HAL
open science

Justified Sequences in String Diagrams: a Comparison Between Two Approaches to Concurrent Game Semantics

Clovis Eberhart, Tom Hirschowitz

To cite this version:

Clovis Eberhart, Tom Hirschowitz. Justified Sequences in String Diagrams: a Comparison Between Two Approaches to Concurrent Game Semantics. 7th Conference on Algebra and Coalgebra in Computer Science (CALCO 2017), 2017, Ljubljana, Slovenia. pp.10, 10.4230/LIPIcs.CALCO.2017.10 . hal-01715405

HAL Id: hal-01715405
https://hal.science/hal-01715405v1

Submitted on 22 Feb 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Justified Sequences in String Diagrams: a Comparison Between Two Approaches to Concurrent Game Semantics

Clovis Eberhart[1] and Tom Hirschowitz[2]

1 **LAMA, CNRS, Université Savoie Mont Blanc, Bâtiment Le Chablais, Campus Scientifique, 73376 Le Bourget-du-Lac Cedex, France**
2 **LAMA, CNRS, Université Savoie Mont Blanc, Bâtiment Le Chablais, Campus Scientifique, 73376 Le Bourget-du-Lac Cedex, France**

Abstract

Recent developments of game semantics have given rise to new models of concurrent languages. On the one hand, an approach based on *string diagrams* has given models of CCS and the π-calculus, and on the other hand, Tsukada and Ong have designed a games model for a non-deterministic λ-calculus. There is an obvious, shallow relationship between the two approaches, as they both define innocent strategies as sheaves for a Grothendieck topology embedding "views" into "plays". However, the notions of views and plays differ greatly between the approaches: Tsukada and Ong use notions from standard game semantics, while the authors of this paper use string diagrams. We here aim to bridge this gap by showing that even though the notions of plays, views, and innocent strategies differ, it is mostly a matter of presentation.

1998 ACM Subject Classification F.3.2 Semantics of Programming Languages

Keywords and phrases Concurrency, Sheaves, Presheaf models, Game Semantics

Digital Object Identifier 10.4230/LIPIcs.CALCO.2017.10

1 Introduction

1.1 Two approaches to concurrent game semantics

Game semantics [11], a branch of denotational semantics in which types are interpreted as some sort of games and programs as strategies in these games, has led to fully abstract models for a variety of functional languages. Recent advances in concurrent game semantics have produced new games models for a non-deterministic, simply-typed λ-calculus on the one hand [19], and for CCS and the π-calculus on the other hand [9, 10, 6]. These models are based on categories of innocent and concurrent strategies that are defined in both cases as categories of *sheaves* over a site of plays.

The first model, by Tsukada and Ong, has proven to successfully extend the most fundamental results of game semantics (interpreting terms as innocent strategies and composing strategies to form a CCC of arenas and innocent strategies) to a non-deterministic λ-calculus. The other approach, while arguably more complex and not as well developed, aims to give a general framework to build games models for different calculi, in order to study translations between them.

There is a clear, yet informal relationship between the two approaches in that they both define innocent strategies as sheaves for a Grothendieck topology induced by embedding *views* into *plays*. However, despite this similarity, the notions of views and plays differ significantly. Indeed, Tsukada and Ong [19] define them as *justified sequences* of moves satisfying

© Clovis Eberhart and Tom Hirschowitz;
licensed under Creative Commons License CC-BY
7th Conference on Algebra and Coalgebra in Computer Science (CALCO 2017).
Editors: Filippo Bonchi and Barbara König; Article No. 10; pp. 10:1–10:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

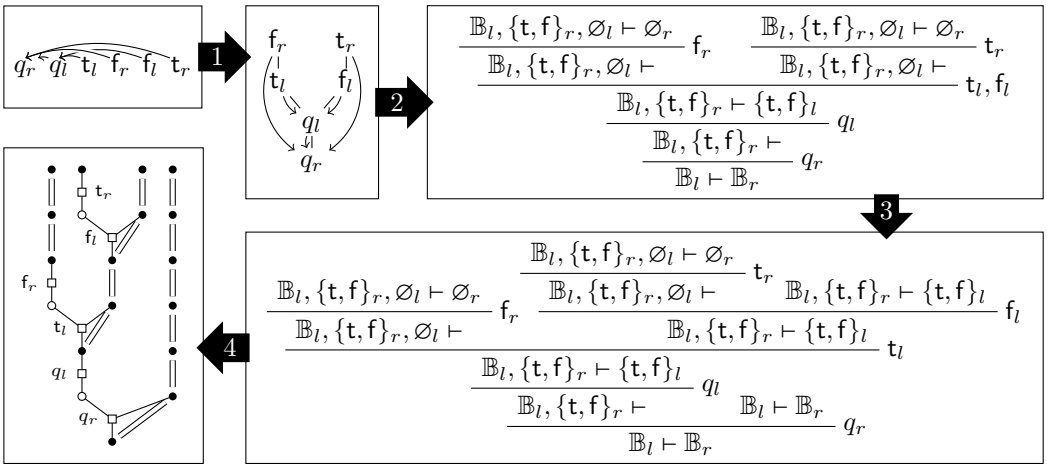


Figure 1 The big picture.

additional conditions, as in standard Hyland-Ong/Nickau (HON) game-semantics [11, 16], while in [9, 10, 6], plays are defined as *ad hoc* string diagrams describing the game, as originally suggested by Melliès in a different setting (*circa* 2008, published as [15]). Since the notions of plays differ significantly, it is legitimate to wonder to what degree the approaches are related.

1.2 The level of views and plays

In this paper, we go beyond this informal similarity and show a tight correspondence between the two approaches at the level of plays. Specifically, for each pair of *arenas* [11] A and B, we design categories $\mathbb{E}(A \vdash B)$ and $\mathbb{E}^{\mathbb{V}}(A \vdash B)$ respectively of plays and views for HON games, in the same spirit as those of the models for CCS and the π-calculus.

There are also standard categories $\mathbb{P}_{A,B}$ and $\mathbb{V}_{A,B}$ of plays and views as defined in standard game semantics (and which we call *HON-plays* and *HON-views* to disambiguate), with a notion of morphism inspired by Melliès [14]. We then embed these categories into $\mathbb{E}(A \vdash B)$ and $\mathbb{E}^{\mathbb{V}}(A \vdash B)$ respectively (whose objects we simply call *plays* and *views*).

We define this embedding using a third model, whose plays are proof trees in an *ad hoc* sequent calculus, and whose views are branches of those trees. The categories of trees and branches are equivalent to those of plays and views respectively, so they can be thought of as another possible representation of these objects. Trees may also be seen as a maximal parallelisation of HON-plays, while branches are simply equivalent to HON-views.

Let us show how this embedding works on an example, illustrated in Figure 1. HON-plays are based on the notion of *arena*:

▶ **Definition 1.** An *arena* is a simple forest, i.e., a directed, simple graph in which all vertices are uniquely reachable from a unique root (vertex without a parent).

Vertices are called *moves*, and roots deemed *initial*. A move m' is said to be *justified* by m when it is one of its children.

Let M_A be the set of moves of A, the *ownership* of any move $m \in M_A$ is O (for *Opponent*) if the length of the unique path from a root to m is even, and P (for *Proponent*) otherwise. So, e.g., all roots have ownership O. We denote this map $M_A \to \{P, O\}$ by λ_A.

▶ **Example 2.** The boolean arena $\mathbb{B}$ has a single root q, which is an Opponent move, and two Proponent moves t and f, both justified by q.

A HON-play on a pair of arenas (A, B) is a sequence of moves in $M_A + M_B$ that verifies some additional properties. The full definitions are recalled in Section 2. In our example, we study a play on $(\mathbb{B}_l, \mathbb{B}_r)$, where l and r are only present to show in which copy of $\mathbb{B}$ moves are played. We show how s may be mapped to a play P in our setting and P may conversely be mapped back to a HON-play isomorphic to s.

We here give a simplified description of our plays that should give the reader enough intuition to understand the mapping. The precise definitions can be found in Section 3.1. Our plays are based on the notion of *position*. Positions are sets of *players* that are labelled by sequents of arenas, either positive sequents $(\Gamma \vdash)$ or negative ones $(\Gamma \vdash A)$. In the first case, players are depicted $\circ$, and $\bullet$ in the second case. Positions may evolve over time according to the *moves* the different players play. Each type of player can only do the following: positive players $(\Gamma, A, \Delta \vdash)$ may turn into negative players $(\Gamma, A, \Delta \vdash A \cdot m)$ for any root m of A, and negative players $(\Gamma \vdash A)$ may spawn a positive player $(\Gamma, A \cdot m \vdash)$ for any root m of A (and they continue to exist after the move). These two moves may be depicted as and , with the initial position of the move at the bottom, and its final position at the top. When a move is played by a player, the rest of the position is left untouched. For example, is a move on a position with two players, with the left player playing and the right player left untouched. A *play* is then just a vertical pasting of moves, up to permutation of independent moves. A morphism of plays $f\colon P \to Q$ is an injective mapping of moves of P to those of Q that respect the play structure: $f(x)$ must have the same type as x, and if the player who plays x is equal to the one who plays y, then the players who play $f(x)$ and $f(y)$ must be equal too (and similarly if x creates a player who is equal to the one who plays y, etc).

As a first step (step 1 in Figure 1), we map our example play s to its P-view tree [3], a tree whose branches are the HON-views of s. There are two types of "arrows" in this tree: the ones depicted as real arrows, which correspond exactly to the ones in s, and the ones that create the tree structure, which correspond to the *view* of each move. This tree has some nice properties: in particular, it may only branch at odd depth. We then map this tree (step 2) to a proof tree in a sequent calculus based on arenas. A node x labelled by a sequent S in this tree may be a child of y labelled by S' exactly when, in our game, a player labelled by S' may make a move that produces of player labelled by S. Just like P-view trees, these proof trees may branch with an arbitrary arity, but only at odd depth. Even though both notions are trees and look alike (the structure is clearly similar), there are some subtle differences: the P-view tree is labelled by moves, while the proof tree is labelled by sequents of arenas; the P-view tree contains pointers, while the proof tree does not. A little bit of work has to be done to prove that this mapping is a full embedding. We give a proof for the composite of steps 1 and 2 in Section 4.

Steps 3 and 4 are conceptually simple, but for lack of space, we do not explain them in the rest of the paper. Step 3 maps the proof tree, which may branch with an arbitrary degree, to a similar tree that only branches in a binary way. Let us call the result of step 3 a *sequential* proof tree (in the sense that we have sequentialised some of the tree structure). To do this, we change the rules on which our proof trees are built. Here, the rules that have a positive sequent as conclusion are the same as in the previous proof trees, but the rules that have a negative sequent as conclusion have necessarily two premises: a positive one like in the previous proof trees, and a copy of the conclusion. Sequentialising a proof tree is now completely obvious: if a node has n premises, we just apply the sequential rule

n times. There is a choice to be made on the order in which we sequentialise the rules, but all these choices lead to isomorphic plays in the end. For example, we have decided to linearise the tree by applying t_l first, and then f_l in our example. Step 4 is completely direct: all the rules of the proof tree now exactly correspond to moves in our game, so we just mimic the structure of the proof tree. Nodes in the tree become players with the same label, and deduction rules become moves in our game. Here again, since the tree is a branching structure, we may choose to apply the moves in different orders, but all choices lead to isomorphic plays. In our example, we have chosen to play t_l, f_r, f_l, and t_r in that order, but we could have played f_r at any point after t_l (and all the resulting plays are isomorphic).

To show that this mapping reduces to an equivalence when restricted to views on both sides, we must explain what a view is in our setting. If we see the equal signs of a play as mathematical equality (which is indeed the case in the actual definition of plays), then we may collapse the whole structure along these equal signs to obtain a tree structure. Let us call such a structure a *play tree*. A play is a view when its play tree is non-branching. Now, our embedding reduces to an equivalence when restricted to views because HON-views are mapped to non-branching P-view trees, which are ultimately mapped to non-branching plays, i.e., views, and views conversely all come from non-branching proof trees, which all come from non-branching P-view trees, which in turn all come from HON-views.

Let us finally mention the case of morphisms. Morphisms of HON-plays are injective functions that respect views. Through step 1, they are exactly transported to (injective) morphisms of trees. Through step 2, they are exactly transported to inclusions of proof trees. Now, the easiest way to understand the equivalence, in terms of morphisms, is to see the resulting play as a tree by collapsing all equal signs, as explained above. Then inclusions of proof trees are exactly transported to inclusions of play trees, which are exactly morphisms of plays.

We thus obtain, for each pair of arenas A and B, a commuting square

$$\begin{tikzcd} \mathbb{V}_{A,B} \arrow[r, hook, "\mathsf{i}_{HON}"] \arrow[d, "F^{\mathbb{V}}"'] & \mathbb{P}_{A,B} \arrow[d, "F"] \\ \mathbb{E}^{\mathbb{V}}(A \vdash B) \arrow[r, hook, "\mathsf{i}"'] & \mathbb{E}(A \vdash B) \end{tikzcd} \tag{1}$$

of embeddings of categories, where i_{HON} denotes the embedding of HON-views into HON-plays, i denotes the embedding of views into plays, and $F^{\mathbb{V}}$ and F denote the constructed embeddings respectively from HON-views into views and from HON-plays into plays. Our first result is that all these embeddings are full and that $F^{\mathbb{V}}$ is an equivalence of categories (Theorem 29).

1.3 The level of strategies

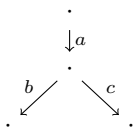
However, we are not only interested in comparing views and plays, but also *innocent strategies*, which are at the core of game semantics. The square (1) gives a correspondence at the level of plays, but it also yields a tight correspondence between strategies in both approaches. More precisely, it induces an equivalence between innocent strategies in both contexts and shows that this equivalence is compatible with *innocentisation*.

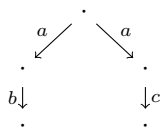
To be more precise, there are two notions of innocent strategy in standard game semantics: the first one is a prefix-closed set of views, the second one is a prefix-closed set of plays verifying an extra condition called *innocence*. In both approaches, the first notion generalises to *presheaves* on views, which we call *behaviours* and *TO-behaviours* (for

Tsukada-Ong). The second notion generalises to *sheaves* on plays, which we simply call *innocent strategies* and *innocent TO-strategies*. In particular, mere presheaves on plays are possibly non-innocent strategies.

The idea behind this generalisation is the following: a prefix-closed set of views (in, say, $\mathbb{V}_{A,B}$) is a presheaf of booleans $B\colon \mathbb{V}_{A,B}^{op} \to 2$ (where 2 is the ordinal $0 \to 1$ viewed as a category). Similarly, a prefix-closed set of plays is a presheaf $S\colon \mathbb{P}_{A,B}^{op} \to 2$. This presheaf is a sheaf for the Grothendieck topology induced by the embedding of $\mathbb{V}_{A,B}$ into $\mathbb{P}_{A,B}$ when $S(p)$ is accepted if and only if $S(v)$ is accepted for all views $v \to p$ (which exactly states *innocence*, the condition stating that a player can change its behaviour only according to what they have "seen", i.e., their view, of the play).

However, that is not sufficient to model concurrent strategies, because there may be several different ways to accept a play (or, in other words, a machine may be in several different states after a given trace). The classical example is that of Milner's coffee machines, which are the labelled transition systems depicted below.





Both machines accept exactly the same traces: ε, a, ab, and ac. However, one has a single way of accepting the trace a, after which it still accepts b and c, while the other makes a choice when accepting a whether to accept b or c. The first machine has one way of accepting a, while the second has two. Thus, modelling concurrent strategies correctly requires to know all the different states the strategy can end in after reading a trace: it is not a presheaf of booleans, but a presheaf of sets.

The functors F and $F^{\mathbb{V}}$ give rise to $\Delta_F\colon \mathbb{E}\widehat{(A \vdash B)} \to \widehat{\mathbb{P}_{A,B}}$ and $\Delta_{F^{\mathbb{V}}}\colon \mathbb{E}^{\mathbb{V}}\widehat{(A \vdash B)} \to \widehat{\mathbb{V}_{A,B}}$, where Δ_f is precomposition by f^{op}. Since $F^{\mathbb{V}}$ is an equivalence of categories, so is $\Delta_{F^{\mathbb{V}}}$, so behaviours and TO-behaviours are equivalent.

A strategy is *innocent* when it is in the essential image of $\prod_{\mathsf{i}}$ (or $\prod_{\mathsf{i}_{HON}}$), where $\prod_f$ denotes right Kan extension along f^{op}. This may also be seen as a sheaf condition stating that an innocent strategy accepts a play if and only if it accepts all the views that can be embedded into that play. Using Guitart's theory of *exact squares* [8], the square (1) provides a categorical explanation of why both induced categories of innocent strategies are equivalent. Indeed, it is exact (Corollary 33), which means that the square

$$\begin{tikzcd}
\widehat{\mathbb{V}_{A,B}} \arrow[r, hook, "\prod_{\mathsf{i}_{HON}}"] & \widehat{\mathbb{P}_{A,B}} \\
\mathbb{E}^{\mathbb{V}}\widehat{(A \vdash B)} \arrow[u, "\Delta_{F^{\mathbb{V}}}"] \arrow[r, hook, "\prod_{\mathsf{i}}"'] & \mathbb{E}\widehat{(A \vdash B)} \arrow[u, "\Delta_F"']
\end{tikzcd} \tag{2}$$

commutes up to isomorphism. In other words, Δ_F turns into an equivalence when restricted to innocent strategies and the *innocentisation* functors $\prod_{\mathsf{i}}$ and $\prod_{\mathsf{i}_{HON}}$ (which map any behaviour to the innocent strategy that "behaves similarly") are compatible with this equivalence.

▶ Remark. Had we wanted to make F an equivalence of categories rather than a mere full embedding, we could easily have imposed an additional condition on our plays akin to alternation in a classical HON-game setting. The point is that we want to compare the purely diagrammatic notion of play with the classical one. And since we obtain an

equivalence between both notions of innocent strategies anyway, we feel the result is in fact more convincing.

Note however that this work does not take composition of strategies into account, which is admittedly the Achilles' heel of the string diagrammatic approach as it stands now. Indeed, while Tsukada and Ong prove that their innocent strategies compose, we still don't know how to compose strategies into cut-free strategies in our setting. (The equivalence between both notions of innocent strategies actually gives a way to compose innocent strategies in our setting by composing in the other model, but it is not exactly an illuminating result, and in particular does not lift to non-innocent strategies.)

1.4 Related work

This paper compares Tsukada and Ong's model [19] to a model inspired by *playgrounds* [9, 10, 6] and which builds on ideas from *presheaf models* [12], *causal models* [17], and *game semantics* [1]. The notion of play, which is based on string diagrams, is close in spirit to Melliès's work [15]. The notion of trees that we use to bridge the gap between both models is reminiscent of the notion of legally justified trees (also known as P-view trees) by Boudes [3]. Let us also mention different approaches to concurrent game semantics [7, 13], based on variations of traditional games semantics, or [18], based on event structures.

Another paper by Tsukada and Ong [20] is quite close to this work. Their main result is a link between the notion of HON-play and another well-known notion of terms. The papers differ in that Tsukada and Ong link the notion of HON-play to notions that do not belong to game semantics (namely *resource terms*), thus finding new links between game semantics and other models, while we want to connect different models of game semantics.

The reader will notice that the interpretation of terms as strategies is not treated in the current paper. This point is treated in an unpublished paper [5], in which we use singular and geometric realisation functors to give an interpretation of terms of the non-deterministic λ-calculus studied by Tsukada and Ong into innocent strategies, show that it is the same as Tsukada and Ong's, and recover the *definability* result (that any innocent strategy is isomorphic to the interpretation of a normal form).

1.5 Overview

We start by recalling standard notions of game semantics as well as Tsukada and Ong's work in Section 2. Then, in Section 3, we set out to define two new models of HON games, one based on string diagrams, the other on proof trees, and show that the two have equivalent categories of plays, views, and strategies. In Section 4, we use the equivalences proved in the previous section to give the core result of this paper, which is the relationship between the categories of plays and views in our model based on string diagrams and Tsukada and Ong's. Finally, in Section 5, we build on the relationships shown in Section 4 to show the second result of this paper, which is that both models have equivalent categories of strategies, and that this equivalence is compatible with the embedding of behaviours into strategies.

2 Tsukada and Ong's model

Let us start with a brief recapitulation on Tsukada and Ong's categories of views and plays, as well as their notion of strategy.

As usual in game semantics, games are based on arenas (Definition 1).

▶ **Notation 3.** *We denote by $\sqrt{A}$ the set of roots of A. If A is an arena and m is in $\sqrt{A}$, then $A \cdot m$ is the arena strictly below m. If A is an arena, we denote by $m.A$ the tree t whose root is m and such that $t \cdot m = A$. Note that any arena is a coproduct of trees, so it can be written $A = \sum_{m \in \sqrt{A}} m.(A \cdot m)$.*

Let us fix arenas A and B. Let $A \rightarrowtriangle B$ denote the simple graph obtained by adding to $A + B$ an edge $b \to a$ for all $b \in \sqrt{B}$ and $a \in \sqrt{A}$ (if B is non-empty, otherwise $A \rightarrowtriangle B = \varnothing$). The notion of ownership straightforwardly extends to $A \rightarrowtriangle B$ since all paths from any root to some vertex v have the same length. Concretely, ownership is left unchanged in B but reversed in A.

▶ **Definition 4.** A *justified sequence* on (A, B) consists of a natural number $n \in \mathbb{N}$, equipped with maps $f: n \to M_A + M_B$ and $\varphi: n \to \{0\} \uplus n$ (here and later in the paper, we use n to denote the set $\{1, \ldots, n\}$) such that, for all $i \in n$,

- $\varphi(i) < i$,
- if $\varphi(i) = 0$ then $f(i) \in \sqrt{B}$, and
- if $\varphi(i) \neq 0$, then $f(\varphi(i))$ is a parent of $f(i)$ in $A \rightarrowtriangle B$.

We will draw a justified sequence (n, f, φ) as the sequence of its $f(i)$'s, with arrows to denote φ, as is standard in game semantics.

For any $i \in n$, the *view* $\lceil (n, f, \varphi) \rceil_i$ *of* i *in* (n, f, φ) is the subset of n defined inductively by:

- $\lceil (n, f, \varphi) \rceil_i = \{i\}$ if i is an Opponent move with $\varphi(i) = 0$,
- $\lceil (n, f, \varphi) \rceil_i = \lceil (n, f, \varphi) \rceil_j \cup \{i\}$ if i is an Opponent move with $\varphi(i) = j > 0$,
- $\lceil (n, f, \varphi) \rceil_i = \lceil (n, f, \varphi) \rceil_{i-1} \cup \{i\}$ if i is a Proponent move.

A justified sequence $s = (n, f, \varphi)$ on (A, B) is *P-visible* when, for all Proponent moves i, $\varphi(i) \in \lceil s \rceil_i$. We further say that s is *alternating* when, for all $i \in n-1$, $\lambda_{A \rightarrowtriangle B}(i) \neq \lambda_{A \rightarrowtriangle B}(i+1)$.

▶ **Definition 5.** A *HON-play* on the pair of arenas (A, B) is a P-visible, alternating, justified sequence on (A, B) of even length.

A morphism of HON-plays $g: (n, f, \varphi) \to (n', f', \varphi')$ is an injective map $g: n \to n'$ such that: $f'(g(i)) = f(i)$ for all $i \in n$, $\varphi'(g(i)) = g(\varphi(i))$ for all $i \in n$ (with the convention that $g(0) = 0$), and $g(2i) = g(2i - 1) + 1$ for all $i \in n/2$. The last condition states that g should preserve blocks of an Opponent move and the next Proponent move (so-called *OP-blocks*).

▶ **Proposition 6.** *HON-plays and morphisms between them form a category $\mathbb{P}_{A,B}$, with composition given by composition of underlying maps.*

▶ **Example 7.** The sequence at the top-left of Figure 1 is a play on $(\mathbb{B}, \mathbb{B})$, where we have written m_l when m is played in the left-hand copy of $\mathbb{B}$, and m_r when it is played in the right-hand one.

▶ **Definition 8.** If $s = (n, f, \varphi)$ is a justified sequence, i and j are in n, and $\varphi(j) = 0$, we say that i is *hereditarily justified* by j if $i = j$ or $\varphi(i)$ is hereditarily justified by j.

A *thread* of s is a maximal sub-sequence of s in which all moves have the same hereditary justifier. The pointers of a thread are inherited from s.

▶ **Definition 9.** A *HON-view* on (A, B) is a non-empty HON-play $s = (n, f, \varphi)$ such that $\lceil s \rceil_n = s$. Let $\mathbb{V}_{A,B}$ denote the full subcategory of $\mathbb{P}_{A,B}$ spanning HON-views.

▶ **Proposition 10.** *A HON-play $s = (n, f, \varphi)$ is a HON-view if and only if for all odd $i \in n$, $\varphi(i) = i - 1$.*

The embedding $\mathsf{i}_{HON} : \mathbb{V}_{A,B} \hookrightarrow \mathbb{P}_{A,B}$ induces an adjunction $\begin{tikzcd} \widehat{\mathbb{P}_{A,B}} \arrow[r, bend left, "\Delta_{\mathsf{i}_{HON}}"] \arrow[r, phantom, "\perp"] & \widehat{\mathbb{V}_{A,B}} \arrow[l, bend left, "\Pi_{\mathsf{i}_{HON}}"] \end{tikzcd}$.

▶ **Definition 11.** We call $\widehat{\mathbb{V}_{A,B}}$ the category of *TO-behaviours*. We denote by $\mathrm{Sh}(\mathbb{P}_{A,B})$ [1] the category of *innocent TO-strategies* on (A, B), which is the essential image of $\prod_{\mathsf{i}_{HON}}$.

By full faithfulness of i_{HON}, $\prod_{\mathsf{i}_{HON}}$ restricts to an equivalence $\mathrm{Sh}(\mathbb{P}_{A,B}) \simeq \widehat{\mathbb{V}_{A,B}}$.

3 Two concurrent models of HON games

In this section, we build two models of HON games, one based on string diagrams, the other on proof trees, and show that they have equivalent categories of plays, views, and strategies.

3.1 String diagrams

The first model we build is in the same spirit as our previous models of CCS and the π-calculus. There are several salient differences between the notions of views and plays in our model and those from standard game semantics: first, they are interpreted *causally*, when the standard ones are interpreted *temporally* (i.e., our notion of play only retains causal dependency between moves, and there is no fixed order between moves that are independent from one another); secondly, they are intrinsically *multi-party* and put a strong focus on the notion of *position*, while standard HON games are two-player games, and the theory is devoid of the notion of position (though it exists in other settings [2]).

Let us first give an intuition of how to build models in this setting. The idea is to start from a sequent calculus that is an operational description of the calculus we study, and to build a multi-player game from it. Sequents show what positions should be, and derivation rules show what moves players are allowed to play. Plays are then simply sequences of moves, up to permutation of independent moves. While this sounds like it involves a cumbersome quotient of sequences of moves, the formal definition based on presheaves does not involve any such quotient. We will later design another model based on proof trees for a sequent calculus, but the the sequent calculus we introduce right now is just a tool internal to our games and has nothing to do with proof trees. In our case, the sequent calculus that will guide our construction is:

$$\frac{\Gamma, A \cdot m \vdash}{\Gamma \vdash A}\,\Lambda_{(\Gamma \vdash A),m} \qquad \frac{\Gamma, A, \Delta \vdash A \cdot m}{\Gamma, A, \Delta \vdash}\,@_{(\Gamma, A, \Delta \vdash), |\Gamma|+1, m} \qquad \frac{\Gamma \vdash A\Delta, A, \Delta' \vdash}{\Delta, \Gamma, \Delta' \vdash ,}\,\mathrm{CUT} \tag{3}$$

where sequents are lists of arenas, with possibly a distinguished arena, written $(A_1, \ldots, A_n \vdash)$ or $(A_1, \ldots, A_n \vdash A)$. Let Γ range over lists of arenas, $|\Gamma|$ denote the length of Γ, and for all $i \in |\Gamma|$, Γ_i the ith arena of Γ.

[1] As the notation $\mathrm{Sh}(-)$ suggests, this is also a category of *sheaves* for the Grothendieck topology embedding $\mathbb{V}_{A,B}$ into $\mathbb{P}_{A,B}$, though this fact is not used in this paper.

▶ Remark. The sequent calculus we use here does not make much sense from a logical point of view. It however makes sense when seen in relationship with the following sequent calculus:

$$\frac{\dots \quad \Gamma, A\cdot m \vdash \quad \dots \quad (\forall m \in \sqrt{A})}{\Gamma \vdash A}\ \text{RIGHT} \qquad \frac{\Gamma, A, \Delta \vdash A\cdot m}{\Gamma, A, \Delta \vdash}\ \text{LEFT} \qquad \frac{\Gamma \vdash A\Delta, A, \Delta' \vdash}{\Delta, \Gamma, \Delta' \vdash .}\ \text{CUT} \tag{4}$$

Note that this sequent calculus is a fragment of intuitionistic logic when an arena $A = \sum_{i\in n} m_i.A_i$ is interpreted as $[\![A]\!] = \bigwedge_{i\in n} (\neg [\![A_i]\!])$. Proofs in this calculus therefore correspond to proofs of intuitionistic logic. The objects of interest in calculus (3) are partial proofs (proofs whose branches may be left unfinished), which correspond to *explorations* of proofs in calculus (4).

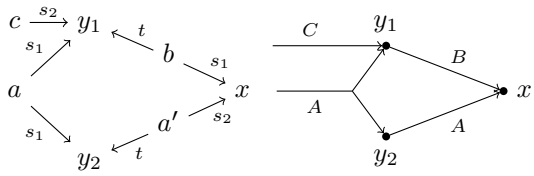
The notion of position we have given in the introduction is a simplification of the real notion of position. In our approach, positions are some kind of graphs. We call their vertices *players* and their edges *channels*. The idea is that players correspond to placeholders for program fragments, and thus for proofs of a certain type, while channels are the way players use to communicate with other players during a play. In the case of HON games, our game is based on arenas, and so is our sequent calculus. Positions will thus be some kind of graphs, whose vertices are labelled by such sequents, and whose edges are labelled by arenas. Players labelled $(\Gamma \vdash A)$ are linked to $|\Gamma|$ incoming channels of type Γ_i and to one outgoing channel of type A, and similarly for players labelled $(\Gamma \vdash)$. The discussion at the end of Example 13 gives more intuition on this notion. Positions are formally represented as presheaves over the following base category:

▶ **Definition 12.** Let $\mathbb{L}_1$ be the category with objects all arenas and sequents, and morphisms $s_i\colon \Gamma_i \to (\Gamma \vdash)$, $s_i\colon \Gamma_i \to (\Gamma \vdash A)$, and $t\colon A \to (\Gamma \vdash A)$.

An object of $\widehat{\mathbb{L}_1}$ is exactly a set of channels labelled A for each arena A, as well as a set of players labelled S for each sequent S, and maps mapping players to their incoming and outgoing channels. A natural transformation $X \to Y$ between such presheaves is akin to a graph morphism, in the sense that it sends each player labelled S in X to some player labelled S in Y (and similarly for channels), while preserving incoming and outgoing channels.

▶ **Example 13.** It is often helpful to represent positions (and more generally plays) graphically. We here give an example of a position given by a presheaf X, which we describe in mathematical terms on the left below (we only give the different sets that compose the presheaf, the functions can be inferred from the category of elements). We also draw its category of elements to the side, and finally how we draw this particular position. Note that the drawing of the position reflects exactly the structure of the presheaf, since it is just another representation of the category of elements (except for the order of the elements of the lists composing the sequents).

- $X(A) = \{a, a'\}$, $X(B) = \{b\}$, $X(C) = \{c\}$,
- $X(A, C \vdash B) = \{y_1\}$, $X(A \vdash A) = \{y_2\}$, $X(B, A \vdash) = \{x\}$,
- $X(-)$ empty otherwise.



In this particular position, there are three players x, y_1 and y_2, and four channels on which they may communicate. There is no need to depict positive and negative players differently

anymore: negative players have an outgoing channel, while positive ones don't. We see two interesting phenomena on this position: there is an input channel shared between y_1 and y_2, and some channels have an open end. In terms of semantics, the first phenomenon corresponds to the fact that several players may use a given proof of some formula A, while the second one corresponds to the fact that some resources may be given by the environment, instead of another player.

We now want to express the dynamics of the game. We start by an informal description in terms of a sequent calculus before augmenting $\mathbb{L}_1$ into a category $\mathbb{L}$ that will represent this dynamics. Our sequent calculus again guides us to desing this dynamics. The CUT rule shows when two players may interact: when the first player's outgoing channel is equal to one of the second player's incoming channels. The interaction between players is governed by the shapes of the Λ and @ rules. In other words, two players can interact according to the Λ and @ rules if and only if they are linked by a CUT rule. Now we just have to show the "shape" of this interaction.

As prescribed by Curry-Howard, we see this interaction as a step in a form of proof of a certain formula. We write $A = \sum_{i \in n} m_i.A_i$ for the arena shared by the two players. Let x by the negative player, labelled $(\Gamma \vdash A)$, and y the positive one, labelled $(\Delta, A, \Delta' \vdash)$. x should provide a proof of $[\![A]\!]$, while y should require one to prove a contradiction. When they interact, y chooses some $i \in n$ and asks x for a proof of $\neg [\![A_i]\!]$, i.e., y now provides a proof of $[\![A_i]\!]$ which x may inspect to prove a contradiction. Therefore, y turns into $(\Delta, A, \Delta' \vdash A_i)$, while x turns into $(\Gamma, A_i \vdash)$.

Moreover, x should not only change into a proof of contradiction assuming $[\![A_i]\!]$, but since y may inspect the proof of $[\![A]\!]$ again, the original x player should also remain in the final position.

In order to model those moves, we augment the base category $\mathbb{L}_1$:

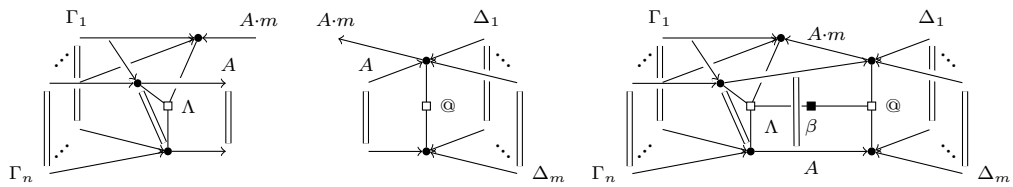
▶ **Definition 14.** Let $\mathbb{L}$ be the category $\mathbb{L}_1$ augmented with an object named after all Λ and @ rules in (3), as well as an object $\beta_{(\Gamma\vdash A),(\Delta\vdash),i,m}$ for all Γ, A, Δ, $i \in |\Delta|$ such that $\Delta_i = A$, and $m \in \sqrt{A}$; and with morphisms $t\colon S \to R$ for each sequent S that is the conclusion of a rule R, and $s\colon S \to R$ for each sequent S that is the premise of a rule R, as well as two morphisms $\lambda\colon \Lambda_{S,m} \to \beta_{S,S',i,m}$ and $@\colon @_{S',i,m} \to \beta_{S,S',i,m}$ for all S, S', i, and m. These morphisms are quotiented by:

$$
\begin{array}{ll}
t \cdot s_i = s \cdot s_i & \text{for all rules } R \text{ and } i \text{ such that } s_i \text{ is defined,} \\
\lambda \cdot t \cdot t = @ \cdot t \cdot s_i & \text{for all } \beta_{S,S',i,m}, \\
\lambda \cdot s \cdot s_{|\Gamma|+1} = @ \cdot s \cdot t & \text{for all } \beta_{(\Gamma\vdash A),S',i,m}.
\end{array}
$$

In simple terms, we add an object for each possible basic shape of move in the game: objects Λ and @ for the rules they respectively represent, and objects β for each possible interaction of two rules.

This is however not sufficient to model moves. Indeed, moves have initial and final positions. To take this fact into account, we model moves not only by a representable, but by a cospan $Y \to \mathsf{y}_m \leftarrow X$, where X and Y are positions (i.e., presheaves empty except over $\mathbb{L}_1$), with X the initial position of m and Y its final position.

▶ **Definition 15.** The *seeds* corresponding to the representables $\Lambda_{(\Gamma\vdash A),m}$, $@_{(\Delta\vdash),i,m}$, and $\beta_{(\Gamma\vdash A),(\Delta\vdash),i,m}$ are the cospans represented graphically below.



Each drawing is the representation of the category of elements of the corresponding representable, with its initial position at the bottom and its final one on top, and where the squares correspond to the elements of type Λ, @, and β respectively (see the long version of this article [4] for formal definitions).

Apart from the existence of channels, there is a single difference between the moves we have shown in the introduction and the moves shown above: the β moves, which correspond to synchronisation. However, it can be shown that β moves can never be played in plays over $(A \vdash B)$, so we may treat them as non-existent. We still show them because they arise naturally in our game and they are necessary to model plays on more complex positions.

Once we have defined seeds, we reuse the same machinery as for CCS and the π-calculus to produce notions of plays and views, which we organise into categories. Since the description is long and technical, and of interest only to some readers, we here content with the following informal sketch. In seeds, all of the position participates in the move: a *move* is any cospan obtained by gluing (by pushout in $\widehat{\mathbb{L}}$) some seed and some position. In any categories with pushouts, cospans form the morphisms of a well-known bicategory. A *play* is any cospan isomorphic to some finite composite of moves in that bicategory.

▶ **Example 16.** Step 4 in Figure 1 produces a composite of six moves, whose gluing does not enforce any particular ordering between f_r and, say f_l.

Plays over any position X are the objects of a category $\mathbb{E}(X)$, whose morphisms are injective morphisms between underlying presheaves, preserving the initial position. A *view* is then merely a "non-branching" play of positive, even length. Views over X form a full subcategory $\mathbb{E}^{\mathbb{V}}(X)$ of $\mathbb{E}(X)$.

▶ **Lemma 17.** *For any sequent S, $\mathbb{E}(S)$ (resp. $\mathbb{E}^{\mathbb{V}}(S)$) is equivalent to a subcategory of $S/\widehat{\mathbb{L}}$ spanning morphisms $S \to U$ such that there exists a play (resp. view) $Y \to U \leftarrow X$.*

In order to define the different notions of strategies and link them to Tsukada and Ong's work, we restrict our attention to positions containing a single player $(A \vdash B)$ and both its channels, for any pair of arenas (A, B).

▶ **Definition 18.** A *behaviour* is a presheaf over $\mathbb{E}^{\mathbb{V}}(A \vdash B)$. A *strategy* is a presheaf over $\mathbb{E}(A \vdash B)$. An *innocent strategy* is a strategy in the essential image of $\prod_{\mathsf{i}}$.

3.2 Proof trees

In the long version of this paper [4], there is a direct proof of the correspondence between the notions of play and view in both approaches. However, that proof is technical and hardly illuminating. We here present another, more intuitive proof based on a simpler model that is equivalent to the one based on string diagrams in the particular case of plays whose initial position consists of one player $(A \vdash B)$.

This new model is based on proof trees in the following, *ad hoc* sequent calculus:

$$\frac{\dots \quad \Gamma, A \cdot m(i) \vdash \quad \dots \quad (\forall i \in n)}{\Gamma \vdash A}\ \text{RIGHT} \qquad \frac{\Gamma, A, \Delta \vdash A \cdot m}{\Gamma, A, \Delta \vdash}\ \text{LEFT} \tag{5}$$

where the RIGHT rule can be applied with any $m{:}n \to \sqrt{A}$ for a positive n and the LEFT rule can be applied with any $m \in \sqrt{A}$. Notice how, apart from the absence of a CUT rule, the sequent calculus (4) we have introduced at the beginning differs only slightly from this one: there, in the RIGHT rule, m must always be a bijection. Moreover, while the objects of interest are "complete" trees in the first case, the objects of interest here are "incomplete" trees. This once again corresponds to the fact that we are interested in explorations of proofs, and not proofs themselves. Therefore, some branches of the proof may not be explored (hence the fact that m need not be surjective) and others may be explored more than once (hence m need not be injective).

An S-*tree* is a partial proof tree (i.e., a proof tree whose branches may be left unfinished) whose conclusion is S. We define $\mathbb{T}(S)$ to be the category of S-trees and morphisms of such, where morphisms of S-trees are inclusions, both in width and depth, of S-trees, and are defined inductively by:

- the empty S-tree has exactly one morphism to all other S-trees,
- the set of morphisms from $\dfrac{T_1 \quad \dots \quad T_n}{\Gamma \vdash A}$ to $\dfrac{T'_1 \quad \dots \quad T'_m}{\Gamma \vdash A}$ is the disjoint union, for all injective $g{:}n \to m$, of $\prod_{i \in n} \mathbb{T}(\Gamma, A \cdot m_i \vdash)(T_i, T'_{g(i)})$,
- the set of morphisms from $\dfrac{T}{\Gamma, A, \Delta \vdash}$ to $\dfrac{T'}{\Gamma, A, \Delta \vdash}$ is $\mathbb{T}(\Gamma, A, \Delta \vdash A \cdot m)(T, T')$.

Notice that morphisms treat the premises of a $(\Gamma \vdash A)$ node as if they were unordered.

▶ **Definition 19.** An S-*branch* is a non-branching S-tree (i.e., an S-tree whose RIGHT rules are all unary) of positive, even depth. Let $\mathbb{B}(S)$ be the full subcategory of $\mathbb{T}(S)$ spanning S-branches.

▶ **Example 20.** The tree at the top-right of Figure 1 is an example of $(\mathbb{B}, \mathbb{B})$-tree. Here is an example of $(A \vdash B)$-tree: $\dfrac{A, B \cdot m \vdash \qquad A, B \cdot m \vdash}{A \vdash B}\, m, m$. None of these trees are branches (they both branch at some point). Note that, while the first tree intuitively represents a HON-play, the other one does not, as it lacks alternation: it is as if Opponent had played m twice in a row. This example also shows that morphisms treat premises as if they were unordered, in the sense that there are two morphisms from that tree to itself (one that maps both branches to themselves and one that swaps them).

We will later need this technical result:

▶ **Proposition 21.** $\mathbb{T}(A, B \vdash C)$ *and* $\mathbb{T}(A + B \vdash C)$ *are isomorphic.*

As announced in the introduction:

▶ **Lemma 22.** $\mathbb{T}(A \vdash B)$ *and* $\mathbb{E}(A \vdash B)$ *are equivalent, and so are* $\mathbb{B}(A \vdash B)$ *and* $\mathbb{E}^{\mathbb{V}}(A \vdash B)$*, and these equivalences are such that the following square commutes.*

$$\begin{tikzcd}
\mathbb{B}(A \vdash B) \arrow[r] \arrow[d] & \mathbb{T}(A \vdash B) \arrow[d] \\
\mathbb{E}^{\mathbb{V}}(A \vdash B) \arrow[r] & \mathbb{E}(A \vdash B)
\end{tikzcd}$$

4 The level of views and plays

In the previous section, we have introduced the embedding $\mathbb{E}^{\mathbb{V}}(S) \to \mathbb{E}(S)$ of views into plays based on string diagrams, and shown it equivalent to the embedding $\mathbb{B}(S) \to \mathbb{T}(S)$ based on

partial proof trees. We now want to show in which sense the latter is related to Tsukada and Ong's embedding $\mathsf{i}_{HON}\colon \mathbb{V}_{A,B} \hookrightarrow \mathbb{P}_{A,B}$. Namely, we build a full embedding F from $\mathbb{P}_{A,B}$ to $\mathbb{T}(A \vdash B)$ and then show that it restricts to an equivalence $F^{\mathbb{V}}\colon \mathbb{V}_{A,B} \to \mathbb{B}(A \vdash B)$. This will then yield a further correspondence between categories of strategies.

Let us first build $F(s)$ by induction on s. If s is empty, $F(s)$ is simply the empty $(A \vdash B)$-tree. Otherwise, s can be written as a sum of threads (Definition 8) $s = \sum_{i \in n} t_i$. Now, each thread t_i is of the form $m_i^1 m_i^2 s_i$ for moves m_i^1, m_i^2 and a play s_i, and a slight generalisation of a result at the start of Section 3.5 in [19] gives:

▶ **Proposition 23.** *For all arenas A and B, if $\mathsf{T}_{A,B}$ is the full subcategory of $\mathbb{P}_{A,B}$ spanning threads, $\chi\colon (mm')/\mathsf{T}_{A,B} \to \mathbb{P}_{C,C\cdot m'}, (mm's) \mapsto s$ is an isomorphism, where $C = A + B \cdot m$.*

Each s_i can therefore be mapped to a $(C_i \vdash C_i \cdot m_i^2)$-tree recursively, where $C_i = A + B \cdot m_i^1$. By Proposition 21, this gives an $(A, B \cdot m_i^1 \vdash C_i \cdot m_i^2)$-tree $\widetilde{F}(s_i)$, which can in turn be composed with the LEFT and RIGHT rules to give $F(s)$ as below:

$$\dfrac{\dots \quad \dfrac{\widetilde{F}(s_i)}{A, B \cdot m_i^1 \vdash}\, m = m_i^2 \quad \dots}{A \vdash B.}\, m(i) = m_i^1$$

▶ **Lemma 24.** *F is a full embedding.*

Proof. Injectivity on objects and faithfulness are easy to prove. The proof of fullness is mostly straightforward, except that it requires a reading of the pointers of s inside $F(s)$ to show that the antecedent though F of a morphism of $(A \vdash B)$-trees is indeed a HON-morphism. We explain in details how to read pointers from $F(s)$ below. ◀

First, recovering the moves in s from $F(s)$ is easy: they are all the m's used in any LEFT or RIGHT rule, with the obvious multiplicity. Recovering pointers is a bit trickier and requires to know what an *occurrence* of an arena in a tree is, as well as what it means for a move to *create* such an occurrence, and what it means for a move to be *played on* the occurrence of an arena. Lemmas 142 and 148 from [4] then explain how to recover pointers from our structure of plays. They can be stated as follows: each move m is justified by the move that created the arena occurrence m was played on.

▶ **Definition 25.** Let T be an $(A \vdash B)$-tree. The set of *moves* occurring in T is the disjoint union of all the moves of the rules of T, where, in (3), there is a single move m occurring in the LEFT rule, and the moves that occur in the RIGHT rule are all the $m(i)$'s.

In the LEFT rule, *m plays on A* and *creates $A \cdot m$*. In the RIGHT rule, *$m(i)$ plays on A* and *creates $A \cdot m(i)$*. Additionally, if $\dfrac{\Gamma, A \cdot m(1) \vdash \quad \dots \quad \Gamma, A \cdot m(n) \vdash}{\Gamma \vdash A}$ is the first rule of T, then $m(i)$ creates all the arenas in $\Gamma, A \cdot m(i) \vdash$.

The *occurrences* of an arena in an S-tree T is an arena that is either the negative arena in the conclusion of T (if it exists) or an arena created by a rule in T.

▶ **Example 26.** There are four occurrences of the empty arena $\varnothing$ in the tree of Figure 1, created by t_l, f_l, f_r, and t_r respectively. There are also two occurrences of the boolean arena: the first one exists ($\mathbb{B}_r$) at the beginning, and the second one ($\mathbb{B}_l$) is created by q_r (and shared by all sequents above in the tree). In the tree of Example 20, there are two occurrences of the A arena, created by both m moves. This is an artefact of playing on the arena pair (A, B) rather than on $A \to B$, but this is more natural in our setting (for example, the notion of *interaction sequence*, which is used to study composition of strategies, is the same as that of play, but on particular positions).

In the tree in Figure 1, the q_l move is played on the occurrence of $\mathbb{B}$ called $\mathbb{B}_l$, which is created by q_r, so q_l is justified by q_r. Similarly, f_r is played on the occurrence of $\{\mathsf{t},\mathsf{f}\}$ called $\{\mathsf{t},\mathsf{f}\}_r$, which is also created by q_r, so f_r is also justified by q_r. We thus recover the pointer structure from the P-view tree, i.e., the pointer structure of the play we started from.

▶ **Lemma 27.** *F restricts to a functor $F^{\mathbb{V}}\colon \mathbb{V}_{A,B} \to \mathbb{E}^{\mathbb{V}}(A \vdash B)$.*

Proof. Let us take a HON-view $s = (n, f, \varphi)$, and show that $F(s)$ is a branch. The proof trees of our sequent calculus can only branch with the use of a RIGHT rule. In that case, we get by the method described above to recover pointers that two Opponent moves are justified by the same Proponent move. But we know by Proposition 10 that all Opponent moves in s are justified by the preceding move, so there can be no two Opponent moves justified by the same Proponent move. ◀

▶ **Lemma 28.** *$F^{\mathbb{V}}$ is an equivalence of categories.*

Proof. Since i, i_{HON}, and F are fully faithful, $F^{\mathbb{V}}$ is also fully faithful by left cancellation. Now, to show that $F^{\mathbb{V}}$ is essentially surjective on objects, we simply need to build an antecedent through $F^{\mathbb{V}}$ of any view v. The candidate HON-view is given by taking all moves of v from the root to the top (this is unambiguous since v is non-branching) and pointers given by the method described above. All that is left is to verify that the candidate HON-view is indeed a HON-view, which is done by verifying that it is a HON-play and that all Opponent moves are justified by the preceding move. The first point is easy, since the way we have chosen the antecedent may be generalised to any play, and the antecedent verifies all properties of HON-plays, except perhaps for alternation and having even length, which are both trivial in our case because views do not branch and have even depth. The second point is obvious by construction. ◀

▶ Remark. The proof above gives some insight on the only fundamental difference between our plays and HON-plays: ours only verify a weak form of alternation, where Opponent may play several moves in a row, but Proponent may only answer once per Opponent move.

By putting everything together, we get:

▶ **Theorem 29.** *The square* (1) *commutes, F is a full embedding, and $F^{\mathbb{V}}$ is an equivalence of categories.*

5 The level of strategies

A useful tool to compare strategies and behaviours in both settings is Guitart's theory of *exact squares* [8], which we now recall.

▶ **Definition 30.** A *square* is a natural transformation as on the left of (6).

Any square yields by restriction a square as in the middle below, and so by adjunction a further square as on the right:

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "u"'] \arrow[dr, phantom, "\Rightarrow\phi"] & B \arrow[d, "v"] \\ C \arrow[r, "g"'] & D \end{tikzcd} \qquad \begin{tikzcd} \widehat{A} & \widehat{B} \arrow[l, "\Delta_f"'] \arrow[dl, phantom, "\Leftarrow\Delta_\phi"] \\ \widehat{C} \arrow[u, "\Delta_u"] & \widehat{D} \arrow[l, "\Delta_g"] \arrow[u, "\Delta_v"'] \end{tikzcd} \qquad \begin{tikzcd} \widehat{A} \arrow[r, "\Pi_f"] & \widehat{B} \arrow[dl, phantom, "\Leftarrow\tilde{\phi}"] \\ \widehat{C} \arrow[u, "\Delta_u"] \arrow[r, "\Pi_g"'] & \widehat{D}. \arrow[u, "\Delta_v"'] \end{tikzcd} \tag{6}$$

▶ **Definition 31.** A square ϕ is *exact* when $\tilde{\phi}$ is an isomorphism.

The result we ultimately want to prove to show the tight relationship between both approaches is that the square (2) commutes up to isomorphism. This corollary reduces to exactness of (1) filled with the identity.

▶ **Lemma 32.** *Any square as on the left of* (6) *in which* ϕ *is an identity,* u *is an equivalence, and* v *is fully faithful is exact.*

▶ **Corollary 33.** *The square* (2) *commutes up to isomorphism.*

6 Conclusion

Even though Tsukada and Ong's approach differs significantly from ours in terms of presentation, both approaches define similar notions of play (even though our notion of play is slightly looser than the one from traditional game semantics) and views, and the resulting notions of behaviours and innocent strategies are related in a very strong way. This shows that the differences between the two approaches are mainly choices of presentation.

However, Tsukada and Ong's approach goes further than ours: they define a cartesian closed category of arenas and strategies, which allows them to compose strategies. Two steps are required to compose strategies, called parallel composition and hiding: the first executes two strategies in parallel, and the second one hides the middle arena. While parallel composition is easy to manipulate in our setting because our game is intrinsically multi-party, hiding could admittedly be more difficult to handle.

References

1. Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. Full abstraction for PCF. *Information and Computation*, 163(2):409–470, 2000. doi:10.1006/inco.2000.2930.
2. Samson Abramsky and Paul-André Melliès. Concurrent games and full completeness. In *LICS*, pages 431–442. IEEE, 1999. doi:10.1109/LICS.1999.782638.
3. Pierre Boudes. Thick subtrees, games and experiments. In *TLCA*, volume 5608 of *Lecture Notes in Computer Science*, pages 65–79. Springer, 2009. doi:10.1007/978-3-642-02273-9_7.
4. Clovis Eberhart and Tom Hirschowitz. Justified sequences in string diagrams: a comparison between two approaches to concurrent game semantics. Preprint, 2016. URL: https://hal.archives-ouvertes.fr/hal-01372582.
5. Clovis Eberhart and Tom Hirschowitz. Game semantics as a singular functor, and definability as geometric realisation. working paper or preprint, 2017. URL: https://hal.archives-ouvertes.fr/hal-01527171.
6. Clovis Eberhart, Tom Hirschowitz, and Thomas Seiller. An intensionally fully-abstract sheaf model for pi. In *CALCO*, volume 35 of *LIPIcs*, pages 86–100. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015. doi:10.4230/LIPIcs.CALCO.2015.86.
7. Dan R. Ghica and Andrzej S. Murawski. Angelic semantics of fine-grained concurrency. In *FoSSaCS*, volume 2987 of *Lecture Notes in Computer Science*, pages 211–225. Springer, 2004. doi:10.1007/978-3-540-24727-2_16.
8. René Guitart. Relations et carrés exacts. *Annales des Sciences Mathématiques du Québec*, 4(2):103–125, 1980.
9. Tom Hirschowitz. Full abstraction for fair testing in CCS. In *CALCO*, volume 8089 of *Lecture Notes in Computer Science*, pages 175–190. Springer, 2013. doi:10.1007/978-3-642-40206-7_14.
10. Tom Hirschowitz. Full abstraction for fair testing in CCS (expanded version). *Logical Methods in Computer Science*, 10(4), 2014. doi:10.2168/LMCS-10(4:2)2014.

11 J. M. E. Hyland and C.-H. Luke Ong. On full abstraction for PCF: I, II, and III. *Information and Computation*, 163(2):285–408, 2000. doi:10.1006/inco.2000.2917.

12 André Joyal, Mogens Nielsen, and Glynn Winskel. Bisimulation and open maps. In *LICS*, pages 418–427. IEEE, 1993. doi:10.1109/LICS.1993.287566.

13 James Laird. Game semantics for higher-order concurrency. In *FSTTCS*, volume 4337 of *Lecture Notes in Computer Science*, pages 417–428. Springer, 2006. doi:10.1007/11944836_38.

14 Paul-André Melliès. Asynchronous games 2: the true concurrency of innocence. In *Proc. 15th International Conference on Concurrency Theory*, volume 3170 of *Lecture Notes in Computer Science*, pages 448–465. Springer, 2004. doi:10.1007/978-3-540-28644-8_29.

15 Paul-André Melliès. Game semantics in string diagrams. In *LICS*, pages 481–490. IEEE, 2012. doi:10.1109/LICS.2012.58.

16 Hanno Nickau. Hereditarily sequential functionals. In *LFCS*, volume 813 of *Lecture Notes in Computer Science*, pages 253–264. Springer, 1994. doi:10.1007/3-540-58140-5_25.

17 M. Nielsen, G. Plotkin, and G. Winskel. Event structures and domains, part 1. *Theoretical Computer Science*, 13:65–108, 1981.

18 Silvain Rideau and Glynn Winskel. Concurrent strategies. In *LICS*, pages 409–418. IEEE, 2011. doi:10.1109/LICS.2011.13.

19 Takeshi Tsukada and C.-H. Luke Ong. Nondeterminism in game semantics via sheaves. In *LICS*. IEEE, 2015.

20 Takeshi Tsukada and C.-H. Luke Ong. Plays as resource terms via non-idempotent intersection types. In *LICS*, pages 237–246. ACM, 2016. doi:10.1145/2933575.2934553.