publishing Company, 1986.

[16] Fiore, M., *Axiomatic Domain Theory in Category of Partial Maps*, PhD thesis, University of Edinburgh, 1994. Published by Cambridge University Press in Distinguished Dissertations Series, 1996.

[17] Fiore, M., An Enrichment Theorem for an Axiomatisation of Categories of Domains and Continuous Functions, *Math. Struct. in Comp. Science*, 1996.

[18] Fiore, M. and Honda, K., Recursive Types in Games: axiomatics and process representation, *Logic in Computer Science*, July 1998.

[19] Fiore, M. and Plotkin, G., An Axiomatisation of Computationally Adequate Domain Theoretic Models of FPC, *LICS'94*, pp.92–102, IEEE, 1994.

[20] Freyd, P., Algebraically Complete Categories, In Proc. of *Como. Category Theory Conference*, LNM 1488, pp.95–104, Springer Verlag, 1991.

[21] Freyd, P., Remarks on Algebraically Complete Categories, In Proc. of *Application of Categories in Theory Conference*, LMS 177, pp.95–106, Cambridge University Press, 1992.

[22] Girard, J.-Y., Linear Logic, *Theoretical Computer Science*, Vol. 50, pp.1–102, 1987.

[23] Gunter, C., *Semantics of Programming Languages: Structures and Techniques*, MIT Press, 1992.

[24] Honda, K. and Yoshida, N., Name-Passing Games (II): A Functional Universe, Typescript, 35 pp. November, 1996.

[25] Honda, K. and Yoshida, N., Game-theoretic analysis of call-by-value computation (short version). *Proc. ICALP'97, Proceedings of 24th International Colloquium on Automata, Languages, and Programming*, Lecture Notes in Computer Science 1256, pp.225–236, Springer-Verlag, July, 1997.

[26] Honda, K. and Yoshida, N., Game-theoretic analysis of call-by-value computation (a full version), 43pp. Announced at the type mailing list, February, 1997. ftp-able at `ftp.dcs.ed.ac.uk/export/kohei/cbvfull.ps.gz`, Feb, 1997.

[27] Hyland, M. and Ong, L., "On Full Abstraction for PCF": I, II and III. 130 pages. ftp-able at `theory.doc.ic.ac.uk/papers/Ong`, 1994.

[28] Hyland, M. and Ong, L., Pi-calculus, dialogue games and PCF, Proceedings of 7th Annual ACM Conference on Functional Programming Languages and Computer Architecture, June 26-28, 1995.

[29] Kahn G. and Plotkin, G. D., Concrete domains, *Theoretical Computer Science*, Vol. 121, Number 1-2, pp. 187-277, December 1993. Originally appeared as INRIA Report 336, 1978.

[30] Laird, J., Full abstraction for functional languages with control, *LICS'97*, IEEE, 1997.

[31] Longo, G. and Moggi, E., Cartesian closed categories of enumerations for effective tyep-structures, LNCS 173, Springer-Varlag, 1984.

[32] Harmer, R., Malacaria, P., Linear foundations of game semantics, a typescript, Sep. 1996.

[33] McCusker, G., Games and Full Abstraction for FPC. *LICS'96*, IEEE, 1996.

[34] McCusker, G., *Games and Full Abstraction for a Functional Metalanguage with Recursive Types*, Ph.D. Thesis, Imperial College, 1996.

[35] Milner, R., *A Calculus of Communicating Systems*, LNCS 76, Springer-Verlag, 1980.

[36] Milner, R., Functions as Processes. *Mathematical Structure in Computer Science.* 2(2), pp.119–146, 1992.

[37] Milner, R., Sorts and Types of π-Calculus, a manuscript, 43pp.

[38] Milner, R., Polyadic π-Calculus: a tutorial. *Proceedings of the International Summer School on Logic Algebra of Specification*, Marktoberdorf, 1992.

[39] Milner, R., Parrow, J.G. and Walker, D.J., A Calculus of Mobile Processes, *Information and Computation* 100(1), pp.1–77, 1992.

[40] Milner, R., Tofte, M. and Harper, R., *The Definition of Standard ML*, MIT Press, 1990.

[41] Moggi, E., Partial morphisms in categories of effective objects, *Information and Computation*, 76:250–277, 1988.

[42] Moggi, E., Notions of Computations and Monads. *Information and Computation*, 93(1):55–92, 1991.
[43] Nickau, M., Hereditarily Sequential Functionals, LNCS 813, pp.253–264, Springer-Verlag, 1994.
[44] Peter W. O'Hearn and Jon G. Riecke, Kripke Logical Relations and PCF, *Information and Computation*, 120:1, pp.107–116, 1995.
[45] Ong, L., Correspondence between Operational Semantics and Denotational Semantics, *Handbook of Logic in Computer Science*, Vol. 4, pp.269–356, Oxford University Press, 1995.
[46] Plotkin, G., Call-by-name, call-by-value and the lambda-calculus, *TCS*, 1:125–159, 1975.
[47] Plotkin, G., LCF considered as a programming language, *TCS*, 5:223–255, North-Holland, 1975.
[48] Plotkin, G., Lecture on Predomains and Partial Functions. Notes for a course given at the Center for the Study of Language and Information, Stanford 1985, 1985.
[49] Pierce, B.C. and Sangiorgi. D, Typing and subtyping for mobile processes. *LICS'93*, pp.187–215, IEEE, 1993.
[50] A.J. Power and E.P. Robinson, Premonoidal categories and notions of computation, Proc. LDPL 95, Math. Structures in Computer Science 7 (1997) 453-468.
[51] Priami, C., *Enhanced Operational Semantics for Concurrency*, Ph.D. thesis, Department of Computer Science, Universitá di Pisa, 1995.
[52] Riecke, J., and Sandholm,A. Relational Account of Call-by-value Sequentiality, *LICS'97*, 1997.
[53] Sieber, K., Relating Full Abstraction Results for Different Programming Languages, *10th FST/TCS*, pp. 373–387, LNCS 472, Springer-Verlag, 1990.
[54] Winskel, G., Synchronization Trees, *TCS*, Vol. 34, pp. 33–82, North-Holland, 1985.
[55] Winskel, G., *The Formal Semantics of Programming Languages: An Introduction*, MIT Press, 1993.
[56] Winskel, G. and Nielsen, M., Models for Concurrency, *Handbook of Logic in Computer Science*, Vol. 4, pp.1–148, Oxford University Press, 1995.

Appendix A. Basic Operational Structures of Call-by-value Games

This appendix gives the proofs omitted from Sections 2 and 3, which pertain to the basic operational structure of call-by-value games. Each proof is independent, though we often use a matrix (game board) presentation of action sequences and their composition, whose formal definitions are given in A.6 and A.12.

Remark. Basic proofs for operational structures of Hyland-Ong games can be found in Hyland and Ong [27] and, later in a somewhat different framework, in McCusker's thesis[34], both in the setting of call-by-name computation. [27] is nearer to the present one in that we start from as small restriction on the operational structure as possible. However a notable difference in presentation exists (for example the composition of strategies), apart from the difference between call-by-name and call-by-value. [34] shares a couple of important technical points with the present construction, in particular in the treatment of bracketing. However, in addition to the difference between call-by-value and call-by-name, a few basic differences exist in operational structures, notably whether the projection convention is stipulated (as in [34]) or derived (as here). Thus, for the sake of the completeness, we believe the listing of the verification for basic operational properties of call-by-value games may not be superfluous. Many properties are in common with the call-by-name games: even in such cases, we try to give a reasoning which is intuitive and which reveals the internal structure of games as clearly and intelligibly as possible (one such effort may be seen in our proof for P-view projection in A.4–A.8).

We start with the following proposition, which is often used later. It shows that the present simpler bracketing condition in Definition 2.7 (i) is equivalent to the one in the conference version [25].

A.1. **Proposition.** Let $s \stackrel{\text{def}}{=} x_0..x_j..x_{n-1}$ be a well-bracketed action sequence from A to B and x_j $(j \neq 0)$ is an initial P-signal from B (which is only such in s by Definition 2.5 (3)). Then all questions occurring in s before j are answered by questions occurring before j.

Proof: Using the labels like OA etc. to denote actions of that kind, suppose by way of contradiction we have the following situation in s:

$$\mathrm{OA}_0 \ldots \mathrm{Q}_i \langle \ldots . \rangle \mathrm{PA}_j \langle ::::: \rangle \mathrm{A}_k \ldots$$

where PA_j is the initial P-signal from B and Q_j is answered by A_k, i.e. $i \curvearrowright k$, while $\langle ....\rangle$ and $\langle :::::\rangle$ denotes the subsequences in the respective positions. We use the following notion: a subsequence t of an action sequence is *closed w.r.t. bracketing*, or simply *closed*, when all questions in t are answered in t and each answer in t answers some question in t. Notice a closed subsequence always has an even number of actions (which can be zero). With this preparation, assume i is the maximum such that $i \curvearrowright k$, i.e. no question between i and j is answered by an answer occurring after j. Notice this assumption implies a subsequence $\langle ....\rangle$ between i and j is closed by the bracketing condition. Similarly $\langle :::::\rangle$ is closed, thus both subsequences are of even length. But this means Q_i and A_k are both O-actions by strict alternation, so that $i \curvearrowright k$ cannot be the case, a contradiction. $\square$

We next prove two basic properties of legal positions. We start from the closure properties of legal positions.

A.2. **Proposition.** If s is a legal action sequence from A to B, then so are $\ulcorner s \urcorner$ and $\llcorner s \lrcorner$ as well as prefixes of s.

PROOF: The closure under prefix is by definition. For the closure under view functions, we first note that the resulting sequence always inherits the original justification by the visibility condition, and strict alternation and other conditions for action sequences are immediate from the construction, so that both views are indeed action sequences. Thus we have only to show they are legal.

(visibility) Take for example the case of the P-view of some sequence. We first observe each O-action is justified by the P-action which immediately precedes the O-action, i.e. it has a from

$$\mathrm{O}_0\mathrm{P}_1\mathrm{O}_1...\mathrm{P}_i\mathrm{O}_i...\mathrm{P}_n\mathrm{O}_n$$

where each P_i justifies O_i. Thus trivially each O-action in a P-view is justified in its view, while the view of a P-action goes through all preceding O-actions in the sequence, so its justifier (if any) surely occurs among them. Similarly for O-views which are OP-dual to P-views (except the initial action).

(bracketing).[3] We use induction on the length of a legal action sequence. We again take the case of P-view. Base cases are **(pv0,1)**, where the resulting views are obviously well-bracketed. For **(pv2)**, we have $\ulcorner s_0 x_i^P s_1 x_j^O \urcorner = \ulcorner s_0 \urcorner x_i^P x_j^O$. If x_j is a question, $\ulcorner s_0 \urcorner x_i^P x_j^O$ is well-bracketed since $\ulcorner s_0 x_i^P \urcorner = \ulcorner s_0 \urcorner x_i^P$ is so by induction. If x_j is an answer, $\ulcorner s_0 \urcorner$ is well-bracketed by induction and x_i is a question answered by x_j, so we again get a well-bracketed sequence. For **(pv3)**, we have $\ulcorner s_0 x_j^P \urcorner = \ulcorner s_0 \urcorner x_j^P$. If x_j is a question this P-view is well-bracketed by induction. If x_j is an answer to some question, say x_i, in s_0, the original sequence has a form:

$$s \stackrel{\text{def}}{=} s_0' \; x_i^{\mathrm{OQ}} \; s' \; x_j^{\mathrm{PA}} \tag{A.1}$$

where x_j answers x_i. We now show:

Claim. In $\ulcorner s \urcorner$, the subsequence corresponding to s' of s has the following form:

$$\mathrm{PQ}_0 \cdot \mathrm{OA}_0 \cdot \mathrm{PQ}_1 \cdot \mathrm{OA}_1 ... \mathrm{PQ}_{n-1} \cdot \mathrm{OA}_{n-1}$$

where each OA_i answers PQ_i.

PROOF: Assume there are n P-actions in s' which remains in $\ulcorner s \urcorner$. If $n = 0$ the statement is vacuous so suppose not. We show by induction each $n-k$-th P-action ($1 \le k \le n$) satisfies the above mentioned property. The base case is when $k = 1$, in which case the O-action preceding x_j can only be an answer (since if not x_j wrongly answers to it) hence let it be OA_{n-1}, and its justifier can only be a question, say PQ_{n-1}, as required. For induction, s' is now of form:

$$s'' \cdot \mathrm{O}_{n-k-1} \;\cdot\; \mathrm{PQ}_{n-k} ... \mathrm{OA}_{n-k} \cdot \mathrm{PQ}_{n-k+1} ... \mathrm{OA}_{n-k+1} \cdot \mathrm{PQ}_{n-1} ... \mathrm{OA}_{n-1}$$

[3] For an alternative short proof, see McCusker's thesis [34, Lemma 3.1.1].

with the justification from OA_i to PQ_i for each i. Thus if O_{n-k-1} is a question it violates the bracketing condition hence O_{n-k-1} is an answer, so we again get OA_{n-k-1} and PQ_{n-k-1}, as required. [end of the proof of the claim]

By the claim all questions in the subsequence between x_i and x_j in $\ulcorner s \urcorner$ are answered within it, thus we now know $\ulcorner s \urcorner$ is well-bracketed. For O-views, we have exactly the same reasoning (for **(ov3)** any P-action which we meet in s' when we trace back in O-view, cannot be initial, because of Proposition A.1), so that we now know the views of legal action sequences are indeed again legal action sequences. $\square$

A.3. **Proof of Proposition 2.8 (switching condition).** Following Hyland and Ong, this is proved simultaneously with:

> (O-view projection) Suppose $s \in \sigma : A \rightarrow B$ ends with a P-action from A (resp. from B), then $\llcorner s \lrcorner = \llcorner s \restriction A \lrcorner = \llcorner s \lrcorner \restriction A$ (resp. $\llcorner s \lrcorner = \llcorner s \restriction B \lrcorner = \llcorner s \lrcorner \restriction B$).

by induction on the length of $s \stackrel{\text{def}}{=} s'x_i$. For the base case, $s' = x_0$ with x_0 a free O-action in A, in which case two statements are obvious. For the inductive step, we first show O-view projection. If x_i is free, the result is trivial. So suppose x_i is not free and is in, say, A (the case it is in B is the same). We can then write $s = s_0 x_0 s_1 x_i$ where x_0 is an O-action binding x_i. We first notice x_0 is in A, and by induction for the switching condition, the last action of s_0 should be also in A. Now we have:

$$\begin{array}{rcll} \llcorner s_0 x_0 s_1 x_i \lrcorner \restriction A & = & (\llcorner s_0 \lrcorner x_0 x_i) \restriction A & (\text{by } \mathbf{(ov2)}) \\ & = & (\llcorner s_0 \lrcorner \restriction A) x_0 x_i & \\ & = & \llcorner s_0 \lrcorner x_0 x_i \;=\; \llcorner s \lrcorner & (\text{induction on } s_0) \end{array}$$

For the switching condition, suppose $s = s_0 x_i x_{i+1}$, x_i is a P-action in A and x_{i+1} is an O-action in B. If x_{i+1} is in B, then by visibility it should be bound by some action x_j of B in $\llcorner s_0 x_i \lrcorner$. But by inductive hypothesis of the O-view projection, $\llcorner s_0 x_i \lrcorner \restriction A = \llcorner s_0 x_i \restriction A \lrcorner$, which says x_j cannot be in $\llcorner s_0 x_i \lrcorner$, hence a contradiction. The case x_i is in B and x_{i+1} is in A is exactly the same. $\square$

We also prove a technical lemma on legal action sequences which we use later. Henceforth we write $s \preccurlyeq s'$ when s' is a (not necessarily consecutive) subsequence of s', inheriting the justification relation (the justification relation may not be uniquely determined by $\underline{s'}$ and s, but this does not concern us here).

A.4. **Lemma.** (P-view projection) Let $s \in \sigma : \overline{A_1} \uplus A_2$ and suppose s ends with an A_i-action, $i = \{1, 2\}$. Then we have: $\ulcorner s \urcorner \restriction A_i \preccurlyeq \ulcorner s \restriction A_i \urcorner$.

One may notice the statement is simpler than the corresponding Proposition 4.4.4 of [27] since we have only one, if any, occurrence of initial actions in a respective component. The proof of the lemma is done based on the idea of Hyland and Ong in the proof of the corresponding Lemma in Appendix A of [27], with a simplification of an argument. It is given in a form more general than necessary, assuming that multiple free occurrences of P-actions and O-actions are possible, so that it is also valid for, say, call-by-name strategies. We first list basic notions we shall use. (ii) is a key idea used in the proof, coming from [27].

A.5. **Definition.** Let $s \in \sigma : A_1 \rightarrow A_2$ below and $s', .., \gamma, .., \theta, ..$ range over its consecutive subsequences.

(i) (block) An A_i-*block* in s ($i = 1, 2$) is a consecutive subsequence (including the null sequence) of s which solely consists of A_i-actions.

(ii) (bounded segment, cf. p.121 in [27]) An A_i-*bounded segment* ($i = 1, 2$) in s is a consecutive subsequence of s which has a form $w s_0 w'$ where (1) $w \mapsto w'$, (2) w is a P-action and w' is an O-action, and (3) both are from A_i. w and w' are called its *boundary*. An A_i-*open* *segment* is a singleton subsequence w' such that w' is a free O-action from A_i, which itself is its boundary.

(iii) (block sequence) An A_i*-block sequence* in s is a consecutive subsequence s' of s such that, assuming $i \neq j$, $s' = \gamma_0\theta_0\gamma_1\theta_1...\gamma_n\theta_n$ $(n \geq 0)$ where each θ_k $(k \geq 0)$ is an A_i-block, each γ_k $(k \geq 1)$ is an A_j-bounded segment, and either (1) γ_0 is also an A_j-bounded segment, in which case s' is *non-terminal*, or (2) γ_0 is a singleton sequence of a free O-action from A_j, in which case s' is *terminal*. Each θ_i is called its *component block*, and each γ_i $(i \geq 0)$ is called its *component segment* (which are together called simply *components*).

Note that components are uniquely determined once we are given a block sequence, and that any $s \in \sigma : A_1 \rightarrow A_2$ which ends with an A_i action and which contains at least one A_j action $(i \neq j)$ can be, in general non-uniquely, written as s_1s_2 for an A_i-block sequence s_2, as an immediate consequence of the switching condition. Also observe that an A_i-bounded segment may naturally contain A_i-blocks and A_j-bounded segments which are not its components.

From now on we often use a board presentation of strategies. The idea is formalised as follows. By a *partial matrix* we mean a matrix whose indexing function is partial, i.e. some of whose co-efficients may not be filled. We count the columns/rows from 1, as is customary.

A.6. **Definition.** (board presentation of action sequences) Let $s \stackrel{\text{def}}{=} x_0..x_n$ be an action sequence from A_1 to A_2. The *game board presentation of* s is a partial $2 \times \omega$ matrix whose all and only defined coefficients are given as follows (here $x \restriction A_i$ denotes the original element of a sort in A_i when x is its injection):

if $j - 1$ is an index of s and x_{j-1} is from A_i $(i = 1, 2)$, then the coefficient at the i-th row, j-th column is $x \restriction A_i$.

together with the justification relation over the defined coefficients corresponding to the original justification relation. We usually call the rows of the matrix by the associated arenas (sometimes writing e.g. $\overline{A}$ to make its presentation in the pre-arena explicit).

We now prove a simple lemma on block sequences, using which we prove the P-view projection. Below we say the P-view (resp. O-view) of s *touches* an action (index) in s if it is in $\mathsf{PV}(s)$ (resp. $\mathsf{OV}(s)$), i.e. the view is constructed using the action.

A.7. **Lemma.** (block lemma) Let $s \in \sigma : A_1 \rightarrow A_2$ and $s = s_1s_2$ when s_2 is an A_i-block sequence. Then for each action x_i in any component block of s_2, (1) x_i is hereditarily justified in the component blocks as well as, if s_2 is non-terminal, in s_1, and (2) the P-view of x_i touches only actions in the component blocks, the boundaries of the component segments, and, if s_2 is non-terminal, those in s_1; Similarly for the O-view of x_i, excepting the boundaries of the component segments.

PROOF: We let $s_2 = \gamma_0\theta_0\gamma_1\theta_1...\gamma_n\theta_n$, $n \geq 0$ and reason by induction on n. For the base case, we have $n = 0$. If $\theta_0 = \varepsilon$, there is nothing to prove, so assume not. First let s_2 be non-terminal, so that $s_2 = \gamma_0\theta_0$ where $\gamma_0 = ws_0w'$ is a A_j-bounded segment and θ_0 is an A_i block. The situation can be depicted as:

$$\begin{array}{lllll}
 & & & x_0y_0x_1y_1....x_iy_i & = \theta_0 \\
A_1 & ... & & \mathrm{P\ O\ P\ O....P\ O} & \\
A_2 & ... & \mathrm{P} \cdots \mathrm{O} & & \\
 & & w \;\; s_0 \;\; w' & &
\end{array}$$

We argue by induction on the length of θ_0. For the base case, $\theta_0 = x_0$, and x_0 is a P-action by the switching condition. The P-view of x_0 goes to w and then to w' thus reaching s_1 without touching s_0, which also shows it can only be justified (if ever) in s_1. This in turn shows its O-view can only touch actions in s_1, as required. For the induction step, suppose $\theta_0 = s'yx$ with y being a P-action (resp. O-action) and that the statement holds for $s'y$. Then the O-view (resp. P-view) of x is x together with the O-view (resp. P-view) of y, and this only touches θ_0 (resp. θ_0, w and w') by inductive hypothesis, so in particular it is justified (if ever) within $s'y$ and s_1. This shows, using induction as necessary, that its P-view (resp. O-view) only touches θ_0, s_1 and $\{w, w'\}$ (resp. θ_0 and s_1), as required. The case when s_2 is terminal is proved in the same way, first noticing, for the base case as above, x_0 should be free, so that its O-view is itself and its P-view only goes to w', then reasoning for the induction step as above. This concludes the case for $n = 0$. For the inductive step, we assume the statement holds for some n, and establish the case $n + 1$, which is depicted in the following figure:

$$
\begin{array}{llllllllll}
 & & & & & & & & x_0 y_0 x_1 y_1 \ \ldots.\ x_i y_i & = \theta_{n+1} \\
A_1 & \ldots & & \theta_0 & \theta_1 & \ldots & \theta_n & & \mathrm{P\ O\ P\ O}\ \ldots.\ \mathrm{P\ O} & \\
A_2 & \ldots & \gamma_0 & & \gamma_1 & \ldots & \gamma_n & \mathrm{P} \cdots \mathrm{O} & & \\
 & & & & & & & w \ \ s_0 \ \ w' = \gamma_{n+1} & &
\end{array}
$$

Then we can readily verify that the same reasoning as the base case suffices, arguing by induction on the length of θ_{n+1}, replacing θ_0 with $\theta_0, \theta_1, .., \theta_n$, as well as w, w' with the boundaries of $\gamma_0, \gamma_1, .., \gamma_n$, in the argument above. $\square$

A.8. **Proof of Lemma A.4.** We can now establish the P-view Projection. Let $s = s'x \in \sigma : A_1 \rightarrow A_2$ and x is an action of, say, A_1 (the argument is the same exchanging A_1 and A_2 below). We use induction on the length of s to prove $\ulcorner s \urcorner \upharpoonright A_1 \preccurlyeq \ulcorner s \upharpoonright A_1 \urcorner$. For the base case, $s' = \varepsilon$, in which case the result is immediate. For induction we do a case analysis.

(i): x is a P-action and $s' = s''y$ with y being an O-action of A_1. In this case, we use induction to calculate:

$$\ulcorner s \urcorner \upharpoonright A_1 \ = \ \ulcorner s''yx \urcorner \upharpoonright A_1 \ = \ \ulcorner s''y \urcorner \upharpoonright A_1\, x \ \preccurlyeq \ \ulcorner s''y \upharpoonright A_1 \urcorner x \ = \ \ulcorner s''yx \upharpoonright A_1 \urcorner$$

as required.

(ii): x is a P-action and $s' = s''y$ with y being an O-action of A_2. Then we have either:

(a) $s = s_0 y' \xi_1 .. \xi_n x$ $(n \geq 0)$ where each ξ_i is an A_2-bounded segment and y' is a free O-action of A_2 (note $y' = y$ if $n = 0$).

(b) $s = s_0 \xi_0 .. \xi_n x$ $(n \geq 0)$ where each ξ_i is an A_2-bounded segment and s_0 ends with an O-action of A_1.

If (a) holds, immediately $\ulcorner s \urcorner \upharpoonright A_1 = \ulcorner s \upharpoonright A_1 \urcorner$. For (b), we first note $\ulcorner s \urcorner \upharpoonright A_1 = \ulcorner s_0 \urcorner \upharpoonright A_1 x$ since each ξ_i is A_2-bounded. The structure of $\ulcorner s \upharpoonright A_1 \urcorner$ is clarified by the following claim due to Hyland and Ong, cf. Lemma A.1.3 (i) of [27], which says $\ulcorner s x^{A_i} \urcorner$ orderly shrinks $\ulcorner s x^{A_i} \upharpoonright A_i \urcorner$ without jumping over the boundaries.

Claim. Suppose $s = s_1 w s_2 w' \in \sigma : A_1 \rightarrow A_2$ where $w s_2 w'$ is an A_i bounded segment. Then there are s', s'' such that: (1) $s = s's''$ and s' is a prefix of s_1 (2) s'' is an A_i block sequence, and (3) w occurs in a component block of s''. That is, we have the following situation between w and w'.

$$
\begin{array}{lllllllll}
 & & w & & & & & w' \\
A_1 & \ldots & \mathrm{OP..O} & & \theta_0 & \theta_1 & \ldots & \theta_n = \mathrm{PO...O} \\
A_2 & \ldots & & \gamma_0 & & \gamma_1 & \ldots\ \gamma_n &
\end{array}
$$

For the proof, suppose there is no s', s'' which satisfy (1) and (2), then there is a terminal A_i-block sequence as a postfix of $s_2 w'$, but this contradicts $w \mapsto w'$ by Lemma above, so we can find s', s'' satisfying (1) and (2), in which case w occurs at least within s''. But Lemma A.7 also says that if an A_i-action occurs in a component block of an A_i block sequence, and it is justified by another action in the same block sequence, then the latter should occur in some component block of the block sequence, therefore (3) is also satisfied. [end of the proof of the claim]

Returning to the case (b), we notice each ξ_i is an A_2-bounded sequence, so that we have:

$$\ulcorner s \restriction A_1 \urcorner = \ulcorner s_0 \xi_0..\xi_{n-1} \restriction A_1 \urcorner \delta_n x$$

where δ_n is the sequence of the boundaries of the component (bounded) segments of ξ_n. In this way we can write $\ulcorner s \restriction A_1 \urcorner = \ulcorner s_0 \urcorner \delta_0..\delta_n x$, therefore:

$$\ulcorner s \urcorner \restriction A_1 \;=\; \ulcorner s_0 \urcorner \restriction A_1\, x \;\preccurlyeq\; \ulcorner s_0 \restriction A_1 \urcorner x \;\preccurlyeq\; \ulcorner s_0 \restriction A_1 \urcorner \delta_0..\delta_n x \;=\; \ulcorner s \restriction A_1 \urcorner,$$

as required.

(iii): $s = s'x$ and x is an O-action. Then either x is free, in which case the result is immediate, or $s' = s''y$ with y is a P-action of the same component by the switching condition, in which case we can use (i) and (ii) above. This exhausts all cases. □

Using P-view projection together with other results, we can now prove another basic property of legal action sequences. Given an arena A, an *action sequence in* A is a sequence of elements from sorts of A together with the relation $\curvearrowright$ on its indices just as Definition 2.5, except (3) is replaced by a simpler: (3) x_0 is initial in $\overline{A}$. It is *legal* when it satisfies precisely the same conditions as given in Definition 2.7 (iii).

A.9. **Proposition.** (projection convention) If $s^{A \multimap B}$ is legal, $s \restriction A$ (resp. $s \restriction B$) is always a legal action sequence in A (resp. in B).

Proof: Let s be a legal action sequence from A to B. In the board presentation of s:

A	OPO...O		PO...
B		PO...O	

we call consecutive subsequence solely consisting of A (resp. B) actions, *A-block* (resp. *B-block*) (which already appeared in A.5). Then as the above picture shows, s consists of alternating maximal A-blocks and maximal B-blocks. But the switching condition tells us, as shown above, each maximal block, except the initial one, starts with a P-action and ends with an O-action. Thus when we project s onto A, it is surely strictly alternating. Since in the projection justification is preserved (notice there is no justification between different components), we now know it is an action sequence. Because if it is not well-bracketed then so is s, it should also be well-bracketed. Finally, for the visibility condition, take any O-action from A, then by the O-view projection we know it is justified from within its view even when projected onto A. For a P-action from A, the P-view projection (Lemma A.4) says that the P-view taken in A-projection covers more A-actions than that taken in the original s, hence as required. □

We next prove the basic properties of innocent strategies.

A.10. **Proof of Proposition 2.9.**

(i) (characterisation by innocent function) $\sigma \subset \tau$ implies $f_\sigma \subset f_\tau$ by definition. In another direction, suppose $f_\sigma \subset f_\tau$. We show by induction all action sequences in σ are also in τ. The base case is trivial. Suppose $s' \stackrel{\text{def}}{=} sx \in \sigma$, so by induction $s \in \tau$. If x is an O-action, $s' \in \tau$ follows from contingency completeness. If on the other hand x is a P-action, then $f_\sigma(s) = s'$, hence $f_\tau(s) = s'$, hence $s' \in \tau$, as required.

(ii) (the set of strategies as dI-domain) We use (i) and argue just like we deal with the set of partial functions on say ω. We only show the basic data. The situation is exactly the same as described by Hyland-Ong [27].

(lub): Given directed (or bounded) $\{\sigma_i\}$, just take $\cup_i f_{\sigma_i}$. (This is because of the following fact: a partial function from the odd-length P-views to the even-length P-views, for each case incrementing the original P-view by an additional action, is an innocent function of some strategy if and only if $f(s) = s'$ implies, for each odd-length prefix, say s'_0, of s', $f(s'_0)$ is defined and is also a prefix of s'. Notice this trivially holds in the union of a directed (or bounded) set of strategies.)

(meet): Set-theoretic meet of the original innocent functions (defined only when compatible).

(compatibility): Compatibility as sets of sequences, or as innocent functions.

(finite element): A strategy whose innocent function is of finite cardinality.

(prime element): A finite element which has only one defined P-action at each stage: equivalently a strategy σ with the maximum even-length sequence, of which other even-length sequences are prefixes. $\square$

We prove Lemma 2.11 which makes the notion of the composition of strategies well-defined.

A.11. **Proof of Lemma 2.11.** Assuming $s_1^{A\to B}$ and $s_2^{B\to C}$, we argue by induction of $n = |s_1| + |s_2| - |s_1 \upharpoonright B|$ (remember $|s|$ denotes the length of s as an ordinary sequence). In the base case, $n = 0$, then (1) trivially holds. Suppose it holds up to $n = m$ and consider $n = m + 1$. If $s_1 = \varepsilon$ then $s_2 = \varepsilon$ so that we can set $s_1 = s'_1 x$. We do case analysis on x.

(x is from A). Then by assumption $s'_1 \asymp s_2$. If (1) holds for s'_1 and s_2, again (1) holds with s_1 and s_2. If (2) holds for s'_1 and s_2, then (3) holds for s_1 and s_2. If (3) holds for s'_1 and s_2, then, by contradiction, suppose s_2 ends with a C-action. Then (since (3) holds) s'_1 ends with a B-action. Take the minimum prefix of s_2, say s'_2, such that $s_1 \upharpoonright B = s'_1 \upharpoonright B = s'_2 \upharpoonright B$. The eliminated postfix consists only of C-actions (because, if else, we have $s_1 \upharpoonright B \neq s_2 \upharpoonright B$). By switching condition the last B-action of s'_2 is by Opponent. Therefore, the action compensating x, which is the last action of s'_1, is by Player, hence $s'_1 x$ cannot be legal, again by switching condition. Therefore s_2 ends with a B-action, showing (3) does hold.

(x is from B). Then $s_2 \neq \varepsilon$, so that let $s_2 = s'_2 y$. If y is from B we have (2), while if y is from C we have (3). $\square$

From now on we often use the board presentation of composition of action sequences, which is again formalised as a partial matrix (cf.A.6).

A.12. **Definition.** (board presentation of composition)[4] Let $s_1^{A\to B}$ and $s_2^{B\to C}$ are composable. We define a sequence of (i) the elements of sorts of the prearena $\overline{A} \uplus C$, and (ii) the pairs of elements of sorts of the pre-arena $\overline{B} \uplus B$, in the following way: (1) $\varepsilon : \varepsilon = \varepsilon$, (2) $s_1 x^B : s_2 y^{\overline{B}} = (s_1 : s_2)\langle x^B, y^{\overline{B}}\rangle$, and (3) $s_1 x^A : s_2 = (s_1 : s_2)x^A$ as well as $s_1 : s_2 x^C = (s_1 : s_2)x^C$. We now define the *game board presentation of composition of* s_1 *and* s_2, as a partial $4 \times \omega$ matrix whose all defined coefficient are given as:

(I) If $j-1$ element of $s_1 : s_2$ is x from A (resp. C), then the coefficient at the first (resp. fourth) row, i-th column has the value $x \upharpoonright A$ (resp. $x \upharpoonright C$).

(II) If $j-1$ element of $s_1 : s_2$ is $\langle x, y\rangle$, then the coefficient at the second (resp. third) row, i-th column, has the value $x \upharpoonright B$ (resp. $y \upharpoonright B$).

together with the justification relation over the (defined) coefficients corresponding to the original justification relations. We often call such a presentation as a *game board for composition.* We again call the rows of the matrix by the associated arenas, sometimes writing e.g. $\overline{A}$ to make its presentation in the pre-arena explicit.

We give an example of such a presentation.

$\overline{\mathsf{nat} \Rightarrow \mathsf{nat}}$	$*$	3	n		
nat				$n+1$	
$\overline{\mathsf{nat}}$				$n+1$	
nat					$2n+2$

[4] The notion corresponds to *uncovering* in [27].

In the following proofs, we often use labels like O, P, ... as actions in the board presentation. We now prove Proposition 2.14. We first show:

A.13. **Lemma.** If $s_1^{A \to B}$ and $s_2^{B \to C}$ are composable legal action sequences, then $s_1; s_2$ is always a legal action sequence from A to C.

Proof: We use the game board presentation. Notice $s_1; s_2$ is nothing but the projection of the board presentation onto the A-C components. We first show $s_1; s_2$ is an action sequence from A to C satisfying the switching condition (i.e. it is always Player who switches between A and B), then shows it is legal.

(i) ($s_1; s_2$ is an action sequence satisfying the switching condition.) Since the justification on A-actions (resp. C-actions) is precisely as in s_1 (resp. s_2), the only issue is strict alternation. By Lemma 2.11, the board starts with a $\overline{A}$-block, which might be followed by a B-$\overline{B}$ block, then either a $\overline{A}$ or C block, which again might be followed by a B-$\overline{B}$ block, and again $\overline{A}$ or C block, etc., that is, there is no possibility that a A block and a C block is contiguous (violating (3) of Lemma 2.11). Moreover each new $\overline{A}$ or C block after a B-$\overline{B}$ block always starts with a P-action, and each $\overline{A}$ or C block before B-$\overline{B}$. This establishes both the strict alternation and the switching condition (the situation is depicted below).

$\overline{A}$	OPO...O				PO..
B		PO...P		OP...O	
$\overline{B}$		OP...O		PO...P	
C			PO...O		

(ii) ($s_1; s_2$ is legal) We establish the bracketing and the visibility independently.

1. bracketing. For bracketing, we notice $(s_1; s_2) \upharpoonright A = s_1 \upharpoonright A$ and $(s_1; s_2) \upharpoonright C = s_2 \upharpoonright C$. Thus the only possible violation of the condition is when $s_1; s_2 = s_0 x_i s y_j s' x_k s''$ where x_i, x_j are from A (resp. C) and y_j is from C (resp. A) and moreover x_k answers x_i and y_j is left unanswered in s'. We draw the situation as a game board:

$\overline{A}$	...OPO...PO				P...O	...
B	...	PO...P		OP...O		...
$\overline{B}$	...	OP...O		PO...P		...
C	...		PO...O			...
	(a)	(b)	(c)	(d)	(e)	

with x_i in (a), y_j in (c), and x_k in (e). Note that the length of (d) is odd by switching condition. But in (d), if the number of questions is larger than answers, then x_k answers some question in B in (d). Contrarily if the number of answers is larger than questions, then y_j is answered by some answer in $\overline{B}$ in (d), hence both cases contradict the assumption that s_1 and s_2 are well-bracketed.

2. visibility. Suppose $s_1; s_2 = sx$. We consider the visibility of x in its view. If x is free, there is nothing to prove, so suppose not. There are two cases.

(x is an O-action). Then $s = s'y$ with y a P-action. We first note y and x should be in the same component, by the switching condition. The justifying action of y in s' is again in the same component and is an O-action, so we just continue this until arriving at the initial free P-action, but this is the same thing as taking the O-view in the original s_i (if x, y come from s_i), hence by the visibility in s_i we are done.

(x is a P-action). Let x be originally in, say, s_1. If $\ulcorner s_1 \urcorner$ just touches the A-component, then this is the same thing as taking $\ulcorner sx \urcorner$, hence we are done. If however this is not the case, then, when tracing back in the view construction (in both $\ulcorner s_1 \urcorner$ and $\ulcorner sx \urcorner$), we come upon some P-action, say y, which is at the "brim" of consecutive actions, like:

$\overline{A}$	...		P $(= y)$.....
B	...	OP...O	...
$\overline{B}$	...	PO...P	...
C	...PO...O $(= z)$		...

Notice, in the view construction of $\ulcorner s \urcorner$, when one comes upon such x, then we directly go back to z rather than going through the intermediate B actions (which do not exist in s). Call a maximal B-$\overline{B}$ block which immediately precedes a non-free P-action that is touched in $\ulcorner s \urcorner$, a *s-bridge*, or simply a *bridge*. We now establish the following claim.

Claim. Any action in a bridge is either free or is justified by an action in a (same or different) bridge.

Notice, if the claim holds, we know that all A-actions in $\ulcorner s_1 \urcorner$ are already in $\ulcorner s \urcorner$, so that the justifier of x in $\ulcorner s_1 \urcorner$ still occurs in $\ulcorner s \urcorner$ (by the visibility condition in s_1). The claim is proved by induction on n where n is the number of bridges the claim holds counting from the left-most one.

Case $n = 1$. In the left-most bridge, the first B-action should be free, since if not $\ulcorner s \urcorner$ will go back through another bridge on the left, which contradicts that we are now in the left-most bridge. For the second action in the bridge, by the visibility in s_1 (or s_2), we immediately know it is (free or) justified by the preceding B-action. In this way it is clear all B-actions in this left-most bridge are justified (if ever) within the bridge.

Case $n = m + 1$. We assume that the property holds for actions in up to the m-th bridge. Now go to the next bridge. In the first pair of B-actions, it is Player who has switched: and that P-action, say y, is justified in s_i (if ever) by the P-view going backward from the action which immediately precedes y (the position of z in the above diagram). But this is the same thing as taking the P-view in s, so it can only reach the previous brim and then back to the previous bridge, thus ending finally in some free action in the bridge, as expected. Now if we take the P-view or O-view for the second action we start from the first action so at most we can only reach the left-most bridge, thus as required. □

A.14. **Proof of Proposition 2.14.** Assume given $\sigma : A \rightarrow B$ and $\tau : B \rightarrow C$.

(i) ($\sigma;\tau$ is innocent) We already know $s \in \sigma;\tau$ is a legal sequence from A to C. That $\sigma;\tau$ is prefix-closed is because, for example, if $s_1 x^A; s_2 \in \sigma;\tau$ then $s_1; s_2 \in \sigma;\tau$ similarly for $s_1; s_2 x^C \in \sigma;\tau$ (note we do not have to consider $s_1 x^B; s_2 \overline{x^B}$). Contingency completeness is immediate from the O-view projection. Thus we have only to show $\sigma;\tau$ is innocent. Given a composed sequence:

$\overline{A}$	OPOPO		PO		
B		POPO		POP	
$\overline{B}$		OPOP		OPO	
C					PO...

Let s denote it. Then write $s \upharpoonright A, B$ for taking the first two rows from s, considered as an action sequence, similarly write $s \upharpoonright A, C$ and $s \upharpoonright B, C$ for taking the first and forth rows and the third and fourth rows, respectively. We now prove, as done by Hyland and Ong [28], the following properties: given two such extended sequences, say s_1 and s_2,

(i) If both end with O-actions at A, then:

$$\ulcorner s_1 \upharpoonright A, C \urcorner = \ulcorner s_2 \upharpoonright A, C \urcorner \quad \Rightarrow \quad \ulcorner s_1 \upharpoonright A, B \urcorner = \ulcorner s_2 \upharpoonright A, B \urcorner$$

(ii) Similarly if both end with O-actions at C, then:

$$\ulcorner s_1 \upharpoonright A, C \urcorner = \ulcorner s_2 \upharpoonright A, C \urcorner \quad \Rightarrow \quad \ulcorner s_1 \upharpoonright B, C \urcorner = \ulcorner s_2 \upharpoonright B, C \urcorner.$$

We argue by induction of the length of a longer sequence (let it be s_1). For the base case the above holds immediately. So suppose it holds for sequences of length no more than k. There are a few cases according to how s_1 ends. We only show the following key case. We assume s_1 ends in the way:

$\overline{A}$	...... PO		PO
B	......	POPO.....PO	
$\overline{B}$	......	OPOP.....OP	
C	......		

We may write $s_1 = s_1' \tilde{w}^B xy$ and $s_2 = s_2' \tilde{z}^B xy$ where $\tilde{w}^B$ and $\tilde{z}^B$ denote the consecutive interactions at B, and x is a P-action and y is an O-action (at this moment $\tilde{z}^B$ may be empty, though we shall soon know it is not). Now write $s_1 \backslash B$ for $s_1 \upharpoonright A, C$. Since $\ulcorner s_1 \backslash B \urcorner = \ulcorner s_2 \backslash B \urcorner$, we know $\ulcorner s_1' \backslash B \urcorner = \ulcorner s_2' \backslash B \urcorner$. By induction we know $\ulcorner s_1' \upharpoonright A, B \urcorner = \ulcorner s_2' \upharpoonright A, B \urcorner$. Therefore, the action in s_2 after doing s_2' should be the same P-action in B as that of s_1 after s_1'. Then (like a zipper) this decides what would be done in the lower B component by σ_2, and in this way precisely the same sequence of B-actions will be done in s_2 as in s_1, leading to the switching into A and the final two actions, thus we know $\ulcorner s_1 \upharpoonright A, B \urcorner = \ulcorner s_2 \upharpoonright A, B \urcorner$, as required. Other cases can be similarly verified. Once (i) and (ii) above are proved, suppose, after s in the composite $\sigma_1; \sigma_2$, an O-action is done at, say, A. Then σ_1's next action is determined by the corresponding P-view, thus if its next action is at A this is the action of the composite. On the other hand if the next action of σ_1 is at B then the consecutive actions at B and the visible action at A or C after them, are uniquely determined, using the "zipper" argument as above, hence the action at the composite strategy is again uniquely determined by the P-views, hence as required.

(iii) (associativity) We show $s \in (\sigma_1; \sigma_2); \sigma_3$ iff $s \in \sigma_1; (\sigma_2; \sigma_3)$ with $\sigma_1 : A_i \rightarrow A_{i+1}$. But this is the same thing as showing $(s_1; s_2); s_3 = s_1; (s_2; s_3)$ with $s_i \in \sigma_i$ when s_1, s_2, s_3 are pairwise composable, because if not we can always reduce them to composable ones. The case $s_i = \varepsilon$ for some $i = 1, 2, 3$ is obvious. Suppose not, then there are four cases in which both $(s_1; s_2); s_3$ and $s_1; (s_2; s_3)$ are defined. First, we have:

$$\begin{aligned} (s_1' x^{A_1}; s_2' y^{A_2}); s_3' z^{A_3} &= ((s_1'; s_2' y^{A_2}) x^{A_1}); s_3' z^{A_3} \\ &= ((s_1'; s_2' y^{A_2}); s_3' z^{A_3}) x^{A_1}, \end{aligned}$$

while

$$\begin{aligned} s_1' x^{A_1}; (s_2' y^{A_2}; s_3' z^{A_3}) &= s_1' x^{A_1}; ((s_2'; s_3' z^{A_3}) y^{A_2}) \\ &= (s_1'; ((s_2'; s_3' z^{A_3}) y^{A_2})) x^{A_1} \\ &= (s_1'; (s_2' y^{A_2}; s_3' z^{A_3})) x^{A_1}. \end{aligned}$$

But by inductive hypothesis we have $(s_1'; s_2' y^{A_2}); s_3' z^{A_3} = s_1'; (s_2' y^{A_2}; s_3' z^{A_3})$. Secondly,

$$(s_1' x^{A_2}; s_2' \overline{x^{A_2}}); s_3' z^{A_3} = (s_1'; s_2'); s_3' z^{A_3},$$

while

$$\begin{aligned} s_1' x^{A_2}; (s_2' \overline{x^{A_2}}; s_3' z^{A_3}) &= s_1' x^{A_2}; (s_2'; s_3' z^{A_3}) \overline{x^{A_2}} \\ &= s_1'; (s_2'; s_3' z^{A_3}), \end{aligned}$$

then we can again use inductive hypothesis. Other two cases are similar.

(iv) (continuity) We conclude the proof of Proposition 2.12 by showing ; is a -continuous operator. It is clearly monotone by construction. Thus $(\sqcup \sigma_i); \tau \sqsupseteq \sqcup(\sigma_i; \tau)$. But because any $s \in \sqcup \sigma_i$ is in some σ_i, $(\sqcup \sigma_i); \tau \sqsubseteq \sqcup(\sigma_i; \tau)$. We can similarly reason for the other case, so that we now know ; is a well-defined bi-continuous operator on strategies. □

We next give the proof for recursion.

A.15. **Proof of Proposition 3.11.** Below assume A is pointed, $\sigma : C \otimes A \rightarrow A$ and $\tau : B \rightarrow C$.

(i) Direct from the construction of $p\lambda(\sigma)$. In detail: In $\mathbf{rec}(\tau \otimes \mathsf{id}_A ; \sigma) \stackrel{\text{def}}{=} p\lambda(\tau \otimes \mathsf{id}_A ; \sigma); \mathsf{fix} : B \rightarrow A \rightleftharpoons A \rightarrow A$, $p\lambda(\tau \otimes \mathsf{id}_A ; \sigma)$ first gets an initial signal at B, to which it reacts by going to the initial Player signal at $A \rightleftharpoons A$ (invisible), then fix reacts by asking at $\overline{A}$ of $A \rightleftharpoons A$ (again invisible), so that now τ can interact at B until it reaches C, at which point σ can start interaction. In $\tau; \mathbf{rec}(\sigma) \stackrel{\text{def}}{=} \tau; p\lambda(\sigma); \mathsf{fix}$, given the same signal at B, τ would interact just as above, and, when it reaches C, first $p\lambda(\sigma)$ goes to $A \rightleftharpoons A$ to which fix reacts by asking at $\overline{A}$, so that σ can now start interaction. The rest of the behaviour is precisely simulated between these two, modulo the copy-cat of the identity which we can neglect, hence done.

(ii) We give behavioural description of two strategies which shows how these two strategies simulate each others' behaviour. Underlying is the induction on the length of expanded composition sequences where we inductively construct the simulation relation between two sequences (by expanded composition we mean the composition as in 2.12 except we do not eliminate the interacting actions, see Appendix A.12 for the definition). The behavioural reasoning below is hopefully clear enough to indicate the underlying formal construction while being much more concise. We articulate the behaviour in three stages. We assume τ is total, since, if not, there is nothing to prove (both simply become $\perp$). Now both strategies start from the unique signal at $\mathbf{1}$, and we observe no further action is possible at $\mathbf{1}$, so we can forget this arena in the following. Below as elsewhere we distinguish two A in $A \rightleftharpoons A$ as $\overline{A}$ (for the first copy) and A (for the second one).

(Initial stage)

- In $\tau; \mathbf{rec}(\sigma) \stackrel{\text{def}}{=} \tau; p\lambda(\sigma); \mathsf{fix} : \mathbf{1} \rightarrow C \rightarrow A \rightleftharpoons A \rightarrow A$. First τ reacts to the initial signal by going to C, to which $p\lambda(\sigma)$ reacts by reaching $A \rightleftharpoons A$ immediately. fix then reacts by asking at $\overline{A}$, to which $p\lambda(\sigma)$ would react in one of the two ways, according to the underlying behaviour of σ. (1) Possibly through interaction at C, it answers at A without interacting at $\overline{A}$. Or (2) it either deadlocks or asks something at $\overline{A}$. In (2) the whole behaviour deadlocks. In (1) fix now answers at A in the co-domain.
- In $\tau; \langle\!\langle \mathsf{id}_C, \mathbf{rec}(\sigma)\rangle\!\rangle; \sigma \stackrel{\text{def}}{=} \tau; \langle\!\langle \mathsf{id}_C, p\lambda(\sigma); \mathsf{fix}\rangle\!\rangle; \sigma : \mathbf{1} \rightarrow C \rightarrow C \otimes A \rightarrow A$. Initially τ reacts and goes to C. Then since both id_C and $\mathrm{P}\sigma$ are total, now σ gets the initial signal. We now use the classification of σ's behaviour given above. First, if (2) is the case, clearly the whole behaviour deadlocks, just as above. Second, if (1) is the case, then σ would eventually answer at A, again precisely as above (notice that, because A is pointed, only one kind of reaction is possible). Thus these two behaviours (up to this stage) simulate each other.

(Second stage)

- In $\tau; p\lambda(\sigma); \mathsf{fix} : \mathbf{1} \rightarrow C \rightarrow A \rightleftharpoons A \rightarrow A$. fix now relates its co-domain A and the A of its domain $A \rightarrow A$ by copy-cat, prefixed by the two view clearing actions in the co-domain, so that, because for $p\lambda(\sigma)$ the P-view does not concern the repetition of the view clearing actions, the situation is precisely the same as that of $\sigma : C \otimes A \rightarrow A$ interacting at A of its domain, as far as it does not interact at $\overline{A}$ of its domain. Now at any further point it does not do so, this is the last stage. If it does, however, we move to the third stage.
- In $\tau; \langle\!\langle \mathsf{id}_C, p\lambda(\sigma); \mathsf{fix}\rangle\!\rangle; \sigma : \mathbf{1} \rightarrow C \rightarrow C \otimes A \rightarrow A$. After the interaction at A in the co-domain by σ, if it has any interactions at C then they are mediated by id_C to τ, so precisely the same behaviours as above continue. If σ does not ask at A of its domain, the simulation is exact, up to the emulation by id_C here and fix there which is negligible, by the same reasoning we gave in 2.13. If however σ does so, it is fed to

$p\lambda(\sigma);\mathsf{fix}$ and we move to the third stage.

(Third stage)

- In $\tau; p\lambda(\sigma); \mathsf{fix} : \mathbf{1} \to C \to A \rightleftharpoons A \to A$. fix now in effect relates $p\lambda(\sigma)$'s action at $\overline{A}$ to A (of fix's domain) by copy-cat. Notice now $p\lambda(\sigma)$ interacts with itself. Notice however $p\lambda(\sigma)$ who is questioned acts in the fresh view which starts from the second level sort of A, which is the basis of its behaviour, due to innocence. Thus there is an interaction between the second copy of $p\lambda(\sigma)$ and the first copy. Later the first $p\lambda(\sigma)$ may ask a fresh question at $\overline{A}$, then there is another interaction sequence between the first and the third copy (these copies themselves would invoke other copies).
- In $\tau; \langle\!\langle \mathsf{id}_C,\, p\lambda(\sigma);\mathsf{fix}\rangle\!\rangle; \sigma : \mathbf{1} \to C \to C \otimes A \to A$. The question is done at A of σ's domain, which is forwarded by fix to $p\lambda(\sigma)$, precisely with the same view as the "second copy" above. Thus interactions between σ and $p\lambda(\sigma)$ precisely proceed following the above case: for example, if σ further asks a fresh question at A of its domain, the first copy of $p\lambda(\sigma)$ does precisely the same thing at $\overline{A}$ of its co-domain, and the invoked copy reacts with the same fresh view. In this way two histories of interactions precisely correspond to each other, up to the copy-cat of id_C and fix. In particular visible behaviours at the co-domain A of the whole strategy coincide.

Notice the innocence of σ is essential in the reasoning above. The stronger statement for the case when σ is total is verified similarly.

(iii) We again assume τ is total. It is immediate $\{\rho_i\}$ is an increasing ω-chain. Notice also if $\rho_1 = \bot$, $\rho_i = \bot$ for all i, while, if not, $\rho_{i+1} = \langle\!\langle \tau,\, \rho_i\rangle\!\rangle; \sigma$ for all $i \geq 1$. We first unfold the recursion, using (ii) above, as follows:

$$\begin{aligned} \tau; \mathbf{rec}(\sigma) &= \tau; \langle\!\langle \mathsf{id}_C,\, \mathbf{rec}(\sigma)\rangle\!\rangle; \sigma \\ &= \langle\!\langle \tau,\, \tau; \mathbf{rec}(\sigma)\rangle\!\rangle; \sigma \end{aligned} \tag{A.2}$$

(the final equation is because τ is total, cf. Proposition 3.9 (iv)). We now show $\rho_i \sqsubseteq \tau; \mathbf{rec}(\sigma)$ for $i \geq 1$ by induction.

(Base Case.) $\rho_1 \stackrel{\text{def}}{=} \langle\!\langle \tau,\, \bot^\dagger; \mathsf{dn}'\rangle\!\rangle$. After the initial signal at $\mathbf{1}$, $\langle\!\langle \tau,\, \bot^\dagger; \mathsf{dn}'\rangle\!\rangle$ immediately reacts at $A \otimes C$. At this point σ would interact, and (1) it would reach at A without asking at A in the domain again, or (2) it would do otherwise, thus it deadlocks. Now in the case of (1) we already know, by the proof (ii) above, that $p\lambda(\sigma)$ deadlocks, so the behaviour coincides. In the case of (2), then there is a signal at $C \otimes A$, hence σ now reacts precisely as σ in ρ_i. However after visiting A in the co-domain, σ in ρ_i would ask at A in the domain, at which point the behaviour deadlocks, while this may not be the case in $\tau; \mathbf{rec}(\sigma)$. Thus we know $\rho_1 \sqsubseteq \tau; \mathbf{rec}(\sigma)$.

(Induction.) If $\rho_i = \bot$ for $i \geq 1$, there is nothing to prove, so suppose not. Then, using (A.2), we have $\rho_{i+1} = \langle\!\langle \tau,\, \rho_i\rangle\!\rangle; \sigma \sqsubseteq \rho_{i+1} \sqsubseteq \langle\!\langle \tau,\, \tau; \mathbf{rec}(\sigma)\rangle\!\rangle; \sigma$ hence done.

To establish $\sqcup\rho_i = \tau; \mathbf{rec}(\sigma)$, we reason by the length of composed action sequences (including invisible moves). Again using (A.2), we can unfold $\tau; \mathbf{rec}(\sigma)$ as many times as we like. In particular we have exactly the same configuration as ρ_i by unfolding $\mathbf{rec}(\sigma)$ by i-times, except we have $\bot^\dagger; \mathsf{dn}'$ at the first stage. This means, until a question reaches $\bot^\dagger; \mathsf{dn}'$, exactly the same behaviour is done between two strategies. Thus, for each interaction sequence with a certain length, there is always some ρ_i which does the same, which shows in particular each action sequence (i.e. visible behaviour) in $\tau; \mathbf{rec}(\sigma)$ is contained in some ρ_i, hence done. $\square$

We also list the proof of further switching conditions for legal action sequences whose co-domains are arenas of specific kinds (tensors and exponentials). These properties are used in the proof of Proposition 3.7. They correspond to similar properties of call-by-name games, see [27, 34].

A.16. **Proposition.**

(i) (switching condition for tensor) Let s be a legal action sequence from C to $A \otimes B$. Let $s' \stackrel{\text{def}}{=} s \upharpoonright A \otimes B$. Then if one action in s' is from the A-component (resp. from the B-component) and the succeeding one is from the B-component (resp. from the A-component), then the latter is always an O-action.

(ii) (switching condition for partial exponential) Let s be a legal action sequence from C to $A \rightharpoonup B$. Let $s' \stackrel{\text{def}}{=} s \upharpoonright A \rightharpoonup B$. Then if one action in s' is from the $\overline{A}$-component (resp. from the B-component) and the succeeding one is from the B-component (resp. from a non-initial sort of the $\overline{A}$-component), then the latter is always a P-action.

Proof: For (i), the situation of a legal sequence from C to $A \otimes B$ may be displayed like:

$\overline{C}$	OP..O			PO...O			P...
$A \otimes B$		P					
A			OP..O		PO...P		
B						OP...O	

where the row $A \otimes B$ only records a unique free P-signal of $A \otimes B$ if any (notice we decompose the second row of the game board in the sense of A.6 into its three components). Now let us say (an action in) a $\overline{C}$-block *belongs to A (resp. B)*, if the block is immediately after an A-block (resp. a B-block). We now prove the specified switching condition for tensors together with: (1) if we take the P-view from an O-action in A (resp. B), then it only touches actions in C, A (resp. B) and $A \otimes B$, (2) if we take the P/O-view from an O/P-action in C which belongs to A (resp. B), then it only touches actions in C, A (resp. B) and $A \otimes B$, and (3) if there is a maximal $\overline{C}$-block belonging to A (resp. B) then the immediately succeeding P-action (if any) should be an action in A (resp. B). The proof is by induction on the number of non-initial C-blocks in the whole sequence, in each case by induction on the number of actions after the initial action of the last C-block. For the base case, first (1) holds for the O-action immediately following a P-signal, while the induction for (1) as well as the switching condition is immediate by inductive hypothesis for the preceding actions. If there is a non-initial $\overline{C}$-block which belongs to, say, A, then its first action satisfies (2) by inductive hypothesis for the preceding action (which is in A), from which (2) follows as far as we are in the $\overline{C}$-block. When we switch out of $\overline{C}$, by induction on (2) we can only go to A, so (3) now holds. Then inductively (1) and the switching condition hold for later actions by combining the inductive hypotheses on (1), (2) and (3). (ii) is proved similarly. □

A.17. **Discussions.** We conclude this Appendix with discussions on the strategies which constitute the bottom map in $\widehat{\mathcal{CBV}}$. While a natural idea would be that $\sigma \in \perp_{A \rightharpoonup B}$ implies σ being insensitive, this is actually *not* the case, as was already mentioned after Proposition 4.2. The following gives a counterexample.

Fact. Let $\sigma : \mathbf{1} \rightharpoonup \mathsf{nat} \rightarrow \mathbf{1}$ be the following strategy: after the unique initial signal at the domain, it asks at $\mathbf{1}$ in the domain, and iff the Opponent answers by 1, it again questions at $\mathbf{1}$ in the domain (else it has no action), and iff the Opponent answers by 2, it answers by the unique initial signal at $\mathbf{1}$ of the co-domain (else it has no action), then no more action is possible. Then $\sigma \lessapprox \perp$.

Proof: Notice no innocent strategy $\tau : \mathbf{1} \rightarrow \mathbf{1} \rightharpoonup \mathsf{nat}$ would answer differently for the two questions at $\mathbf{1}$ in the co-domain, since the P-view for τ is the same. So $\tau; \sigma = \perp$ always, that is, by the last statement of Proposition 4.2 (ii), $\tau \lessapprox \perp$, as required. □

We note that we can avoid this situation by changing the notion of strategy in the following way: in Definition 2.9, we change the *contingency completeness* by:

(contingency completeness) If $s \in \sigma$ is empty or it ends with a P-action, then if $s' \stackrel{\mathrm{def}}{=} sx$ is legal and, moreover, *for each odd-length prefix* $s'y$ *of* s, $\llcorner s' \lrcorner = \llcorner s \lrcorner$ *implies* $\llcorner s'y \lrcorner = \llcorner sx \lrcorner$, then $sx \in \sigma$.

The extra condition, written in italic above, says that a strategy does not expect the possibility that Opponent is not innocent. The resulting universe has exactly the same algebraic properties, including the intensional full abstraction (indeed the extensional universe is isomorphic), except that such strategies as σ in the above Fact disappear, i.e. we now have $\sigma \in \perp$ in $\widehat{\mathcal{CBV}}$ iff σ is insensitive. In many senses this is a "cleaner" universe, but we preferred the present formulation due to its simplicity and, in particular, conformance to the original definition by Hyland and Ong. One may also observe that, if we (further) identify all insensitive strategies, $\mathbf{0}$ now becomes (intensionally) the zero object: it is notable that in this setting the intensional universe becomes algebraically compact in the sense of [21] with respect to CPO functors, even though such revision may not be natural from an operational viewpoint.

Appendix B. Process Representation of strategies

As we observed in Section 2 (before Lemma 2.11), strategies and composition in $\mathcal{CBV}$ can be precisely captured by their process representation together with the standard notion of process composition. We also note that the present work itself has arisen from our study on the relationship between strategies in game semantics and name passing processes, cf. [24]. Among others we can use the process representation as a succinct and clear way to describe strategies and their composition. In the following we give a brief review of such a representation, which is then used to describe and reason about conspicuous strategies which appear in the main sections. The reasoning can be considered as a way of describing composition of actions in a concise and precise way. The full discussions on the process representation of games and its applications will be found elsewhere: we only discuss basic technical constructions needed for our present purpose.

The basic idea of process-based description of strategies is to consider each prearena as a sorting in the sense of [38], assign a π-calculus to each action in an action sequence (with justification represented by binding), so that, in effect, a strategy becomes a name passing synchronisation tree or, essentially equivalently, a term in the π-calculus. There are a few ways of formulating the idea. In the following we use what may be most concise to present.

In this Appendix, we write $S, S', \ldots$ to denote sorts to clearly distinguish sorts in prearenas and names as used in the labels of π-calculus.

B.1. **Names, sorting and labels.**

(i) Let $\mathcal{N}$ denote a proper class of *names*, ranged over by $a, b, c, \ldots$. Fix two arenas, say A and B. Then its *$\mathcal{CBV}$-sorting* $\mathbb{S}$, or simply its *sorting* from now on, is an assignment of mutually disjoint sets of names to the sorts of the pre-arena $\overline{A} \uplus B$ such that (1) a countable set of names is assigned to each non-initial sort, and (2) a singleton name is assigned to each initial sort. We write $a : S$ when a is assigned to a sort S.

(ii) Fix a sorting. Then the set of *action labels*, or often simply *labels*, ranged over by $l, l', \ldots$, are those data of form:
   - $b_0(\{b_S\}_{S \in I})$ (input) and
   - $\overline{b}_0(\{b_S\}_{S \in I})$ (output),

   where if $b_0 : S_0$ then $I = \{S | S_0 \mapsto S\}$, $b_S : S$ for each S, and $b_S \neq b_{S'}$ if $S \neq S'$. Labels are collectively written $b_0^{\pm}(\{b_S\})$. In $l = b_0^{\pm}(\{b_S\})$ and $\mathsf{sub}(l) = b_0$. A label sequence $l_0 \ldots l_n$ has a binding relation where names in $(\cdot)$ act as binders. If an object of l_i binds the subject of another label l_j, we say l_i binds l_j by abuse of terminology. We only deal with a label sequences whose binders are all distinct. The α-equality, denoted $\equiv_\alpha$, between such sequences is defined in the standard way.

Notice we extend the usual labels in the π-calculus so that they carry an arbitrary family of names rather than a sequence of names. Thus, for example, an action sequence from nat to nat is represented by $a_2.\overline{b}_3$, by which we denote a reaction of a successor function to the input 2. Following the notation of CCS [35], we sometimes write such a transition as e.g. $a\langle 2\rangle.\overline{b}\langle 3\rangle$ in concrete examples.

Below we remember we are using $S, S', \ldots$ for sorts, instead of $x, x', \ldots$.

B.2. **Definition and Proposition.** Fix $\mathbb{S}$. A label sequence $l_0...l_n$ *represents* an action sequence $S_0...S_n$ from A to B when, for each $0 \leq i \leq n$: (i) $\mathtt{sub}(l) : S_i$, (ii) S_i is O-action (resp. P-action) iff l_i is input (resp. output). (iii) If S_i is initial, $\mathtt{sub}(S_i)$ is free in the label sequence; if not, then if $S_j \curvearrowright S_i$ then l_j binds l_i. Then for each action sequence s, the label sequence which represents s is determined uniquely up to α-equality.

Remark. In the terminology of [11, 51], we are using the *object dependency* to represent the justification relation.

The following conventions are indispensable for clarity of expressions.

B.3. **Convention.**

(i) Hereafter we identity two α-convertible label sequences when discussing the representation.

(ii) To make the presentation simple, we hereafter assume there is a linear order on the siblings of each sort in an arena, so that we can write, say, $a(b_1b_2)$ instead of $a(\{b_S\})$ (this is used only when we give concrete examples). As here, we often assume the number of siblings of a sort is finite. If in particular the name sequence is null, we often omit it.

We give a simple example of the representation of action sequences below.

B.4. **Example.** Consider a sorting for $(\mathsf{nat} \rightleftharpoons \mathsf{nat}) \to \mathsf{nat}$, in which we assign a to the initial sort of $\mathsf{nat} \rightleftharpoons \mathsf{nat}$ and b_i $(i \in \omega)$ to the initial sorts of nat. then

$$a(a').\overline{a}'_3(a'').a''_6.\overline{b}_7,$$

which is alpha-convertible to (hence identified with)

$$a(c).\overline{c}_3(c').c'_6.\overline{b}_7,$$

represents an action sequence

$$S_0S_1S_2S_3$$

where $S_0 \curvearrowright S_1$ and $S_1 \curvearrowright S_2$, such that $S_1 = 3$, $S_2 = 6'$, and $S_3 = 7''$, where we are distinguishing elements of three copies of ω by superscripts (this sequence is in the strategy in Example 2.10 (iv)). Notice how binding and justification correspond to each other. As we noted before, we may also write the same label sequence as

$$a(a').\overline{a}'\langle 3\rangle(a'').a''\langle 6\rangle.\overline{b}\langle 7\rangle,$$

which may be notationally more intelligible.

Given the process representation of action sequences, the process representation of a strategy may be given as a prefix closed set of label sequences with precisely the same conditions as in Definition 2.9 (taking $\equiv_\alpha$ as the equality). Just as any prefix closed set of sequences induce a tree, this prefix-closed set can be viewed as a *synchronisation tree* in the sense of [35, 54, 56], by which we can define the composition of strategies in the standard way in process theory. This is the primary presentation we shall use, which we define as follows.

B.5. **Strategies.** Fix a sorting of some prearena from A to B and let $\sigma : A \to B$. Then the *synchronisation tree representation of a strategy* σ is a tree (i.e. an acyclic directed connected simple graph), say P, whose edges are labelled by action labels such that: (1) the set of all label sequences each traced from the root, denoted $|P|$, are precisely the set of all label sequences representing the action sequences in σ modulo α, and (2) for each n, if $l_1..l_n$ $(0 \leq n)$ is a label sequence from the root and l_n goes into a node, say, q, then the set of outgoing edges from q one-one correspond to, and are labelled by, the set $\{l' \mid l_1..l_n l' \in |P|\}$, where label sequences are considered modulo $\equiv_\alpha$.

Identifying alpha-equal label sequences, such a tree is determined uniquely. As the shape of the resulting tree, we observe that, counting the root node as the depth 0, each node of even depth has one and only one outgoing edge, which is always an output, while from each odd-depth node only input actions are done.

We may write down such a synchronisation tree as:

$$\sum_{n\in\omega} a_n(efg).P_n(efg)$$

where $\sum$ denotes the branching (the sum with the root nodes adjoined) and $P_n(efg)$ denote a tree in which names e, f and g occur free. The composition of strategies are then carried out by the standard expansion law of name passing processes [39], using the hiding operator induced by new name passing and with a modification that complementary labels vanish, rather than becoming τ. The last point corresponds to the equation $\tau.P \approx P$, the equation always valid in any weak bisimilarity. The rules can be written, assuming appropriate α-conversion, as follows. Below and hereafter P, Q, R, .. range over subtrees of synchronisation trees representing strategies, which we simply call *processes* from now on.

B.6. **Composition.** We write $\mathsf{bn}(l)$ for the set of names in l which is not its subject.

(i) (hiding) Given a process P and a set of names α, we define $(\nu\alpha)P$ by the following recursion:

$$\begin{aligned}
(\nu\alpha)0 &\stackrel{\text{def}}{=} 0\\
(\nu\alpha_1)(\nu\alpha_2)P &\stackrel{\text{def}}{=} (\nu\alpha_1\cup\alpha_2)P\\
(\nu\alpha)l.P &\stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } \mathsf{fn}(l)\subset\alpha\\ l.(\nu\alpha\backslash\mathsf{bn}(l))P & \text{if else}\end{cases}.\\
(\nu\alpha)\sum l.P &\stackrel{\text{def}}{=} \sum(\nu\alpha)l.P
\end{aligned}$$

(ii) (parallel composition) Given two processes $P \stackrel{\text{def}}{=} \sum_i l_i.P_i$ and $Q \stackrel{\text{def}}{=} \sum_j l'_j.Q_j$, we define:

$$P \mid Q \;=\; \sum_{l_i=\overline{l'_j}}(\nu\,\mathsf{bn}(l_i))(P_i \mid Q_j) \;+\; \sum_i l_i.(P_i \mid Q) \;+\; \sum_j l'_j.(P \mid Q_j)$$

(iii) Finally given $\sigma_1 : A \rightarrow B$ and $\sigma_2 : B \rightarrow C$, we consider the sorting for two concerned pre-arenas such that their name assignments on sorts on B coincide, while those on A and C are disjoint. Let β be the set of names used for the initial sorts of B. Let P_1 and P_2 be the resulting process representations. Then we define:

$$P_1; P_2 \stackrel{\text{def}}{=} (\nu\beta)(P_1 \mid P_2)$$

B.7. **Proposition.** With $\sigma_{1,2}$ and $P_{1,2}$ as above, $P_1;P_2$ represents $\sigma_1;\sigma_2$ under the sorting which is the union of the original two sortings.

This holds essentially because of the switching condition of strategies. We omit the proof. For convenience we use further conventions listed in the following.

B.8. **Conventions.**

(i) (Notation for O-question) Unlike P-actions and answers, O-questions may be repeated arbitrarily many times at the discretion of the environment. To represent this situation textually, we use the following notation.

$$!l_1.l_2.P \qquad \text{or} \qquad !\sum_i P_i$$

where l_1 is an O-question and each P_i starts with an O-question. The notation should be read as denoting a process which is ready to do a marked input action (an O-question) whenever the action becomes legal, in other words when the present interaction is being done below, or at the same level as, that question and it is the turn of Opponent (notice the notion is different from $!P$ in π-calculus). The algebraic properties of such expressions may not be simple to analyse: however in the present paper this never causes a problem.

(ii) (Generic Arenas) For simplicity we often regard each involved type has only one initial sort. For such types A and B, a strategy $\sigma : A \rightarrow B$ needs free names only for its two initial sorts, so that we often write σ^{ab} for such a representation (which is unique up to α-conversion). Thus when we compose σ and τ, we can consider the process composition of, say, σ^{ab} and τ^{bc}, resulting in $(\sigma;\tau)^{ac}$. We extend this notation to the case when the domain or the co-domain is a ground arena, such as nat, writing e.g. $\sigma^{a_i b}$ or even σ^{ab}, where we assume the domain is say nat whose initial sorts are assigned names $\{a_i\}_{i\in\omega}$.

B.9. **Copycat law.** We end this brief introduction by touching one frequently occurring behaviour, the copy-cat strategies, whose formal definition for the case of identity is given in Example 2.10 (v). The behaviour infinitely continues replication of dual actions, but its algebraic law is very simple. The copy-cat behaviours we deal with in the present paper are always essentially an identity, connecting one copy of say A to another copy of A. Since two copies are assigned distinct names, we can represent a copy-cat by specifying two families (here for simplicity sequences) of names, thus writing:

$$\mathbf{cc}(a_0 ... a_n \mapsto b_0 ... b_n).$$

Here it should be the case that $a_i : S_i$ where S_i is an Opponent sort while the corresponding S_i' such that $b_i : S_i'$ is a Player sort (since the copy-cat starts when Opponent does some action, which is then copied to another sort). Then we have the following laws for the copy-cat behaviour:

$$P(a_0...a_n); \mathbf{cc}(a_0...a_n \mapsto b_0...b_n) \;=\; P(b_0...b_n)$$

That is, the copycat act as *renaming*. This corresponds to the fact that, by the copy-cat, the interactions done at a_i become those at b_i. We extensively use this law in the following. For a deeper study of copy-cat behaviours, see [18].

We now utilise process representation of strategies for description and reasoning. We notice each algebraic equation of processes precisely describes the deduction we may use when we reason in terms of action sequences and their composition: process composition makes such reasoning both clearer and simpler.

B.10. **Equations for Proposition 3.7 (projection and strength). (i):** We present the involved strategies in process representation. First, the projection from $A \times B$ to A is simply:

$$\pi_1^{ea} \stackrel{\text{def}}{=} e(\tilde{a}_1\tilde{b}).\overline{a}(\tilde{a}_2).\mathbf{cc}(\tilde{a}_2 \mapsto \tilde{a}_1)$$

π_2 is simply the dual of the above. Notice clearly projections are total. We next depict the behaviour of $\langle \sigma,\ \tau \rangle$. Assume:

$$\sigma^{ca} \;=\; c(\tilde{c}).\overline{a}(\tilde{a}).P(\tilde{a}) : C \rightarrow A \qquad \text{and} \qquad \tau^{cb} \;=\; c(\tilde{c}).\overline{b}(\tilde{b}).Q(\tilde{b}) : C \rightarrow B$$

Then we define:

$$\langle \sigma,\ \tau \rangle^{ce} : C \rightarrow A \times B \stackrel{\text{def}}{=} c(\tilde{c}).\overline{e}(\tilde{a}\tilde{b}).(P(\tilde{a})\ ;\ Q(\tilde{b})).$$

The copycat functions as renaming in this setting, so that we get:

$$\langle \sigma,\ \tau \rangle^{ce}; \pi_1^{ea} \;=\; c(\tilde{c}).\overline{a}(\tilde{a}_2).P(\tilde{a}_2)$$

which is nothing but σ^{ca}.

(ii) Strengths can be written down using process representation in the way:

$$\mathtt{st}_{A,B}^{e_1 e_2} \stackrel{\text{def}}{=} e_1(\tilde{a}e_1').\overline{e}_2(e_2').!e_2'(e_2'').\overline{e_1}'(b).!b(\tilde{b}).\overline{e_2}''(\tilde{a}'\tilde{b}').\mathbf{cc}(\tilde{a}'\tilde{b}' \mapsto \tilde{a}\tilde{b}) : A \times TB \rightarrow T(A \times B)$$

with $\mathbf{cc}(\tilde{a}'\tilde{b}' \mapsto \tilde{a}\tilde{b})$ is a copy-cat from $\tilde{a}'\tilde{b}'$ to $\tilde{a}\tilde{b}$. Here again we assumed A and B have singleton unique initial sorts. Using this representation, we can mechanically verify they satisfy the defining equations of strengths, cf. [42]. We show one example. Others are similar. We first notice that the isomorphisms witnessing associativity and commutativity of products are essentially copy-cat strategies (as other isomorphisms are) thus can be neglected. Assume:

$$\begin{array}{rcl}\mathsf{id}_A^{e_1e_2} & \stackrel{\text{def}}{=} & \mathbf{cc}(e_1 \mapsto e_2) \equiv e_1(\tilde{e}_1).\overline{e_2}(\tilde{e}_2).\mathbf{cc}(\tilde{e}_2 \mapsto \tilde{e}_1) \\ \mathtt{st}_{B,C}^{c_1c_2} & \stackrel{\text{def}}{=} & c_1(\tilde{b}e_1').\overline{c}_2(e_2').!e_2'(e_2'').\overline{e_1}'(c).!c(\tilde{c}).\overline{e_2}''(\tilde{b}'\tilde{c}').\mathbf{cc}(\tilde{b}'\tilde{c}' \mapsto \tilde{b}\tilde{c})\end{array}$$

Then

$$(\mathsf{id}_A^{e_1e_2} \times \mathtt{st}_{B,C}^{c_1c_2})^{ef} \quad = \quad e(\tilde{e}_1\tilde{b}e_1').\overline{f}(\tilde{e}_2e_2').(\mathbf{cc}(\tilde{e}_2 \mapsto \tilde{e}_1); !e_2'(e_2'').\overline{e_1}'(c).!c(\tilde{c}).\overline{e_2}''(\tilde{b}'\tilde{c}').\mathbf{cc}(\tilde{b}'\tilde{c}' \mapsto \tilde{b}\tilde{c}))$$

On the other hand, suppose

$$\mathtt{st}_{A,(B\times C)}^{fg} \quad \stackrel{\text{def}}{=} \quad f(\tilde{e}_2e_2').\overline{g}(g').!g'(e_4'').\overline{e_2}'(e_2'').!e_2''(\tilde{b}'\tilde{c}').\overline{e}_4''(\tilde{a}'\tilde{b}'\tilde{c}').\mathbf{cc}(\tilde{a}'\tilde{b}'\tilde{c}' \mapsto \tilde{a}\tilde{b}\tilde{c}).$$

Note when two agents are composed, the pairs $\langle \overline{f}(\tilde{e}_2e_2'),\ f(\tilde{e}_2e_2')\rangle$, $\langle e_2'(e_2''),\ \overline{e_2}'(e_2'')\rangle$, and $\langle \overline{e_2}''(\tilde{b}'\tilde{c}'),\ e_2''(\tilde{b}'\tilde{c}')\rangle$ are cancelled. So we have:

$$(\mathsf{id}_A^{e_1e_2} \times \mathtt{st}_{B,C}^{c_1c_2})^{ef}; \mathtt{st}_{A,(B\times C)}^{fg} \quad = \quad e(\tilde{e}_1\tilde{b}e_1').\overline{g}(g').!g'(e_4'').\overline{e_1}'(c).!c(\tilde{c}).\overline{e}_4''(\tilde{a}'\tilde{b}'\tilde{c}').\mathbf{cc}(\tilde{a}'\tilde{b}'\tilde{c}' \mapsto \tilde{a}\tilde{b}\tilde{c})$$

which is α-equal to $\mathtt{st}_{A\times B,C}^{eg}$, thus establishing its coherence with the associativity of products. $\square$

B.11. **Equations for Proposition 3.8 (iii) (partial exponential).** We first repeat the construction of partial exponential and the evaluation map in the following. The scheme follows those in the usual game semantics [4, 27], adapted to the call-by-value behaviour.

(i) For $\sigma : C \otimes A \to B$, assume σ has a form:

$$\sigma^{eb} \stackrel{\text{def}}{=} e(c_1..c_n a_1..a_m).P(b)$$

where c_i comes from C and a_i comes from A. Then we form $p\lambda(\sigma)^{cw} : C \to A \Rightarrow B$ as:

$$c(c_1..c_n).\overline{w}(w').w'(a_1..a_m b).P(b)$$

(ii) Given A and B, $\mathsf{ev}^{ef} : (A \Rightarrow B) \otimes A \to B$ is given as:

$$e(wa_1..a_n).\overline{w}(a_1'..a_n'b).\mathbf{cc}(a_1'..a_n'b \mapsto a_1..a_n f)$$

Let us check the defining commutativity with the above σ. We use the functionality of the copy-cat as renaming, cf.B.9. The construction of $\otimes$ is easy because the transpose and identity are both total. ln

$$\begin{array}{rcl}(p\lambda(\sigma) \otimes \mathsf{id}_A)^{lm} \ ; \ \mathsf{ev}^{mn} & = & (\nu m)(l(\tilde{c}\tilde{a}).\overline{m}(w\tilde{a}).(w(\tilde{a}'b).P(b) \mid \mathbf{cc}(\tilde{a}' \mapsto \tilde{a})) \mid \mathsf{ev}^{mn}) \\ & = & (\nu m)(l(\tilde{c}\tilde{a}).\overline{m}(w\tilde{a}).w(\tilde{a}b).P(b) \mid \mathsf{ev}^{mn}) \\ & = & (\nu m)(l(\tilde{c}\tilde{a}).\overline{m}(w\tilde{a}).w(\tilde{a}b).P(b) \mid m(w\tilde{a}).\overline{w}(\tilde{a}''b).\mathbf{cc}(\tilde{a}''b \mapsto \tilde{a}n)) \\ & = & (\nu m)(l(\tilde{c}\tilde{a}).P(b)\{\tilde{a}''/\tilde{a}\} \mid \mathbf{cc}(\tilde{a}''b \mapsto \tilde{a}n)) \\ & = & l(\tilde{c}\tilde{a}).P(n) \stackrel{\text{def}}{=} \sigma,\end{array}$$

as required. For uniqueness, we reason by induction on the length of sequences from another possible candidate for a curried map, which is easy. $\square$

B.12. **Interpretation of constants.** Below we give the interpretation of each constant, using a process representation for clarity. The generic strategy γ_A for each type A, which is used to interpret the conditional, is also given at the end. Below we write $\sigma : A$ for $\sigma : \mathbf{1} \rightarrow A$.

$$\begin{array}{rcl} \overline{\Omega} : [\![\alpha]\!] & \stackrel{\text{def}}{=} & \perp \\ \overline{n}^{ef} : \mathsf{nat} & \stackrel{\text{def}}{=} & e.\overline{f}\langle n\rangle \\ \overline{\mathsf{succ}}^{ef} : \mathsf{nat} \Rightarrow \mathsf{nat} & \stackrel{\text{def}}{=} & e.\overline{f}(c).!\sum_{n\in\omega} c\langle n\rangle(c').\overline{c'}\langle n+1\rangle \\ \overline{\mathsf{zero?}}^{ef} : \mathsf{nat} \Rightarrow \mathsf{bool} & \stackrel{\text{def}}{=} & e.\overline{f}(c).\,!(c\langle 0\rangle(c').\overline{c'}\langle\mathsf{true}\rangle + \sum_{n>0} c\langle n\rangle(c').\overline{c'}\langle\mathsf{false}\rangle) \\ \gamma_A^{ea} : \mathsf{bool} \otimes A \otimes A \rightarrow A & \stackrel{\text{def}}{=} & e\langle\mathsf{true}\rangle(\tilde{a_1}\tilde{a_2}).\overline{a}(\tilde{a}).\mathbf{cc}(\tilde{a} \mapsto \tilde{a_1}) \;+\; e\langle\mathsf{false}\rangle(\tilde{a_1}\tilde{a_2}).\overline{a}(\tilde{a}).\mathbf{cc}(\tilde{a} \mapsto \tilde{a_2}). \end{array}$$

We observe that, if we are to have the conditional as a constant in $\mathrm{PCF_V}$, we can use:

$$\overline{\mathsf{cond}_A} : \mathsf{bool} \rightharpoonup A \rightharpoonup A \rightharpoonup A \stackrel{\text{def}}{=} p\lambda(p\lambda(p\lambda(\gamma_A)))$$

Notice also there is an obvious inverse translation.

Appendix C. $\mathrm{PCF_V}$

Assuming the reader is familiar with the language PCF and its call-by-value variant, cf. [23, 55, 53], we give a brief review of syntax and operational semantics we are working with in the present exposition. Our treatment is nearest to [23].

C.1. **Syntax.** Assume given an infinite set of *variables*, ranged over by $x, y, z, \ldots$. Then the syntax of the language is given as follows.

$$\begin{array}{rcl} \alpha & ::= & \iota \mid o \mid \alpha \Rightarrow \beta \\ M & ::= & x \mid \lambda x^{\alpha}.M \mid MM \mid \mathsf{cond}\ L\ M_1\ M_2 \mid \boldsymbol{\mu} x^{\alpha\Rightarrow\beta}.M \mid c \end{array}$$

where c is a constant we introduce in C.2.

C.2. **Typing Rules.** An *environment* is a list of pairs of a variable and a type, where all variables are distinct, ranged over by $\Gamma, \Delta, \ldots$. Then the typing rules of $\mathrm{PCF_V}$ is given as follows.

$$\Gamma, x:\alpha, \Gamma' \rhd x:\alpha \qquad \frac{c \text{ is a constant of type } \alpha}{\Gamma \rhd c : \alpha} \qquad \frac{\Gamma, x:\alpha \rhd M:\beta}{\Gamma \rhd \lambda x^{\alpha}.M : \alpha \Rightarrow \beta}$$

$$\frac{\Gamma \rhd M : \alpha \Rightarrow \beta \quad \Gamma \rhd N : \alpha}{\Gamma \rhd MN : \beta} \qquad \frac{\Gamma \rhd L : o \quad \Gamma \rhd M : \alpha \quad \Gamma \rhd N : \alpha}{\Gamma \rhd \mathsf{cond}\ L\ M\ N : \alpha} \qquad \frac{\Gamma, x : \alpha \Rightarrow \beta \rhd M : \alpha \Rightarrow \beta}{\Gamma \rhd \boldsymbol{\mu} x.M : \alpha \Rightarrow \beta}$$

As a set of constants, we assume: $n : \iota$ for each numeral n, $\Omega : \alpha$ for each type α, $\mathsf{succ} : \iota \Rightarrow \iota$, and $\mathsf{zero?} : \iota \Rightarrow o$.[5] Terms of form $\rhd M : \alpha$ is called a *closed term*. *Values* are either natural numbers, abstraction, boolean values, or function constants. We often write $V, U, \ldots$ for values. One may also include variables into values, but no difference in semantics comes about as far as closed terms go.

C.3. **Evaluation relation.** In the style of natural semantics, we define an evaluation relation $\Downarrow$ as follows (we omit the environment and type assignment below).

$$V \Downarrow V \qquad \frac{M \Downarrow \lambda x.M_0 \quad N \Downarrow V \quad M_0\{V/x\} \Downarrow U}{MN \Downarrow U} \qquad \frac{M\{\boldsymbol{\mu} x.M/x\} \Downarrow V}{\boldsymbol{\mu} x.M \Downarrow V}$$

$$\frac{M \Downarrow n}{\mathsf{succ}\ M \Downarrow n+1} \qquad \frac{M \Downarrow 0}{\mathsf{zero?}M \Downarrow \mathsf{true}} \qquad \frac{M \Downarrow n+1}{\mathsf{zero?}M \Downarrow \mathsf{false}}$$

$$\frac{L \Downarrow \mathsf{true} \quad M_1 \Downarrow V}{\mathsf{cond}\ L\ M_1\ M_2 \Downarrow V} \qquad \frac{L \Downarrow \mathsf{false} \quad M_2 \Downarrow U}{\mathsf{cond}\ L\ M_1\ M_2 \Downarrow U}$$

[5] The conditional can be taken as a constant for each type. No essential difference comes about, see Appendix B.12 for respective interpretations. Similarly for the introduction of (call-by-value) Y-combinators instead of recursion, cf.[53].

C.4. **Remark.** If we assume constants for conditionals for all PCF-types, then for each such $\mathsf{cond}'_\alpha : o \Rightarrow (\alpha \Rightarrow (\alpha \Rightarrow \alpha))$ we have the evaluation rules: $L \Downarrow \mathsf{true} \;\Rightarrow\; \mathsf{cond}'_\alpha\, L \Downarrow \lambda x^\alpha.\lambda y^\alpha.x$ and $L \Downarrow \mathsf{false} \;\Rightarrow\; \mathsf{cond}'_\alpha\, L \Downarrow \lambda x^\alpha.\lambda y^\alpha.y$. Notice that we can translate $\mathsf{cond}\ L\ M\ N : \alpha$ as $(((\mathsf{cond}'_{\iota\Rightarrow\alpha}\, L)(\lambda z^\iota.M))(\lambda z^\iota.N))0$ with z fresh, and cond'_α as $\lambda z^o.x^\alpha.y^\alpha.\mathsf{cond}\ z\ x\ y$. It is easy to check that these translations preserve and reflect the observability.

C.5. **Observational preorder.** Finally we define an *observational preorder* on closed terms by the following: $M \preceq_{obs} N$ iff, for any well-typed context of a program type $C[\cdot]$, we have $C[M] \Downarrow n$ iff $C[N] \Downarrow n$. We note that this is the same thing as considering convergence at *all* types, a situation quite different from the case of call-by-name evaluation. We also write $\approx_{obs}$ for the induced equivalence.