Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 336 (2018) 189–206

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

On Dialogue Games and Graph Games

Clément Jacq Paul-André Melliès

Institut de Recherche en Informatique Fondamentale (IRIF), CNRS, Université Paris Diderot

Abstract

Dialogue games were introduced by Melliès as an attempt to unify two historical paradigms of game semantics: concrete data structures and arena games. The definition of dialogue games relies on the idea that a move m of an arena game can be decomposed as a pair $m = (\alpha, v)$ consisting of a cell α and of a value v. Consequently, a dialogue game is defined as a quadripartite forest whose nodes are separated into four classes: Opponent cells, Opponent values, Player cells, Player values. Although the translation from arena games to dialogue games is essentially immediate, the relationship between dialogue games and concrete data structures is more intricate. In order to clarify it, we study the relationship between dialogue games and graph games which were introduced by Hyland and Schalk to provide a graph-theoretic account of Berry and Curien's sequential algorithm model. We construct a fully faithful functor from a category of dialogue games to the category of graph games and conflict-free strategies. This leads us to an alternative definition of conflict-free strategies in graph games as balanced and bi-invariant strategies in dialogue games.

Keywords: Game semantics, arena games

Introduction

The notion of dialogue game was introduced by Melliès in [14] (Chapter 4, Section 4) in order to provide a direct description of the formulas A, B of tensorial logic with finite sums, generated by the grammar

$$A, B \quad := \quad \neg A \;\mid\; A \otimes B \;\mid\; 1 \;\mid\; A \oplus B \;\mid\; 0$$

modulo the following equations: associativity and commutativity of sum and tensor, unitality of 0 and of 1, distributivity of the tensor product over finite sums:

$$\begin{array}{lcc}
(assoc) & (A \otimes B) \otimes C = A \otimes (B \otimes C) & (A \oplus B) \oplus C = A \oplus (B \oplus C) \\
(unit) & A \otimes 1 = A & A \oplus 0 = A \\
(dist) & A \otimes (B \oplus C) = (A \otimes B) \oplus (A \otimes C) & A \otimes 0 = 0 \\
(comm) & A \otimes B = B \otimes A & A \oplus B = B \oplus A
\end{array}$$

The key observation inspired by ludics [3] and polarized linear logic [8] is that a finite dialogue game is the same thing as an equivalence class of such formulas. One establishes in this way a coherence theorem which characterizes the free dialogue

https://doi.org/10.1016/j.entcs.2018.03.023
1571-0661/© 2018 Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

category with finite sums generated by a given category $\mathscr{C}$ as a specific category of dialogue games and innocent strategies, see [14] (which refines [13]) for details.

One main novelty of dialogue games is that the moves m of the game are not treated as atomic: they are decomposed as pairs $m = (\alpha, v)$ consisting of a cell α and of a value v. This means that every move m of a dialogue game is identified with the action of "filling" a particular cell α with a particular value v. Accordingly, every cell and every value of a dialogue game is assigned a polarity Player or Opponent. The purpose of an Opponent move $m = (\alpha, v)$ is thus to fill an Opponent cell α with an Opponent value v while the purpose of a Player move $n = (\beta, w)$ is to fill a Player cell β with a Player value w.

The idea of decomposing every move m of a dialogue game into a pair $m = (\alpha, v)$ comes from the following proof-theoretic observation: given a formula A of tensorial logic with sums, there is a one-to-one correspondence between the cells α of the associated dialogue game A, and the tensorial negations $X \mapsto \neg X$ of the canonical form of the formula A. By the canonical form, we mean the formula obtained from A by orienting from left to right the equations $(dist)$ and $(unit)$.

By way of illustration, the boolean game $\mathbb{B}$ is defined as the double negation $\neg\neg(1 \oplus 1)$ of the dialogue game $1 \oplus 1$ with two Player values ***true*** and ***false***. By the correspondence between cells and tensorial negations observed above, the game $\mathbb{B}$ has two cells: the Opponent cell α at the root which corresponds to the external negation, and the Player cell β which corresponds to the internal negation:

$$\mathbb{B} \quad = \quad \overset{\alpha}{\neg}\,\overset{\beta}{\neg}\,(1 \oplus 1)$$

So, Opponent starts the game by playing the move $q = (\alpha, \boldsymbol{q})$ which fills the cell α with the value $\boldsymbol{q}$. This Opponent value $\boldsymbol{q}$ justifies the Player cell β. This enables Player to react to Opponent's move q by playing the move $n = (\beta, v)$ which fills the Player cell β with a value $v =$ ***true*** or $v =$ ***false***.

The terminology of "cells" and "values" was adopted in [14] in order to coincide with the terminology used by Kahn and Plotkin in their work on concrete data structures [7]. This choice of terminology reflects our wish to unify dialogue games with the sequential algorithm model designed by Berry and Curien, also based on concrete data structures, see [1] for details. The purpose of the present paper is precisely to investigate this point. Rather than working with the original formulation of the model, we find convenient to start from Curien and Lamarche's reformulation based on filiform concrete data structures, also called sequential data structures [2]. The comparison with dialogue games brings us to a line of research on graph games developed fifteen years ago by Hyland and Schalk [6,15]. This leads us to compare the three game models below:

- sequential data structures – seen as *simple games* played on trees [2,5,4],
- graph games – played on *balanced graphs* [6,15],
- dialogue games – seen as *asynchronous games* [13,14].

In order to connect dialogue games with graph games, we need to tame the usual asynchronous trajectories of dialogue games, and to restrict them to a specific class of trajectories, called balanced trajectories. These balanced trajectories reflect the implicit "switching conditions" required of the trajectories of sequential data structures and of graph games. As we will see, the constraint is formulated in dialogue games as a positional payoff condition on trajectories, see §3 for details. We construct in this way a dialogue category **DG** of dialogue games and balanced strategies.

One primary observation of the paper is that there exists a pair of equivalence relations $\sim_{OP}$ and $\sim_{PO}$ between balanced trajectories of a dialogue game associated to a formula A of tensorial logic, see §4 for details. This pair of equivalence relations is inspired by the Opponent-Player and Player-Opponent permutations $\sim_{OP}$ and $\sim_{PO}$ used in [12] to describe innocent strategies and counter-strategies in asynchronous games. Somewhat surprisingly, we establish (Prop. 4.4) that the two equivalences $\sim_{OP}$ and $\sim_{PO}$ are generated by a number of atomic permutation tiles of the form $s \diamond_{OP} t$ and $s \diamond_{PO} t$. The permutation tiles $s \diamond_{OP} t$ are of a particularly simple form, since they permute the order of two Opponent moves m_1, m_2 and two Player moves n_1, n_2 in a sequence of moves

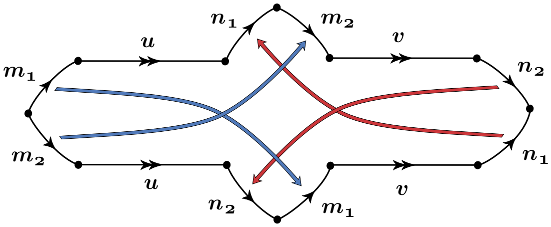
$$s \;=\; m_1 \cdot u \cdot n_1 \cdot m_2 \cdot v \cdot n_2 \quad \diamond_{OP} \quad t \;=\; m_2 \cdot u \cdot n_2 \cdot m_1 \cdot v \cdot n_1$$

of even length, where u, v are even-length sequences of moves. Similarly, the permutation tiles $s \diamond_{PO} t$ always permute the order of two Opponent moves m_1, m_2 and two Player moves n_1, n_2 in a sequence of moves

$$s \;=\; n_1 \cdot u \cdot m_1 \cdot n_2 \cdot v \cdot m_2 \quad \diamond_{PO} \quad t \;=\; n_2 \cdot u \cdot m_2 \cdot n_1 \cdot v \cdot m_1$$

of even length, where u, v are even-length sequences of moves. Typically, every permutation tile $s \diamond_{OP} t$ is depicted in the following way in the dialogue game:

(1)



One thus recovers at the heart of the theory of sequential algorithms two classes of permutation tiles $\diamond_{OP}$ and $\diamond_{PO}$ similar in spirit to the 2-dimensional tiles encountered in the asynchronous theory of innocence [12,14]. The key difference is that the two sequences of moves u, v are empty in the asynchronous definition of innocence, and the OP-permutation tiles are of the form

$$s = m_1 \cdot n_1 \cdot m_2 \cdot n_2 \quad \diamond_{OP} \quad m_2 \cdot n_2 \cdot m_1 \cdot n_1 = t.$$

One benefit of shifting from graph games to dialogue games is that the same sequence of moves u describes two different trajectories in the atomic tile diagram (1), and similarly for the sequence of moves v. For that reason, two balanced trajectories s and t related by an atomic permutation tile $s \diamond_{OP} t$ or by an atomic permutation tile $s \diamond_{PO} t$ play exactly the same moves in the dialogue game, but in a different order. In other words, the two trajectories s, t are equivalent modulo the usual permutation relation $\sim$ of dialogue games, see [12,13,14].

In order to clarify the relationship between dialogue games and graph games, we define a notion of conflict-free strategy in dialogue games, and construct a functor of dialogue categories $G : \mathbf{DG}_{\mathsf{bal}} \longrightarrow \mathbf{GG}$ from the category $\mathbf{DG}_{\mathsf{bal}}$ of rooted dialogue games and balanced conflict-free strategies, to the category $\mathbf{GG}$ of graph games defined in [6,15]. One main result of the paper is that the functor G is fully faithful. This enables us to see every graph game $G(A)$ of the tensorial hierarchy (that is, generated by tensor, sum and negation) as a graph "embedded" in the dialogue game A. This result sheds light on the combinatorial structure of Hyland-Schalk graph games, and the fact that they are secretly regulated by the atomic permutation tiles $\diamond_{OP}$ and $\diamond_{PO}$.

Plan of the paper. After recalling in §1 the definition of dialogue games and in §2 their relationship with tensorial logic, we construct in §3 a dialogue category **DG** of dialogue games and balanced strategies. We analyse in §4 the two permutations $\sim_{OP}$ and $\sim_{PO}$ between trajectories in a dialogue game, and show that they are generated by simple permutation tiles $\diamond_{OP}$ and $\diamond_{PO}$. After reintroducing in §5 the definitions of graph games and pre strategies of Hyland-Schalk [6], we construct in §6 a category $\mathbf{DG}_{\mathsf{bal}}$ of rooted dialogue games and balanced conflict-free strategies, and establish that it embeds fully faithfully (as a dialogue category) into the category **GG** of graph games. We conclude by establishing a bi-invariance theorem in §7.

1 Dialogue games

In this section, we recall the definition of dialogue game formulated in [14].

Definition 1.1 A *rooted dialogue game* is a finite bipartite rooted tree whose nodes are separated in two sets **Cells** and **Values**. We note $\triangleright$ the relation between parent nodes and children nodes. By bipartite tree, we mean that

$$\triangleright \quad \subseteq \quad \textbf{Cells} \times \textbf{Values} \; + \; \textbf{Values} \times \textbf{Cells}.$$

This means that the children of a value are cells and that the children of a cell are values. A rooted dialogue game is equipped with a polarity function

$$\lambda : \textbf{Cells} \uplus \textbf{Values} \to \{-1, 1\}$$

such that, for α a cell and v a value, we have :

$$\alpha \triangleright v \Rightarrow \lambda(\alpha) = \lambda(v) \qquad\qquad v \triangleright \alpha \Rightarrow \lambda(\alpha) = -\lambda(v)$$

Finally, the root of a rooted dialogue game is a value of polarity $+1$.

By convention, we call *Player* a node with a positive polarity and *Opponent* a node with a negative one. We can say that a rooted dialogue game is quadripartite in the sense that its branches alternate in the following way :

Player value $\triangleright$ Opponent cell $\triangleright$ Opponent value $\triangleright$ Player cell $\triangleright$ Player value ...

Definition 1.2 A *dialogue game* A is a family $(A_i \mid i \in I)$ of rooted dialogue games indexed by a finite set I. It can be seen as a forest whose connected components are the rooted dialogue games A_i. We write $\mathbf{Roots}_A$ for the set of roots of A.

Note that every root $* \in \mathbf{Roots}_A$ of a dialogue game $A = (A_i \mid i \in I)$ is a Player value, and that there is a one-to-one correspondence between $\mathbf{Roots}_A$ and the set I of rooted components of A. The basic intuition here is that we see Dialogue Games as a game semantics expansion of Berry's and Curien's Concrete Data Structures where we symmetrize these structures by assigning cells and values to different players.
Next, we introduce the relations used to simulate the idea that a Cell can only be filled by one Value at a time in a play of the game.
We take the usual order relation on nodes of a rooted tree, and write $a \leq b$ when a is an ancestor of b, and $a \wedge b$ for the greatest common ancestor of a and b. We define the following notion of compatibility:

Definition 1.3 Two nodes a and b are called *compatible* when $a \wedge b$ is a value. They are called incompatible otherwise.

Intuitively, compatible nodes stand for concurrent choices, where we can choose a node, then backtrack and try another one, whereas incompatible nodes stand for a definitive choice: if we pick a node, the other branches are forever lost in the current exploration.

Definition 1.4 A *position* of a rooted dialogue game A is a non-empty downward-closed set x of pairwise compatible values of A:

- $\forall v, w \in \mathbf{Values}, \quad v \leq w \,\text{and}\, w \in x \;\Rightarrow\; v \in x$
- $\forall v, w \in \mathbf{Values}, \quad v \in x$ and $w \in x \;\Rightarrow\; v$ compatible with w

A position of a dialogue game $A = (A_i \,|\, i \in I)$ is a non-empty downward-closed set w of pairwise compatible values of the forest A, living in exactly one rooted component A_i of the dialogue game. We write $\mathbf{Pos}(A)$ for the set of positions of a dialogue game A.

Definition 1.5 A *move* of a rooted dialogue game A is a pair (α, v) consisting of a cell α and of a value v such that $\alpha \triangleright v$. We write $\mathbf{Moves}_A$ for the set of moves of A.

Definition 1.6 The *positional graph* $\mathbf{Graph}_A$ of a rooted dialogue game A is a graph whose vertices are the positions of A and whose edges are the moves of A, such that, for two positions x and y and a move (α, v), we have $(\alpha, v) : x \to y$ if and only if $y = x \uplus \{v\}$. The positional graph $\mathbf{Graph}_A$ of a dialogue game $A = (A_i \mid i \in I)$ is the disjoint sum of the graphs of the rooted dialogue games A_i.

Definition 1.7 A *trajectory* $s : x \twoheadrightarrow y$ of a rooted dialogue game A is a path between positions x and y in $\mathbf{Graph}_A$. It can thus be seen as a sequence of moves of A. A trajectory $s : * \twoheadrightarrow x$ starting from the root $*$ of a rooted dialogue game A and alternating between Opponent and Player moves is called a *play* of that game. We write $\mathbf{Plays}_A$ for the set of plays of A.

Definition 1.8 A Player *strategy* σ of a rooted dialogue game A is a set of plays of even length such that:

- σ contains the empty play,
- σ is closed by even-length prefix, in the sense that

$$\forall s \in \mathbf{Plays}_A, \forall m, n \in \mathbf{Moves}_A, \qquad s \cdot m \cdot n \in \sigma \quad \Rightarrow \quad s \in \sigma,$$

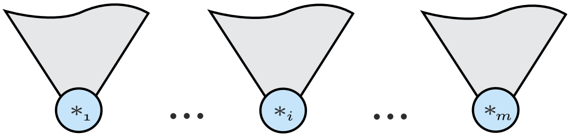
- σ is deterministic, in the sense that

$$\forall s \in \mathbf{Plays}_A, \forall m, n_1, n_2 \in \mathbf{Moves}_A,\ s \cdot m \cdot n_1 \in \sigma, s \cdot m \cdot n_2 \in \sigma \Rightarrow n_1 = n_2.$$

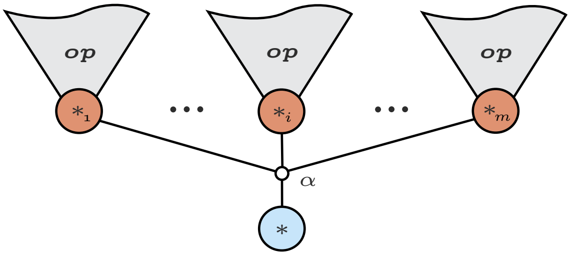
2 Dialogue games and tensorial logic

As explained in the introduction, the notion of dialogue game has been introduced in [14] (as a revision of [13]) because of its one-to-one correspondence with the formulas of tensorial logic. We explain now the correspondence between dialogue games and these formulas and formulate at the end of section, the notion of transverse strategies which will enable us to construct our categories of dialogue games.

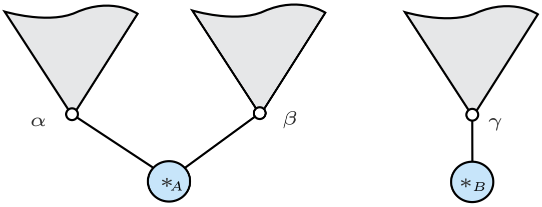
First of all, the dialogue game 0 is defined as an empty family of rooted dialogue games ; while the rooted dialogue game 1 is defined as the game with no cell and a unique value $*$ which defines its root. The sum $A \oplus B$ of two dialogue games $A = (A_i \mid i \in I)$ and $B = (B_j \mid j \in J)$ is defined as the disjoint union $A \oplus B = (C_k \mid k \in I \uplus J)$ where $C_i = A_i$ for $i \in I$ and $C_j = B_j$ for $j \in J$. Note that summing is associative, and that every dialogue game $A = (A_i \mid i \in I)$ is the sum of a finite number of rooted dialogue games A_i. Note also that two nodes coming from different games of the sum are necessarily incompatible, as they share no common ancestor. a play in $A \oplus B$ is thus either a play in A or a play in B. The rooted dialogue game $\neg A$ associated to a dialogue game $A = (A_i \mid i \in I)$ is defined by reversing the polarities of each of the rooted dialogue games of A_i, and then adding a root value $*$ which justifies a cell α which justifies each root $*_i$ of the rooted dialogue game A_i. Thus negation transforms this dialogue game:



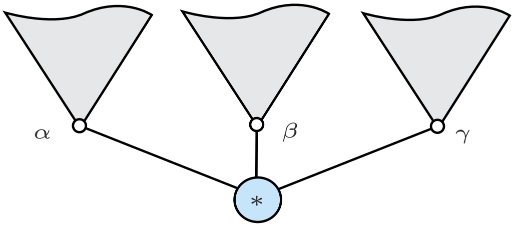
into the rooted dialogue game:



where we depict Player values as light blue disks, Opponent values as dark red disks, and cells as white nodes. The tensor product of two rooted dialogue games A and B is the rooted dialogue game defined by merging the root values $*_A$ and $*_B$ into a single root value $*$. By way of illustration, the two rooted dialogue games



are turned into the tensor product $A \otimes B$ by the following operation:



The tensor product of two dialogue games $A = (A_i \mid i \in I)$ and $B = (B_j \mid j \in J)$ is then defined as the dialogue game $A \otimes B = (A_i \otimes B_j \mid (i,j) \in I \times J)$ with components the rooted dialogue games $A_i \otimes B_j$ just defined. We'll occasionally note $x \otimes y$ for positions of $A \otimes B$ when we need to separate the values coming from A from those coming from B. The correspondence between dialogue games and formulas of tensorial logic comes from the fact that every dialogue game $A = (A_i \mid i \in I)$ can be decomposed into the sum of the rooted game A_i, and that every rooted game A_i is the tensor product of a finite number of negated dialogue games $\neg B_{ij}$.

Every pair of dialogue games A and B defines a rooted dialogue game $A \multimap\!\!\bullet\, B$ by the equation:

$$A \multimap\!\!\bullet\, B \quad = \quad \overset{\alpha}{\neg}\,(\, A \otimes \overset{\beta}{\neg}\, B\,)$$

where we tag the two tensorial negations with α and β so as to indicate the names of the associated cells. In order to define the various categories of dialogue games, we will make a fundamental usage of the notion of *transverse* strategy on the dialogue game $A \mathrel{-\!\!\bullet} B$.

Definition 2.1 A strategy on $A \mathrel{-\!\!\bullet} B$ is called *transverse* when

$$\forall val \in \mathbf{Roots}_A, \exists wal \in \mathbf{Roots}_B, \qquad (\alpha, val) \cdot (\beta, wal) \in \sigma.$$

So, a strategy σ of $A \mathrel{-\!\!\bullet} B$ is transverse when it reacts to every initial move (α, val) played by Opponent, for a value $val \in \mathbf{Roots}_A$, by a move of the form (β, wal) for a value $wal \in \mathbf{Roots}_B$ played in the component $\neg B$ of the dialogue game $A \mathrel{-\!\!\bullet} B = \neg (A \otimes \neg B)$ instead of the component A.

3 Dialogue games and balanced strategies

In this section, we introduce the notions of balanced position and of balanced trajectory in a dialogue game. We describe in §3.1 the payoff condition on positions which enforces the appropriate "switching conditions" on balanced plays and trajectories. We then explain in §3.2 how to construct a dialogue category **DG** with dialogue games as objects and balanced strategies as morphisms.

3.1 Balanced positions and trajectories

One main discrepancy between dialogue games and sequential data structures (or simple games) relies on the way trajectories are defined in both frameworks. In particular, the global definition of a play s in a dialogue game $A \otimes B$ implies that its restrictions $s_{|A}$ and $s_{|B}$ to the subcomponents A and B are not necessarily alternating between Opponent and Player anymore. A typical illustration is provided by the play s of the dialogue game

$$(\mathbb{B} \otimes \mathbb{B}) \mathrel{-\!\!\bullet} (\mathbb{B} \otimes \mathbb{B}) \quad = \quad \neg (\mathbb{B} \otimes \mathbb{B} \otimes \neg (\mathbb{B} \otimes \mathbb{B}))$$

characterized by its sequence of four moves below:

	$\mathbb{B}$	$\otimes$	$\mathbb{B}$	$\mathrel{-\!\!\bullet}$	$\mathbb{B}$	$\otimes$	$\mathbb{B}$
first move by O							q
second move by P			q				
third move by O					q		
fourth move by P	q						

This trajectory is interesting because it is played by the copy-cat strategy $id_{\mathbb{B} \otimes \mathbb{B}}$ associated to the dialogue game $\mathbb{B} \otimes \mathbb{B}$ in the category of dialogue games and innocent strategies formulated in [13,14]. In order to forbid that kind of behavior, we introduce the notion of balanced position, and of balanced trajectory. A simple and concise way inspired by [11,10] to define them is to equip every dialogue game A with a *payoff function*

$$\kappa : \mathbf{Pos}(A) \to \{+1, -1, \mathbf{fail}\}$$

The payoff assigns to every position of the dialogue game a value $+1$, -1 or **fail**. The idea is that every position with payoff $+1$ is winning for Player, and Opponent is thus entitled to play a move from it ; symmetrically, a position with payoff -1 is winning for Opponent, and Player is thus entitled to play a move from it. Positions with payoff **fail** are unbalanced in the sense that they cannot be reached by any balanced play.

Definition 3.1 The *payoff* function $\kappa : \mathbf{Pos}(A) \to \{+1, -1, \mathbf{fail}\}$ associated to a dialogue game A is defined by structural induction on the underlying formula A of tensorial logic:

- The single root of the dialogue game 1 has a payoff of $+1$.
- The payoff of a position x of the dialogue game $\neg A$ is $+1$ when the position is at the root, and the reverse of the payoff of the position in the dialogue game A otherwise,
- The payoff of a position in the component A or B of the dialogue game $A \oplus B$ is its payoff in its component,
- The payoff of a position $x \otimes y$ of the tensor product $A \otimes B$ is computed using the following payoff table:

	$\kappa(y) = +1$	$\kappa(y) = -1$	$\kappa(y) = \mathbf{fail}$
$\kappa(x) = +1$	$+1$	-1	**fail**
$\kappa(x) = -1$	-1	**fail**	**fail**
$\kappa(x) = \mathbf{fail}$	**fail**	**fail**	**fail**

A position $x \otimes y$ in the dialogue game $A \otimes B$ is thus unbalanced when the two positions x and y have negative payoffs $\kappa(x) = \kappa(y) = -1$. The reason is that a play $s : * \twoheadrightarrow x \otimes y$ which reaches such a position $x \otimes y$ would need to play at some point two Opponent moves in a row. Note also that the root of a dialogue game A has always positive payoff $+1$. Finally, the reverse of a payoff in $\{+1, -1, \mathbf{fail}\}$ is defined as expected, with the reverse of **fail** defined as itself.

This definition induces a payoff table for the dialogue game $A \multimap B$ which computes the payoff of a position $x \multimap y$ of the dialogue game $A \multimap B$, given the payoff $\kappa(x)$ of the position x in A and the payoff $\kappa(y)$ of the position y in B:

	$\kappa(y) = +1$	$\kappa(y) = -1$	$\kappa(y) = \mathbf{fail}$
$\kappa(x) = +1$	$+1$	-1	**fail**
$\kappa(x) = -1$	**fail**	$+1$	**fail**
$\kappa(x) = \mathbf{fail}$	**fail**	**fail**	**fail**

Definition 3.2 A trajectory $s : x_1 \to x_2 \to \cdots \to x_n$ of a dialogue game A is called *balanced* when all the positions x_i are balanced, for $1 \leq i \leq n$.

Definition 3.3 A *balanced* play $s : * \twoheadrightarrow x$ of a dialogue game A is a balanced trajectory starting from a root $*$ of the game. We write $\mathbf{Bal}_A$ for the set of balanced plays.

We establish the following property which characterizes the balanced plays among the general asynchronous plays of the dialogue game A:

Proposition 3.4 *A play $s : * \twoheadrightarrow x$ is balanced in a rooted dialogue game A if and only if its restriction to every subcomponent of the dialogue game A is alternating.*

The property is established by an easy structural induction on the rooted dialogue game A seen as a formula of tensorial logic.

3.2 The dialogue category **DG** *of balanced strategies*

Now that we have introduced the notion of *balanced play* in a dialogue game, we construct a dialogue category **DG** with dialogue games as objects and balanced strategies as morphisms:

Definition 3.5 A strategy σ of a rooted dialogue game A is *balanced* when all the plays of σ are balanced, or equivalently, when $\sigma \subseteq \mathbf{Bal}_A$.

We are now ready to define the category **DG** using the notion of transverse strategy introduced in Def. 2.1.

Definition 3.6 The category **DG** has dialogue games A, B as objects and transverse balanced strategies of $A -\!\!\bullet B = \neg\,(A \otimes \neg\, B)$ as morphisms $A \to B$.

Two transverse balanced strategies of $A -\!\!\bullet B$ and of $B -\!\!\bullet C$ are composed as expected. The identity morphism id_A of a dialogue game A is defined as the transverse balanced strategy of $A -\!\!\bullet A$ defined by copycat. Recall from the discussion at the beginning of §3 that the fact that id_A only contains balanced plays means that it is more restrictive with the environment than the usual innocent identity morphism on dialogue games.

Theorem 3.7 *The category* **DG** *defines a dialogue category with finite sums, with tensorial pole defined as the rooted dialogue game $\bot = \neg\, 1$.*

Recall that the dialogue game $\bot = \overset{\alpha}{\neg}\, 1$ has a unique Opponent move $q = (\alpha, \boldsymbol{q})$ defined by filling the unique Opponent cell α by a unique Opponent value $\boldsymbol{q}$.

4 Two permutation equivalences on trajectories

We carry on our analysis of the connection between dialogue games and sequential algorithms by introducing in §4.1 two equivalence relations $\sim_{OP}$ and $\sim_{PO}$ on the balanced trajectories $s, t : x \twoheadrightarrow y$ of a dialogue game A. The two relations are defined by induction on the depth of the dialogue game A, which we treat on that occasion as a formula of tensorial logic. We then exhibit in §4.2 a class of atomic OP-permutations $\diamond_{OP}$ and of atomic PO-permutations $\diamond_{PO}$ which generate for every dialogue game A the equivalence relations $\sim_{OP}$ and $\sim_{PO}$, respectively. We then establish in §4.3 a connectedness property which states that every pair of balanced trajectories $s, t : x \twoheadrightarrow y$ with the same source and target are related by a sequence of $\sim_{OP}$ and $\sim_{PO}$ relations, in the following way:

$$s = s_1 \;\sim_{OP}\; s_2 \;\sim_{PO}\; s_3 \;\sim_{OP}\; \quad \ldots \quad \sim_{PO}\; s_n = t.$$

Connectedness means that the two equivalence relation $\sim_{OP}$ and $\sim_{PO}$ are sufficient to recover the usual permutation equivalence relation $\sim$ of the underlying dialogue game, which identifies two trajectories s and t which play the same moves of the dialogue game but in a different order, see [12,13,14].

4.1 The two permutation equivalences $\sim_{OP}$ and $\sim_{PO}$

We recall the definition of scheduling function (or schedule) formulated in [4] which we will use to study the possible orderings of moves in a trajectory.

Definition 4.1 A *schedule* is a function $e : \{1, \ldots, n\} \to \{0, 1\}$.

A schedule e is thus a sequence of 0s and 1s. We write $|e|$ for the length n of the schedule e, $|e|_0$ for the number of 0s, and $|e|_1$ for the number of 1s. So, $|e| = n = |e|_0 + |e|_1$. Quite obviously, every balanced trajectory $s : x \twoheadrightarrow y$ in a dialogue game $A \otimes B$ defines a schedule e with the same length $|e|$ as the trajectory s, and with $|e|_0$ equal to the length of $s_{|A}$ and $|e|_1$ equal to the length of $s_{|B}$. This schedule is such that $e(i) = 0$ if and only if the $i-$th move of s is in A.
The two equivalence relations $\sim_{OP}$ and $\sim_{PO}$ are defined by structural induction on the dialogue game A, seen as a formula of tensorial logic. Note that the most important step of the construction is the tensor product.

Definition 4.2 The two equivalence relations $\sim^A_{OP}$ and $\sim^A_{PO}$ over balanced trajectories $s, t : x \twoheadrightarrow y$ with same source and target are defined by mutual induction over the dialogue game A. The cases $A = 0$ and $A = 1$ are trivial. In the case of a sum, one defines the two equivalence relations by disjoint union:

$$\sim^{A\oplus B}_{OP} = \sim^A_{OP} \uplus \sim^B_{OP} \qquad\qquad \sim^{A\oplus B}_{PO} = \sim^A_{PO} \uplus \sim^B_{PO}\,.$$

In the case of a tensorial negation, one defines the two equivalence relations on $\neg A$ by restricting the trajectories $s, t : x \twoheadrightarrow y$ with same source and target to the subcomponent A, and by reversing the two equivalence relations in order to reflect the change of point of view:

$$s \sim^{\neg A}_{OP} t \iff s_{|A} \sim^A_{PO} t_{|A} \qquad\qquad s \sim^{\neg A}_{PO} t \iff s_{|A} \sim^A_{OP} t_{|A}$$

In the case of a tensor product, given two trajectories $s, t : x \otimes y \twoheadrightarrow x' \otimes y'$ with same source and target, one defines the two equivalence relations as follows:

$$s \sim_{OP}^{A\otimes B} t \qquad \text{precisely when} \qquad s_{|A} \sim_{OP}^{A} t_{|A} \text{ and } s_{|B} \sim_{OP}^{B} t_{|B}$$

$$s \sim_{PO}^{A\otimes B} t \qquad \text{precisely when} \qquad s_{|A} \sim_{PO}^{A} t_{|A} \text{ and } s_{|B} \sim_{PO}^{B} t_{|B}$$

and moreover, s and t have the same schedule.

The key difference between the two relations is that, at the tensor level, $\sim_{OP}$ allows switching moves between components, while $\sim_{PO}$ does not.
By way of consequence, two trajectories $s, t : x \multimapdot y \twoheadrightarrow x' \multimapdot y'$ related by $\sim_{OP}$ in the dialogue game $A \multimapdot B = \neg\,(A \otimes \neg\, B)$ have the same schedule with respect to the components A and B. The intuition is that Opponent is not allowed to alter using $\sim_{OP}$ the schedule of trajectory in the dialogue game $A \multimapdot B$.

4.2 The atomic permutation tiles $\diamond_{OP}$ and $\diamond_{PO}$

In this section, we show that the equivalence relations $\sim_{OP}$ and $\sim_{PO}$ are generated by two classes of atomic permutation tiles $s \diamond_{OP} t$ and $s \diamond_{PO} t$ on balanced trajectories. By this, we mean that $\sim_{OP}$ is the smallest equivalence relation on balanced trajectories such that $s \sim_{OP} t$ whenever

$$s = u_1 \cdot v \cdot u_2 \quad \text{and} \quad t = u_1 \cdot v' \cdot u_2 \quad \text{and} \quad v \diamond_{OP} v'$$

for some balanced trajectories u_1, u_2, v, v'. In other words, $\sim_{OP}$ is the smallest equivalence relation containing $\diamond_{OP}$ and closed under composition of balanced trajectories. Symmetrically, $\sim_{PO}$ is the smallest equivalence relation containing $\diamond_{PO}$ and closed under composition of balanced trajectories. As we will see, all the permutation tiles $\diamond_{OP}$ and $\diamond_{PO}$ are of the simple form of an atomic permutation described in the introduction. We now define the two generating classes of atomic permutation tiles $\diamond_{OP}$ and $\diamond_{PO}$ by induction on the depth of the dialogue game A, seen as a formula of tensorial logic.

Definition 4.3 We associate to every dialogue game A a class of atomic OP-permutations $s \diamond_{OP} t$ and a class of atomic PO-permutations $s \diamond_{PO} t$. We proceed by structural and mutual induction on the depth of the dialogue game. We start by the case of the tensor product, which is the most important and interesting one.

Tensor product for $\diamond_{OP}$. An atomic OP-permutation $s \diamond_{OP} t$ relates two OP-trajectories $s, t : x \twoheadrightarrow y$ (that is, two trajectories starting with an Opponent move and finishing with a Player move). in the dialogue game $A \otimes B$ in three different cases.

- **Base case:** the two OP-trajectories $s, t : x \twoheadrightarrow y$ are of the form

$$s = m_1 \cdot n_1 \cdot m_2 \cdot n_2 \qquad\qquad t = m_2 \cdot n_2 \cdot m_1 \cdot n_1$$

  where m_1 is an Opponent move and n_1 is a Player move in the dialogue game A while m_2 is an Opponent move and n_2 is a Player move in the dialogue game B.

- **Induction case left:**

  $s_{|A} \diamond_{OP} t_{|A}$ in the dialogue game A and $s_{|B} = t_{|B}$ in the dialogue game B,

- **Induction case right:**

  $s_{|A} = t_{|A}$ in the dialogue game A and $s_{|B} \diamond_{OP} t_{|B}$ in the dialogue game B,

Tensor product for $\diamond_{PO}$. An atomic PO-permutation $s \diamond_{PO} t$ relates two PO-trajectories $s, t : x \twoheadrightarrow y$ in the dialogue game $A \otimes B$ in two different cases.

- **Induction case left:**

  $s_{|A} \diamond_{PO} t_{|A}$ in the dialogue game A and $s_{|B} = t_{|B}$ in the dialogue game B,

- **Induction case right:**

  $s_{|A} = t_{|A}$ in the dialogue game A and $s_{|B} \diamond_{PO} t_{|B}$ in the dialogue game B.

Units. There are no permutation tiles $s \diamond_{OP} t$ or $s \diamond_{PO} t$ for the basic dialogue games 0 and 1.

Tensorial negation. The OP-permutation tiles $s \diamond_{OP} t$ of the dialogue game $\neg A$ are the PO-permutation tiles $s \diamond_{PO} t$ of the dialogue game A seen as permutations between balanced trajectories of $\neg A$. Symmetrically, the PO-permutation tiles of $\neg A$ are the OP-permutation tiles $s \diamond_{OP} t$ of the dialogue game A seen as permutations between balanced trajectories of $\neg A$.

Sum. An OP-permutation tile $s \diamond_{OP} t$ of $A \oplus B$ is either an OP-permutation tile of A or an OP-permutation tile of B. Symmetrically, a PO-permutation tile $s \diamond_{PO} t$ of $A \oplus B$ is either a PO-permutation tile of A or a PO-permutation tile of B.

The definition by structural induction of the permutation tiles $\diamond_{OP}$ and $\diamond_{PO}$ is justified by the generation lemma below, which holds for every dialogue game A.

Proposition 4.4 (Generation) *The OP-permutation equivalence $\sim_{OP}$ is generated by the class of atomic permutation tiles $\diamond_{OP}$. Symmetrically, the PO-permutation equivalence $\sim_{PO}$ is generated by the class of atomic permutation tiles $\diamond_{PO}$.*

4.3 The connectedness theorem

Now that we understand better the combinatorial nature of the two permutation equivalences $\sim_{OP}$ and $\sim_{PO}$, we would like to relate the two relations $\sim_{OP}$ and $\sim_{PO}$ to the structure of the graph $G(A)$ of balanced trajectories for a given dialogue game A. We start by establishing the following property for every pair of balanced trajectories $s, t : x \twoheadrightarrow z$ with same source and same target:

Proposition 4.5 *Suppose that two balanced trajectories $s, t : x \twoheadrightarrow z$ start with two different Opponent moves $m : x \rightarrow y$ for the trajectory s, and $m' : x \rightarrow y'$ for the trajectory t. Then, there exists a balanced trajectory $u : y' \twoheadrightarrow z$ such that $s \sim_{OP} m' \cdot u$.*

Proposition 4.6 *Suppose that two balanced trajectories $s, t : x \twoheadrightarrow z$ start with two different Player moves $n : x \rightarrow y$ for the trajectory s, and $n' : x \rightarrow y'$ for the trajectory t. Then, there exists a balanced trajectory $u : y' \twoheadrightarrow z$ such that $s \sim_{PO} n' \cdot u$.*

The two propositions are established together by structural induction on the dialogue game A. See the proof in Appendix. They imply together that every dialogue game A satisfies the following important connectedness property:

Theorem 4.7 (Connectedness) *Suppose given two balanced trajectories $s, t : x \twoheadrightarrow y$ with same source and target. Then, there exists a sequence of balanced trajectories $s_1, \ldots, s_n : x \twoheadrightarrow y$ and of permutation equivalences*

$$s = s_1 \ \sim_{OP} \ s_2 \ \sim_{PO} \ s_3 \ \sim_{OP} \quad \ldots \quad \sim_{PO} \ s_n = t.$$

5 Graph Games

In this section, we quickly recall the notion of graph game and the construction of **GG** introduced by Hyland and Schalk in [6].

Definition 5.1 A *graph game* A is defined as:

- a set $A = A_P + A_O$ of positions together with an initial position $*_A$,
- a set of oriented edges which makes the graph bipartite and acyclic.

Here, A_P denotes the set of Player positions and A_O the set of Opponent positions of the graph game A. One makes the assumption that there is at most one edge $a \to b$ between two positions of the graph. A play in a graph game A is then defined as a path $s : *_A \twoheadrightarrow x$ starting from the initial position $*_A$. Note that, by definition of a graph game, such a path is necessarily alternating.

Definition 5.2 Let α be a partial function from A_O to A_P such that there is an edge $a \to \alpha(a)$ for every Opponent position a in the domain of α. The set $R(\alpha)$ of *reachable positions* for the partial function α is defined by induction in the following way :

- $*_A \in R(\alpha)$
- if $a \in R(\alpha) \cap A_P$ and $a \to a'$ then $a' \in R(\alpha)$.
- if $a \in R(\alpha) \cap A_O$ and $\alpha(a)$ is defined (and thus $a \to \alpha(a)$), then $\alpha(a) \in R(\alpha)$.

Definition 5.3 A *pre-strategy* of a graph game A is a partial function α from A_O to A_P such that $a \to \alpha(a)$ when $\alpha(a)$ is defined and such that its domain of definition is a subset of $R(\alpha)$.

Definition 5.4 A pre-strategy is *conflict-free* when for all Player position $a \in R(\alpha) \cap A_P$ reachable from $a' \in R(\alpha) \cap A_O$ then $\alpha(a')$ is defined and a is reachable from $\alpha(a')$.

The intuition of conflict-freeness is that no Player move should block access to a potential future player position if the starting Opponent position allowed access to that position. In other words, as long as Opponent chooses moves that are compatible with a given target position, the strategy's answer will still be compatible with that target.

Definition 5.5 *The linear function space* $A \multimap B$ is the game with

- P-positions $A_P \times B_P + A_O \times B_O$
- O-positions $A_P \times B_O$

the initial position is $*_A, *_B$ and there are moves $a, b \to a', b'$ just when

- either $b = b'$ and $a \to a'$ is a move in A
- or $a = a'$ and $b \to b'$ is a move in B

Definition 5.6 *The category of graph games* **GG** is the category defined by :

- graph games as objects
- for two graph games A, B, conflict-free pre-strategies of $A \multimap B$ as arrows of $A \to B$

We have now recalled the category of graph games and conflict-free pre-strategies. In what follows, we will relate this category to our dialogue games.

6 A functor from dialogue games to graph games

We formulate now a notion of *conflict-free strategy* on dialogue games. The notion of conflict-freeness originates from rewriting theory, see [9,11], and adapts the notion formulated by Hyland and Schalk in the framework of graph games [6].

Definition 6.1 A balanced strategy σ is *conflict-free* when for every pair of balanced plays $s : * \twoheadrightarrow x$ and $t : * \twoheadrightarrow y$ in the strategy σ, and for every Opponent move $m : x \to z$ and balanced trajectory $u : z \twoheadrightarrow y$, there exists a Player move $n : z \to x'$ and a balanced trajectory $v : x' \twoheadrightarrow y$ such that $s \cdot m \cdot n \in \sigma$.

We will be particularly interested in the category $\mathbf{DG}_{\mathsf{bal}}$ of rooted dialogue games and balanced conflict-free strategies. In order to perform our comparison with graph games, we adapt the definition of the positional graph (Def. 1.6) associated to a dialogue game, by considering only its balanced positions:

Definition 6.2 The *balanced graph* $G(A)$ of a rooted dialogue game A is the restriction of $\mathbf{Graph}_A$ to the balanced positions and to the edges between them. Note that the paths of $G(A)$ are exactly the balanced trajectories of A.

This definition of the balanced graph $G(A)$ enables us to construct a functor

$$G \quad : \quad \mathbf{DG}_{\mathsf{bal}} \quad \longrightarrow \quad \mathbf{GG}$$

from the dialogue category $\mathbf{DG}_{\mathsf{bal}}$ to the dialogue category **GG** of graph games defined by Hyland and Schalk [6], see the definitions in Appendix 5. The graph game $G(A)$ associated to a dialogue game A is defined in the following way: $G(A)_P$ is the set of balanced positions of payoff $+1$, while $G(A)_O$ is the set of balanced positions of payoff -1. The strategy $G(\sigma)$ associated to a balanced conflict-free strategy σ is defined as the partial function $G(\sigma)$ which assigns to every position

$a \in G(A)_O$ the position $G(\sigma)(a) \in G(A)_P$ whenever there exists a balanced play $s : * \twoheadrightarrow a$ such that the play $s \cdot n : * \twoheadrightarrow a \to G(\sigma)(a)$ is an element of the strategy σ. The next result establishes a clean correspondence between dialogue games and graph games:

Theorem 6.3 *The functor $G : \mathbf{DG}_{\mathsf{bal}} \to \mathbf{GG}$ is fully faithful and transports the dialogue category structure of $\mathbf{DG}_{\mathsf{bal}}$ to the dialogue category structure of $\mathbf{GG}$.*

The intuition behind the proof is that every conflict-free balanced strategy in the category $\mathbf{DG}_{\mathsf{bal}}$ is transported by the functor G to a conflict-free pre-strategy in the sense of Hyland and Schalk, and thus to a morphism of $\mathbf{GG}$.

7 The bi-invariance theorem

The relationship between graph games and dialogue games established in the two previous sections (Thms 4.7 and 6.3) enables us to formulate the notion of conflict-freeness in a different and more combinatorial way.

Theorem 7.1 (Bi-invariance) *A balanced strategy σ is conflict-free in a dialogue game A if and only if it is bi-invariant in the sense that*

$$\forall s \in \sigma, \forall t \in \boldsymbol{Bal}_A, \qquad s \sim_{OP} t \quad \Rightarrow \quad \exists u \in \sigma,\ t \sim_{PO} u.$$

The intuition is that a balanced strategy σ is bi-invariant whenever it is able to adapt to a change performed by Opponent in the order of execution (along $\sim_{OP}$) by changing its own order of execution (along $\sim_{PO}$). This description of conflict-free strategies is reminiscent of the diagrammatic definition of innocent strategies in [12], see also [14] (Chapter 3, Section 3.4).

A basic illustration of the bi-invariance theorem is provided by the left-to-right implementation σ_{LR} of the strict conjunction of type $(\mathbb{B} \otimes \mathbb{B}) \multimap \mathbb{B}$. The strategy σ_{LR} is conflict-free and contains the play s_{LR} below

$$\boxed{\begin{array}{ccccc}
\mathbb{B} & \otimes & \mathbb{B} & \multimap & \mathbb{B} \\
 & & & & q \\
q & & & & \\
true & & & & \\
 & & q & & \\
 & & false & & \\
 & & & & false
\end{array}}$$

which is $\sim_{PO}$-equivalent to the play s_{LR} below

$$\boxed{\begin{array}{ccccc}
\mathbb{B} & \otimes & \mathbb{B} & \multimap & \mathbb{B} \\
 & & & & q \\
 & & q & & \\
 & & false & & \\
q & & & & \\
true & & & & \\
 & & & & false
\end{array}}$$

Clearly, the play s_{RL} is an element of the right-to-left implementation σ_{RL} of the strict conjunction, and not an element of the left-to-right implementation σ_{LR}. Moreover, by the inductive definition of the equivalence $\sim_{PO}$, the two plays

$$q \cdot s_{LR} \cdot true \qquad \text{and} \qquad q \cdot s_{RL} \cdot true$$

are $\sim_{OP}$-equivalent in the balanced dialogue game

$$((\mathbb{B} \otimes \mathbb{B}) \multimap \mathbb{B}) \multimap \mathbb{B} \quad = \quad ((\mathbb{B} \otimes \mathbb{B}) \multimapdot (1 \oplus 1)) \multimapdot (1 \oplus 1)$$

Note that the equivalence relation $\sim_{PO}$ is trivial in that game. From this follows by the bi-invariance theorem (because the relation $\sim_{PO}$ is trivial) that every conflict-free strategy which contains the play $q \cdot s_{LR} \cdot true$ also contains the play $q \cdot s_{RL} \cdot true$.

Conclusion

We have constructed a fully faithful translation from the category of dialogue games and balanced strategies to the category of Hyland-Schalk graph games and conflict-free strategies [6,15]. The bridge between dialogue games and graph games discloses a number of interesting combinatorial structures. In particular, we establish that the structure of the Hyland-Schalk graph games can be recovered from primitive permutation tiles $\diamond_{OP}$ and $\diamond_{PO}$ of a particularly simple shape. This discovery offers a complement and counterpoint to the work by Harmer, Hyland and Melliès [4] and conveys hope for a tighter connection between the existing paradigms of game semantics, from concrete data structures to arena games.

References

[1] G. Berry and P.-L. Curien. Sequential algorithms on concrete data structures. In *Theoretical Computer Science*, volume 20, pages 265–321, 1982.

[2] P.-L. Curien. On the symmetry of sequentiality. In *Mathematical Foundations of Programming Semantics*, number 802 in Lecture Notes in Computer Science. Springer-Verlag, 1993.

[3] J.Y. Girard. Locus Solum: From the rules of logic to the logic of rules. *Mathematical Structures in Computer Science*, 11(03):301–506, 2001.

[4] R. Harmer, J.M.E. Hyland, and P.-A. Melliès. Categorical combinatorics for innocent strategies. In *LICS*, pages 379–388, 2007.

[5] M. Hyland. Game semantics. In *Semantics of Logics and Computation*, pages 131–184. Publications of the Newton Institute, Cambridge University Press, 1997.

[6] M. Hyland and A. Schalk. Games on graphs and sequentially realizable functionals. In *LICS 02*, pages 257–264, Kopenhavn, July 2002. IEEE Computer Society Press.

[7] G. Kahn and G. Plotkin. Structures de données concrètes. *Rapport IRIA-LABORIA*, page p. 336, 1978.

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

[9] P-A. Melliès. Sequentiality and extensoniality. 13 pages, Unpublished notes circulated at the Linear Summer School, Azores, September 2000, September 2000.

[10] P.-A. Melliès. Asynchronous games 4: A fully complete model of propositional linear logic. *LICS'05*, 2005.

[11] P.-A. Melliès. Sequential algorithms and strongly stable functions. *Theoretical Computer Science*, 343(1-2):237–281, 2005.

[12] P.-A. Melliès. Asynchronous games 2: the true concurrency of innocence. *Theoretical Computer Science*, 358:200–228, 2006.

[13] P.-A. Melliès. On dialogue games and coherent strategies. In *CSL 2013*, volume 23 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013.

[14] Paul-André Melliès. Une étude micrologique de la négation. Habilitation à Diriger des Recherches, November 2017. Available at the author's webpage: www.irif.fr/∼mellies/hdr-mellies.pdf.

[15] A. Schalk and J. Palacios-Perez. Concrete data structures as games. *Electronic Notes in Theoretical Computer Science*, 122:193 – 210, 2005. CTCS 2004.