

Categorical Combinatorics of Scheduling and Synchronization in Game Semantics

PAUL-ANDRÉ MELLIES, CNRS, Université Paris Diderot, France

Game semantics is the art of interpreting types as *games* and programs as *strategies* interacting in space and time with their environment. In order to reflect the interactive behavior of programs, strategies are required to follow specific scheduling policies. Typically, in the case of a purely sequential programming language, the program (Player) and its environment (Opponent) will play one after the other, in a strictly alternating way. On the other hand, in the case of a concurrent language, Player and Opponent will be allowed to play several moves in a row, in a non-alternating way. In both cases, the scheduling policy is designed very carefully in order to ensure that the strategies synchronize properly and compose well when plugged together. A longstanding conceptual problem has been to understand when and why a given scheduling policy works and is compositional in that sense. In this paper, we exhibit a number of simple and fundamental combinatorial structures which ensure that a given scheduling policy encoded as *synchronization template* defines a symmetric monoidal closed (and in fact $*$ -autonomous) bicategory of games, strategies and simulations. To that purpose, we choose to work at a very general level, and illustrate our method by constructing two template game models of linear logic with different flavors (alternating and non-alternating) using the same categorical combinatorics, performed in the category of small categories. As a whole, the paper may be seen as a hymn in praise of synchronization, building on the notion of *synchronization algebra* in process calculi and adapting it smoothly to programming language semantics, using a combination of ideas at the converging point of game semantics and of categorical algebra.

CCS Concepts: • **Theory of computation** → **Concurrency**; **Denotational semantics**; **Categorical semantics**;

Additional Key Words and Phrases: game semantics, synchronization algebras, scheduling policies, sequential alternating games, concurrent non-alternating games, template games, bicategorical models of linear logic

ACM Reference Format:

Paul-André Mellies. 2019. Categorical Combinatorics of Scheduling and Synchronization in Game Semantics. *Proc. ACM Program. Lang.* 3, POPL, Article 23 (January 2019), 30 pages. <https://doi.org/10.1145/3290336>

1 INTRODUCTION

In this paper, we want to describe in a simple, uniform and elegant way how proofs and programs of different nature, typically sequential or concurrent, synchronize themselves and interact in space and time. In order to describe these various scheduling policies (sequential, concurrent) in the same framework, we will combine ideas coming from concurrency theory (synchronization algebras) and programming language theory (categorical semantics) following a trend pioneered by game semantics [Abramsky et al. 2000; Hyland and Ong 2000]. We start from the well-known

Author's address: Paul-André Mellies, Institut de Recherche en Informatique (IRIF), CNRS, Université Paris Diderot, 8 place Aurélie Nemours, Paris, 75013, France, mellies@irif.fr.



This work is licensed under a Creative Commons Attribution 4.0 International License.

© 2019 Copyright held by the owner/author(s).

2475-1421/2019/1-ART23

<https://doi.org/10.1145/3290336>

idea in graph theory that the coloring of a graph G with n colors is the same thing as a graph homomorphism

$$\lambda : G \longrightarrow K_n \quad (1)$$

to the complete graph K_n with n vertices. Typically, a graph G is *bipartite* when there exists a map (1) to the complete graph K_2 with two vertices.

1.1 Games as Categories with Polarities

Here, we adapt this simple idea to categories, and explain how to derive from it a new generation of 2-categorical game semantics. Our starting point is provided by the category $\mathfrak{z}_{\text{game}}$ which we call the *template of games* and define as the category with two objects $\langle \oplus \rangle$ and $\langle \ominus \rangle$ freely generated by the oriented graph

$$\langle \ominus \rangle \begin{array}{c} \xleftarrow{O} \\ \xrightarrow{P} \end{array} \langle \oplus \rangle \quad (2)$$

A category A equipped with a functor

$$\lambda_A : A \longrightarrow \mathfrak{z}_{\text{game}}$$

defines a general form of 2-Player game, where every object x of the category A is understood as a *position* of the game, polarized either as *positive* when $\lambda_A(x) = \langle \oplus \rangle$ or as *negative* when $\lambda_A(x) = \langle \ominus \rangle$. Accordingly, the maps $f : x \rightarrow y$ of the category A are understood as the *trajectories* of the game. The category of polarities $\mathfrak{z}_{\text{game}}$ thus plays the same role for games as the graph K_2 for bipartite graphs. In particular, among the trajectories of A , we distinguish two classes of maps: the *Opponent moves* and *Player moves* defined as the trajectories

$$m : x^{\oplus} \longrightarrow y^{\ominus} \qquad n : x^{\ominus} \longrightarrow y^{\oplus}$$

such that $\lambda_A(m) = O$ and $\lambda_A(n) = P$ respectively. Here, we write x^{\oplus} or x^{\ominus} to indicate the polarity (positive or negative) assigned by the *polarity functor* λ_A to the position x in the game (A, λ_A) . The intuition guiding the definition of $\mathfrak{z}_{\text{game}}$ is that a position with polarity \oplus is a *Player position* where Player is satisfied and waits for Opponent to play, and symmetrically, that a position with polarity \ominus is an *Opponent position* where Opponent is satisfied and waits for Player to play. Note that a game A generally contains trajectories of the form

$$f : x^{\oplus} \longrightarrow y^{\oplus} \qquad g : x^{\ominus} \longrightarrow y^{\ominus}$$

which are typically obtained as sequences of moves of even length

$$f, g : x = x_1 \xrightarrow{m_1} x_2 \xrightarrow{m_2} \cdots \xrightarrow{m_{k-1}} x_k \xrightarrow{m_k} x_{k+1} = y$$

alternating between Opponent and Player moves.

1.2 Strategies between Games

Once the notion of game has been formulated, the next step is to define a notion of *strategy* $\sigma : A \rightarrow B$ between two such games A and B . The definition relies on the introduction of a specific category $\mathfrak{z}_{\text{strat}}$ which we call the *template of strategies* with three objects $\langle \oplus, \oplus \rangle$, $\langle \oplus, \ominus \rangle$, and $\langle \ominus, \ominus \rangle$ called its *scheduling polarities*. The purpose of the three scheduling polarities is to represent the pair of polarities reached in the source game A and in the target game B in the course of interaction. The category $\mathfrak{z}_{\text{strat}}$ is defined as the category freely generated by the graph

$$\langle \ominus, \ominus \rangle \begin{array}{c} \xleftarrow{P_s} \\ \xrightarrow{O_s} \end{array} \langle \oplus, \ominus \rangle \begin{array}{c} \xleftarrow{O_t} \\ \xrightarrow{P_t} \end{array} \langle \oplus, \oplus \rangle \quad (3)$$

The morphisms of the category $\mathfrak{A}_{\text{strat}}$ are called its *scheduling trajectories*. Each of the four labels O_s, P_s, O_t and P_t is here to describe a specific kind of Opponent and Player move:

- O_s : Opponent move played in the source game
- P_s : Player move played in the source game
- O_t : Opponent move played in the target game
- P_t : Player move played in the target game

Somewhat unexpectedly, the category $\mathfrak{A}_{\text{strat}}$ satisfies a number of remarkable properties. The category comes equipped with a span of functors

$$\mathfrak{A}_{\text{game}} \xleftarrow{s} \mathfrak{A}_{\text{strat}} \xrightarrow{t} \mathfrak{A}_{\text{game}} \quad (4)$$

where the functor s is defined as the “projection” on the first component:

$$\begin{array}{ll} \langle \ominus, \ominus \rangle \mapsto \langle \ominus \rangle & O_s \mapsto P \quad P_s \mapsto O \\ \langle \oplus, \ominus \rangle, \langle \oplus, \oplus \rangle \mapsto \langle \oplus \rangle & O_t, P_t \mapsto \text{id}_{\langle \oplus \rangle} \end{array}$$

and the functor t is defined as the “projection” on the second component:

$$\begin{array}{ll} \langle \oplus, \oplus \rangle \mapsto \langle \oplus \rangle & O_t \mapsto P \quad P_t \mapsto O \\ \langle \ominus, \ominus \rangle, \langle \oplus, \ominus \rangle \mapsto \langle \ominus \rangle & O_s, P_s \mapsto \text{id}_{\langle \ominus \rangle} \end{array}$$

This leads us to the definition of a strategy $\sigma : A \rightarrow B$ between two games A and B , which is defined in our formal approach as a span of functors

$$A \xleftarrow{s} S \xrightarrow{t} B \quad (5)$$

together with a functor $\lambda_\sigma : S \rightarrow \mathfrak{A}_{\text{strat}}$ making the diagram

$$\begin{array}{ccccc} A & \xleftarrow{s} & S & \xrightarrow{t} & B \\ \lambda_A \downarrow & & \downarrow \lambda_\sigma & & \downarrow \lambda_B \\ \mathfrak{A}_{\text{game}} & \xleftarrow{s} & \mathfrak{A}_{\text{strat}} & \xrightarrow{t} & \mathfrak{A}_{\text{game}} \end{array} \quad (6)$$

commute. The category S is called the *support* of the strategy σ , and λ_σ is called its *polarity functor*. In the same way as for games, the objects x and maps $f : x \rightarrow y$ of the category S are called the *positions* and the *trajectories* of (the support S of) the strategy σ . Among the trajectories of the strategy σ , we distinguish two classes of maps: the Opponent moves defined as the trajectories

$$m : x^{\oplus\oplus} \longrightarrow y^{\oplus\ominus} \quad n : x^{\ominus\ominus} \longrightarrow y^{\oplus\oplus}$$

in the support S such that $\lambda_\sigma(m) = O_t$ or $\lambda_\sigma(n) = O_s$; and the Player moves defined as the trajectories

$$m : y^{\oplus\ominus} \longrightarrow x^{\oplus\oplus} \quad n : y^{\oplus\oplus} \longrightarrow z^{\ominus\ominus}$$

in the support S such that $\lambda_\sigma(m) = P_t$ or $\lambda_\sigma(n) = P_s$. Here, in the same way as we did earlier for positions in games, we write $x^{\oplus\oplus}, y^{\oplus\oplus}$ and $z^{\ominus\ominus}$ to indicate the value of the polarity functor λ_σ taken in $\mathfrak{A}_{\text{strat}}$ and applied to the positions x, y, z of the support S .

1.3 Illustration

The intuition behind our definition of strategy $\sigma : A \rightarrow B$ is that every trajectory $f : x \rightarrow y$ in the support S of the strategy induces a pair of trajectories

$$f_A = s(f) : s(x) \longrightarrow s(y) \quad f_B = t(f) : t(x) \longrightarrow t(y)$$

living in the game A and in the game B , respectively. The trajectory $f : x \rightarrow y$ is transported by the functor λ_σ to the map or scheduling trajectory in the category $\mathfrak{X}_{\text{strat}}$

$$f_\sigma = \lambda_\sigma(f) : \lambda_\sigma(x) \longrightarrow \lambda_\sigma(y)$$

whose purpose is to provide the “glue” which will schedule and synchronize the two trajectories f_A and f_B together. By way of illustration, imagine that the support S of the strategy σ contains a trajectory

$$f : x_1^{\oplus\oplus} \xrightarrow{m_1} x_2^{\oplus\ominus} \xrightarrow{n_1} x_3^{\ominus\ominus} \xrightarrow{m_2} x_4^{\oplus\ominus} \xrightarrow{n_2} x_5^{\oplus\oplus}$$

where m_1, m_2 are Opponent moves, and n_1, n_2 are Player moves of the strategy σ , in the sense just formulated that $\lambda_\sigma(m_1) = O_t$, $\lambda_\sigma(n_1) = P_s$, $\lambda_\sigma(m_2) = O_s$ and $\lambda_\sigma(n_2) = P_t$. As expected, the trajectory $f : x_1 \rightarrow x_5$ is transported by the polarity functor λ_σ to the scheduling trajectory

$$\lambda_\sigma(f) : \langle \oplus, \oplus \rangle \xrightarrow{O_t} \langle \oplus, \ominus \rangle \xrightarrow{P_s} \langle \ominus, \ominus \rangle \xrightarrow{O_s} \langle \oplus, \ominus \rangle \xrightarrow{P_t} \langle \oplus, \oplus \rangle \quad (7)$$

This scheduling trajectory tells us that the sequence of moves $m_1 \cdot n_1 \cdot m_2 \cdot n_2$ of the strategy σ plays successively in the target, then source, then source, and finally target component of the game. For that reason, the sequence of moves $m_1 \cdot n_1 \cdot m_2 \cdot n_2$ would be represented in the following way in the traditional (and informal) notation for plays in sequential games:

| | | | | |
|-------------------------------------|-------|------------------------|-------|-----|
| | A | $\xrightarrow{\sigma}$ | B | |
| first move m_1 of polarity O_t | | | m_1 | |
| second move n_1 of polarity P_s | n_1 | | | (8) |
| third move m_2 of polarity O_s | m_2 | | | |
| fourth move n_2 of polarity P_t | | | n_2 | |

One benefit of our approach based on the category $\mathfrak{X}_{\text{strat}}$ is that the scheduling trajectory (7) gives a precise mathematical meaning to the pattern of scheduling represented in (8).

1.4 The Category of Strategies and Simulations

It should be noted that the categorical ingredients used at this stage in order to define *games* and *strategies* are extremely simple. All in all, they essentially consist in the functorial spans (4) and (5) together with the map (6) between them. In particular, our principled definition of strategies ensures already that there exists a category $\mathbf{Games}(A, B)$ of strategies $\sigma : A \rightarrow B$ between any two games A and B . In order to define this category, we introduce the notion of *simulation*

$$\theta : \sigma \Longrightarrow \tau : A \longrightarrow B$$

between two strategies $\sigma, \tau : A \rightarrow B$, which is defined as a functor

$$\theta : S \longrightarrow T$$

from the support S of σ to the support T of τ , making the three triangles below commute:

| | | |
|--|---|--|
| $\begin{array}{ccc} S & \xrightarrow{\theta} & T \\ \swarrow s & & \searrow s \\ & \mathfrak{X}_{\text{game}} & \end{array}$ | $\begin{array}{ccc} S & \xrightarrow{\theta} & T \\ \swarrow \lambda_\sigma & & \searrow \lambda_\tau \\ & \mathfrak{X}_{\text{strat}} & \end{array}$ | $\begin{array}{ccc} S & \xrightarrow{\theta} & T \\ \swarrow t & & \searrow t \\ & \mathfrak{X}_{\text{game}} & \end{array}$ |
|--|---|--|

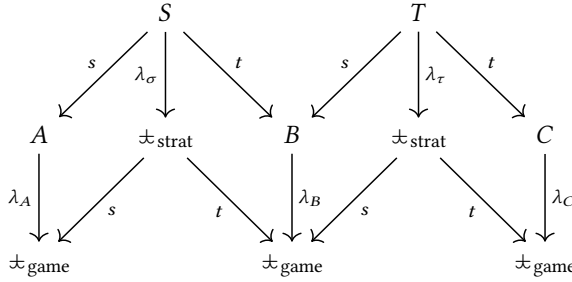
The category $\mathbf{Games}(A, B)$ is defined as the category with strategies $\sigma : A \rightarrow B$ as objects, and the simulations $\theta : \sigma \Rightarrow \tau$ as maps.

1.5 The Bicategory of Games and Strategies [Composition Law]

We explain now how to turn the family of categories $\mathbf{Games}(-, -)$ just constructed into a bicategory \mathbf{Games} of games, strategies and simulations. To that purpose, we need to construct a functor

$$\circ_{A,B,C} : \mathbf{Games}(B, C) \times \mathbf{Games}(A, B) \longrightarrow \mathbf{Games}(A, C)$$

which transports a pair of strategies $\sigma \in \mathbf{Games}(A, B)$ and $\tau \in \mathbf{Games}(B, C)$ to their composite strategy $\tau \circ \sigma \in \mathbf{Games}(A, C)$, for every 3-tuple of games A, B and C . Suppose that σ and τ have the categories S and T as respective supports. The construction starts by putting the pair of functorial spans defining σ and τ side by side, in the following way:



In order to perform the construction, we consider the category \mathfrak{int} called the *template of interactions* and defined by the pullback diagram of small categories below:

$$\begin{array}{ccccc}
 & & \mathfrak{int} & & \\
 & \swarrow \pi_1 & & \searrow \pi_2 & \\
 \mathfrak{strat} & & pb & & \mathfrak{strat} \\
 \swarrow s & & \downarrow t & & \swarrow s \\
 \mathfrak{game} & & \mathfrak{game} & & \mathfrak{game} \\
 & & & & \searrow t
 \end{array} \tag{9}$$

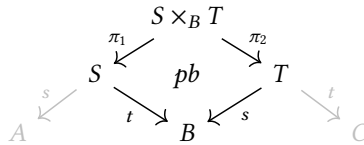
An easy and instructive computation detailed in Appendix A shows that the category \mathfrak{int} has four objects

$$\langle \ominus, \ominus, \ominus \rangle \quad \langle \oplus, \ominus, \ominus \rangle \quad \langle \oplus, \oplus, \ominus \rangle \quad \langle \oplus, \oplus, \oplus \rangle$$

and is in fact freely generated by the following graph:

$$\langle \ominus, \ominus, \ominus \rangle \xleftarrow[\underset{O_s}{\rceil}]{\overset{P_s}{\lceil}} \langle \oplus, \ominus, \ominus \rangle \xleftarrow[\underset{P|O}{\rceil}]{\overset{O|P}{\lceil}} \langle \oplus, \oplus, \ominus \rangle \xleftarrow[\underset{P_t}{\rceil}]{\overset{O_t}{\lceil}} \langle \oplus, \oplus, \oplus \rangle \tag{10}$$

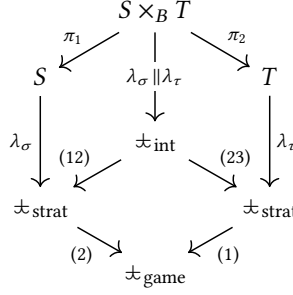
Now, consider the category $S \times_B T$ induced by the pullback diagram below, whose definition is designed to implement the parallel composition of σ and τ above the category B :



By the universality property of \mathfrak{int} , there exists a unique functor

$$\lambda_\sigma \parallel \lambda_\tau : S \times_B T \longrightarrow \mathfrak{int}$$

making the diagram below commute



Here, we find convenient to use the notations (1) and (2) for the functors

$$s = (1) : \text{strat} \rightarrow \text{game} \qquad t = (2) : \text{strat} \rightarrow \text{game}$$

defined in (4) because the notation describes well how the two functors behave on objects. The purpose of this functor $\lambda_\sigma \parallel \lambda_\tau$ is to describe how the *parallel composition* of the two strategies σ and τ are scheduled together. At this stage, we make the first key observation of the paper: the category int of interactions comes equipped with a functor

$$\text{hide} : \text{int} \longrightarrow \text{strat} \tag{11}$$

which makes the diagram below commute:

$$\begin{array}{ccccc}
 \text{strat} & \xleftarrow{(12)} & \text{int} & \xrightarrow{(23)} & \text{strat} \\
 \downarrow (1) & & \downarrow \text{hide} & & \downarrow (2) \\
 \text{game} & \xleftarrow{(1)} & \text{strat} & \xrightarrow{(2)} & \text{game}
 \end{array} \tag{12}$$

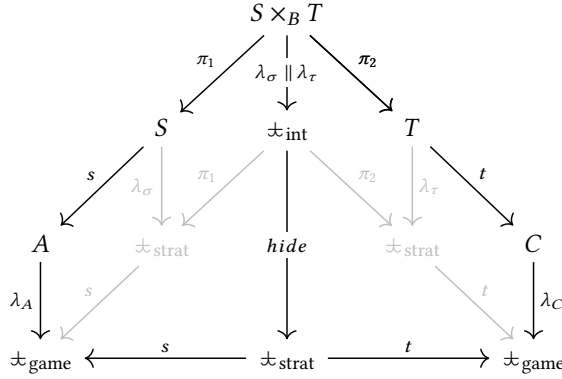
and thus defines a map of functorial spans. The functor *hide* is defined as the “projection” on the first and third components, and is noted (13) for that reason:

$$\begin{array}{lll}
 \langle \ominus, \ominus, \ominus \rangle \mapsto \langle \ominus, \ominus \rangle & O_s \mapsto O_s & P_s \mapsto P_s \\
 \langle \oplus, \ominus, \ominus \rangle, \langle \oplus, \oplus, \ominus \rangle \mapsto \langle \oplus, \ominus \rangle & O|P, P|O \mapsto \text{id}_{(\oplus, \ominus)} & \\
 \langle \oplus, \oplus, \oplus \rangle \mapsto \langle \oplus, \oplus \rangle & O_s \mapsto O_s & P_s \mapsto P_s
 \end{array}$$

The composite strategy $\tau \circ \sigma : A \rightarrow C$ is then defined as the strategy with support $S \times_B T$ obtained by postcomposing the functor $\lambda_\sigma \parallel \lambda_\tau$ with the functor *hide*, in the following way:

$$S \times_B T \xrightarrow{\lambda_\sigma \parallel \lambda_\tau} \text{int} \xrightarrow{\text{hide}} \text{strat}$$

We obtain the strategy $\tau \circ \sigma : A \rightarrow C$ with support $S \times_B T$ depicted as the span below:



This definition of $\tau \circ \sigma$ implements very precisely the slogan that

$$\boxed{\text{composition} = \text{parallel composition} + \text{hiding}}$$

The category $S \times_B T$ implements the parallel composition of σ and τ , and comes equipped with a functor $\lambda_\sigma \parallel \lambda_\tau$ to the category $\mathfrak{A}_{\text{int}}$. This functor $\lambda_\sigma \parallel \lambda_\tau$ describes how the interaction is scheduled between σ and τ . It is worth noticing that the second step of hiding by postcomposing $\lambda_\sigma \parallel \lambda_\tau$ with the functor *hide* keeps the category $S \times_B T$ as support of the strategy $\tau \circ \sigma$, without altering it.

1.6 The Bicategory of Games and Strategies [Identities]

We carry on the construction of the bicategory **Games** by explaining how the identity strategies $\text{cc}_A : A \rightarrow A$ of the bicategory can be defined by purely categorical means. The construction relies on a second key observation, which complements the introduction of the functor *hide* in (11). There exists a functor

$$\text{copycat} : \mathfrak{A}_{\text{game}} \longrightarrow \mathfrak{A}_{\text{strat}} \quad (13)$$

which makes the diagram below commute:

$$\quad (14)$$

The functor *copycat* is defined as follows on the objects and morphisms of $\mathfrak{A}_{\text{game}}$:

$$\begin{aligned} \langle \ominus \rangle &\mapsto \langle \ominus, \ominus \rangle & O &\mapsto O_t \cdot P_s \\ \langle \oplus \rangle &\mapsto \langle \oplus, \oplus \rangle & P &\mapsto O_s \cdot P_t \end{aligned}$$

Given a game A , the *copycat* strategy $\text{cc}_A : A \rightarrow A$ is defined as the strategy with support the category A itself, and with functor λ_{cc_A} obtained by postcomposing the functor λ_A with the functor *copycat*, in the following way:

$$\lambda_{\text{cc}_A} = A \xrightarrow{\lambda_A} \mathfrak{A}_{\text{game}} \xrightarrow{\text{copycat}} \mathfrak{A}_{\text{strat}}$$

We obtain the identity map $\text{cc}_A : A \rightarrow A$ of the bicategory **Games** of games and strategies, depicted as the span below:

$$\begin{array}{ccccc}
 & & A & & \\
 & \swarrow & \downarrow \lambda_A & \searrow & \\
 & & \mathfrak{J}_{\text{game}} & & \\
 & \swarrow \text{id} & & \searrow \text{id} & \\
 A & & \mathfrak{J}_{\text{game}} & & A \\
 \downarrow \lambda_A & \swarrow \text{id} & \downarrow \text{copycat} & \searrow \text{id} & \downarrow \lambda_A \\
 \mathfrak{J}_{\text{game}} & \xleftarrow{s} & \mathfrak{J}_{\text{strat}} & \xrightarrow{t} & \mathfrak{J}_{\text{game}}
 \end{array} \tag{15}$$

Note that the functor *copycat* transports the edges O and P of the category $\mathfrak{J}_{\text{game}}$ into the maps $O_t \cdot P_s$ and $O_s \cdot P_t$ of the category $\mathfrak{J}_{\text{strat}}$, respectively. The two maps $O_t \cdot P_s$ and $O_s \cdot P_t$ are composed of two edges, instead of just one. This means that postcomposing λ_A with the functor *copycat* involves a change of *granularity* of moves, which turns

- every O -move m of the game A into the same move m played *simultaneously* as an O_t -move in the codomain game A and as a P_s -move in the domain game A ,
- every P -move m of the game A into the same move m played *simultaneously* as an O_s -move in the domain game A and as a P_t -move in the codomain game A .

In other words, the *copycat* strategy $\text{cc}_A : A \rightarrow A$ is *synchronous* and transmits the information *instantaneously* from one component A to the other. As far as we know, this phenomenon of “rescaling” and turning moves into plays is entirely new in game semantics. As a matter of fact, we will establish in the technical core of the paper that

Theorem. The bicategory **Games** of sequential alternating games and strategies is symmetric monoidal closed (and in fact $*$ -autonomous.)

The result is important in itself, but what matters even more to us is the methodology we use in order to establish it. First, we derive the statement that **Games** is a bicategory from the purely combinatorial observation that the triple of *synchronization templates* underlying and regulating sequential alternating games and strategies

$$\mathfrak{J}_{\text{alt}}[0] = \mathfrak{J}_{\text{game}} \quad \mathfrak{J}_{\text{alt}}[1] = \mathfrak{J}_{\text{strat}} \quad \mathfrak{J}_{\text{alt}}[2] = \mathfrak{J}_{\text{int}}$$

together with the functors (11) and (13) defines an internal category $\mathfrak{J}_{\text{alt}}$ in the category **Cat** of small categories and functors. Then, we deduce the statement of the theorem from the fact that the internal category $\mathfrak{J}_{\text{alt}}$ is *symmetric monoidal closed* (and in fact $*$ -autonomous) in an appropriate sense, carefully axiomatized in the paper. A clarifying remark about bicategories may be useful at this point. The composition of three strategies

$$\sigma \in \mathbf{Games}(A, B) \quad \tau \in \mathbf{Games}(B, C) \quad \upsilon \in \mathbf{Games}(C, D)$$

is not strictly associative, but only associative up to an invertible simulation

$$\alpha_{\upsilon, \tau, \sigma} : (\upsilon \circ \tau) \circ \sigma \Longrightarrow \upsilon \circ (\tau \circ \sigma) : A \longrightarrow D$$

Similarly, the *copycat* strategy is an identity only up to invertible simulations:

$$\lambda_\sigma : \mathbf{id}_B \circ \sigma \Longrightarrow \sigma : A \longrightarrow B \quad \rho_\sigma : \sigma \circ \mathbf{id}_A \Longrightarrow \sigma : A \longrightarrow B$$

This is the reason why our construction **Games** defines a bicategory in the sense of [Bénabou 1967] and not a 2-category where the three coercions $\alpha_{v,\tau,\sigma}$, λ_σ and ρ_σ would be required (by definition) to be identity 2-cells.

1.7 Contributions of the Paper

We have just explained how to derive our first model of *template games* from the data of a *synchronization template* provided in this case by the internal category $\mathfrak{A}_{\text{alt}}$ of sequential alternating games and strategies. More generally, given a category \mathbb{S} with finite limits, we show how to derive a symmetric monoidal closed bicategory **Games**(\mathfrak{A}) of games, strategies and simulations from the data of an internal category \mathfrak{A} of polarities in \mathbb{S} , equipped with a symmetric monoidal closed structure, formulated in the appropriate sense. The methodology is conceptually clean, and provides a powerful and flexible way to construct new bicategories of games. In this paper, we illustrate the flexibility of the approach by deriving *from the same construction* two game semantics with entirely different flavors (sequential and concurrent) as well as a relational semantics of linear logic. The three constructions are performed in the category $\mathbb{S} = \text{Cat}$ of small categories. We obtain in this way three symmetric monoidal closed (and in fact *-autonomous) bicategories, each of them defining a particular model of linear logic:

- a bicategory **Games**($\mathfrak{A}_{\text{alt}}$) of sequential alternating games,
- a bicategory **Games**($\mathfrak{A}_{\text{conc}}$) of concurrent non-alternating games,
- a bicategory **Games**($\mathfrak{A}_{\text{span}}$) = **Span**(**Cat**) of functorial spans.

Each model of linear logic is derived from a specific internal category \mathfrak{A} of scheduling in $\mathbb{S} = \text{Cat}$:

- the internal category $\mathfrak{A}_{\text{alt}}$ of sequential and alternating schedulings,
- the internal category $\mathfrak{A}_{\text{conc}}$ of concurrent and non-alternating schedulings,
- the internal category $\mathfrak{A}_{\text{span}}$ of no scheduling whatsoever.

In each case, the internal category \mathfrak{A} of polarities and scheduling should be regarded as a sophisticated synchronization algebra [Winskel 1986] in the sense of process calculi, regulating a specific flavor of interaction (alternating, non-alternating, purely positional). One main discovery of our work is that each of the three synchronization templates \mathfrak{A} organizes itself into an internal category with sufficient structure in order to define bicategorical models of linear logic. We believe that this observation clarifies in what sense game semantics is a carefully regulated form of process calculus, specifically designed in order to produce compositional models of programming languages.

Another important contribution of the paper is the discovery that one needs to shift from an *asynchronous* to a *synchronous* definition of the copycat strategy $\text{cc}_A : A \rightarrow A$ in the construction of the bicategory **Games**($-$). The ability to change the granularity of moves, and to define strategies playing two moves *simultaneously* in different games, is certainly one technical breakthrough of the paper. In particular, synchronizing Opponent and Player moves in the definition of the copycat strategy enables us to construct *for the first time* a symmetric monoidal closed bicategory **Games**($\mathfrak{A}_{\text{conc}}$) of concurrent and non-alternating strategies, where the copycat strategy $\text{cc}_A : A \rightarrow A$ does not behave as a buffer. In particular, we avoid in that way to take the notions of *ingenuous* and *receptive* strategies designed in [Melliès and Mimram 2007, 2008] and adopted in [Rideau and Winskel 2011] as primitive notions. We believe that this is a major improvement in our current understanding of the theory of concurrent non-alternating games.

1.8 Related Works

This paper is our contribution to a research program initiated by Martin Hyland about twenty years ago [Hyland and Schalk 1999, 2003]. The general challenge is to understand the nature of

the categorical combinatorics involved in the construction of categories of games and strategies. Our work is in that respect strongly inspired by Hyland’s idea developed with Harmer and Mellies [Harmer et al. 2007] that the category of simple games and alternating strategies is secretly regulated by a combinatorial category Υ of schedulings. Another important influence has been the work by André, Michel and Tom Hirschowitz [Hirschowitz et al. 2007] where the two categories $\mathfrak{g}_{\text{game}}$ and $\mathfrak{g}_{\text{strat}}$ appear explicitly, with similar regulatory purposes. It should be noted that the categories $\mathfrak{g}_{\text{game}}$, $\mathfrak{g}_{\text{strat}}$ and $\mathfrak{g}_{\text{int}}$ also appear as state diagrams in Harmer’s PhD thesis [Harmer 1999]. As a matter of fact, the turning point of our work has been the discovery that $\mathfrak{g}_{\text{game}}$ and $\mathfrak{g}_{\text{strat}}$ define an internal category in $\mathbb{S} = \mathbf{Cat}$ where the category $\mathfrak{g}_{\text{int}}$ of interactions between sequential alternating strategies is computed as the pullback diagram (14). In this way, we were able to put all these strands together and to “internalize” the category Υ of scheduling in [Harmer et al. 2007] into the internal category $\mathfrak{g}_{\text{alt}}$ derived from [Hirschowitz et al. 2007].

Another decisive source of inspiration has been the construction by Mellies and Mimram of a category of *ingenuous* and *receptive* strategies [Melliès and Mimram 2007, 2008] played on event structures with polarity, and its adaptation by Rideau and Winskel to the language of spans [Rideau and Winskel 2011]. One strong limitation of the original model [Melliès and Mimram 2007] inherited by [Rideau and Winskel 2011] is the fact that the identity strategy $\mathbf{buff}_A : A \rightarrow A$ is defined there as a buffer rather than as a synchronous copycat strategy. The problem is inherent to the purely asynchronous nature of the model, and to the lack of synchronization between events (moves) in event structures. We resolve the problem here by recasting the game model inside the more expressive language of concurrency theory, based on synchronization algebras. Interestingly, this shift requires to move away from the category of event structures and to work in the category $\mathbb{S} = \mathbf{Cat}$ of small categories. One main reason is that our definition (15) of the synchronous copycat strategy $\mathbf{cc}_A : A \rightarrow A$ relies on the existence in (13) of the functor $\mathit{copycat} : \mathfrak{g}_{\text{game}} \rightarrow \mathfrak{g}_{\text{strat}}$ which transports the trajectory O (resp. P) of length 1 in the category $\mathfrak{g}_{\text{game}}$ to the trajectory $O_t \cdot P_s$ (resp. $O_s \cdot P_t$) of length 2 in the category $\mathfrak{g}_{\text{strat}}$. Now, as it happens, a morphism $E \rightarrow F$ in the category of event structures is traditionally defined as a partial map of events; such a map cannot increase the length of trajectories in the asynchronous graphs of configurations $\mathcal{G}(E)$ and $\mathcal{G}(F)$ associated to E and F , see [Melliès 2006] for details and terminology, and thus cannot implement the operation $\mathit{copycat}$ on event structures. Looking in retrospect, and following a generic recipe described in [Selinger 1999], we conjecture that the original category (or bicategory) of receptive and ingenuous strategies constructed in [Melliès and Mimram 2007; Rideau and Winskel 2011] can be recovered as a specific sub-(bi)-category of the Karoubi envelope of $\mathbf{Games}(\mathfrak{g}_{\text{conc}})$, with objects defined as the idempotent and asynchronous copycat strategies $\mathbf{buff}_A : A \rightarrow A$. We leave that question for future work.

In their recent work on session types and game semantics, Castellan and Yoshida [Castellan and Yoshida 2019] develop a truly concurrent and fully abstract interpretation (for barbed congruence) of the synchronous session π -calculus. This technical tour de force is achieved by constructing a game model based on *coincident* event structures where, following the same track as we do in the present paper, the *asynchronous* (ingenuous and receptive) copycat strategies are replaced by *synchronous* copycat strategies. The idea of using *coincident* event structures in order to model synchrony in computations originates from [Ghica and Menea 2011] where a synchronous model of Syntactic Control of Interference is constructed. Coincident event structures are event structures where the causal order \leq on events is replaced by a causal preorder, and where two events e and e' are considered synchronized when they produce a causal loop $e \leq e' \leq e$. Despite the apparent difference in the way the synchronous copycat strategies are constructed in the two models, the game model of session types in [Castellan and Yoshida 2019] is in fact very similar to

the game model $\mathbf{Games}(\multimap_{\text{conc}})$ associated to the synchronization template \multimap_{conc} for concurrent and non-alternating games. Building on this timely point of convergence, we believe that a mild variation of the template \multimap_{conc} would enable us to express in another way the fully complete model designed by Castellan and Yoshida for the synchronous session π -calculus. We leave that interesting methodological question for future work, as well as a more detailed comparison between the two models.

1.9 Synopsis of the Paper

We start by recalling in §2 the notions of *internal category* and of *internal functor* in a category \mathbb{S} with finite limits. We then construct the bicategory $\mathbf{Games}(\multimap)$ of games, strategies and simulations associated to such an internal \mathbb{S} -category \multimap . The construction of $\mathbf{Games}(\multimap)$ is then integrated in §3 to the construction of a double pseudo category $\mathbf{Games}(\multimap)$. This enables us to establish (Thm. 3.5) that every acute span of internal functors between internal categories \multimap_1 and \multimap_2 induces a homomorphism between their respective double pseudo categories $\mathbf{Games}(\multimap_1)$ and $\mathbf{Games}(\multimap_2)$. After an intermezzo in §4 on the construction of the tensor product of alternating games, we establish in §5 that $\mathbf{Games}(-)$ defines a lax symmetric monoidal homomorphism from the symmetric monoidal bicategory $\mathbf{AcuteSpan}(\mathbf{Cat}(\mathbb{S}))$ of acute spans of internal functors between internal categories, to the cartesian 2-category $\mathbf{DoubleCat}$ of double categories and vertical natural transformations. From this very abstract statement follows the important fact that every symmetric pseudomonoid \multimap in the bicategory $\mathbf{AcuteSpan}(\mathbf{Cat}(\mathbb{S}))$, what we call an internal *symmetric span-monoidal category*, defines a symmetric monoidal bicategory $\mathbf{Games}(\multimap)$. We also explicate the additional conditions on \multimap which ensure that the symmetric monoidal bicategory $\mathbf{Games}(\multimap)$ is monoidal closed, and establish the main result of the paper (Thm. 5.6). We illustrate the theorem in §6, §7 and §8 by constructing in the category $\mathbb{S} = \mathbf{Cat}$ of small categories three different bicategorical models of linear logic based either on sequential alternating games in the case \multimap_{alt} , concurrent non-alternating games in the case of \multimap_{conc} or functorial spans in the case of \multimap_{span} . We explain in §9 why the three resulting bicategories $\mathbf{Games}(\multimap_{\text{alt}})$, $\mathbf{Games}(\multimap_{\text{conc}})$ and $\mathbf{Games}(\multimap_{\text{span}})$ are in fact $*$ -autonomous, and conclude in §10.

2 INTERNAL CATEGORIES AND THEIR ASSOCIATED BICATEGORY OF SPANS

In this section and in the next one (§3), we suppose given a category \mathbb{S} with finite limits.

2.1 The Bicategory of Spans

Definition 2.1 (spans). A span $(S, s, t) : A \dashv\vdash B$ between two objects A and B in the category \mathbb{S} consists of an object S together with a pair of maps

$$A \xleftarrow{s} S \xrightarrow{t} B$$

The object S is called the support of the span, while the maps s and t are called its source and target, respectively.

Definition 2.2 ($\mathbf{Span}(A, B)$). Every pair of objects A, B induces a category $\mathbf{Span}(A, B)$ whose objects are the spans between A and B , and whose maps

$$\theta : (S, s_S, t_S) \longrightarrow (T, s_T, t_T)$$

are the maps $\theta : S \rightarrow T$ of \mathbb{S} making the diagram below commute:

$$\begin{array}{ccc}
 & S & \\
 s_S \swarrow & \downarrow \theta & \searrow t_S \\
 A & & B \\
 s_T \swarrow & & \searrow t_T \\
 & T &
 \end{array}$$

Because the category \mathbb{S} has finite limits, there exists a functor

$$\bullet_{A,B,C} : \mathbf{Span}(B,C) \times \mathbf{Span}(A,B) \longrightarrow \mathbf{Span}(A,C)$$

for all objects A,B,C . The functor is defined by putting the spans (S, s_S, t_S) in $\mathbf{Span}(A,B)$ and (T, s_T, t_T) in $\mathbf{Span}(B,C)$ side by side, and by computing the pullback diagram below:

$$\begin{array}{ccccc}
 & & S \times_B T & & \\
 & \pi_1 \swarrow & & \searrow \pi_2 & \\
 & S & pb & T & \\
 s_S \swarrow & & & & \searrow t_T \\
 A & & B & & C
 \end{array}$$

The composite of (S, s_S, t_S) and (T, s_T, t_T) is then defined as

$$(T, s_T, t_T) \bullet_{A,B,C} (S, s_S, t_S) := (S \times_B T, s_S \circ \pi_1, t_T \circ \pi_2)$$

This composition law defines a bicategory $\mathbf{Span}(\mathbb{S})$ whose objects are the objects of \mathbb{S} . The identity span \mathbf{id}_A is defined as the span

$$A \xleftarrow{id} A \xrightarrow{id} A$$

2.2 Internal Categories

For every object A of the category \mathbb{S} , the category $\mathbf{Span}(A,A)$ comes equipped with a monoidal structure, provided by the composition law $\bullet_A := \bullet_{A,A,A}$ and the identity span \mathbf{id}_A .

Definition 2.3 (internal category, \mathbb{S} -category). An internal category \mathfrak{A} in the category \mathbb{S} with finite limits is a span

$$\mathfrak{A}[0] \xleftarrow{s} \mathfrak{A}[1] \xrightarrow{t} \mathfrak{A}[0] \tag{16}$$

equipped with the structure of a *monoid object* in the category

$$\mathbf{Span}(\mathfrak{A}[0], \mathfrak{A}[0])$$

with tensor product $\bullet_{\mathfrak{A}[0]}$ and unit $\mathbf{id}_{\mathfrak{A}[0]}$.

This monoid structure is provided by a pair of maps in the category \mathbb{S}

$$m : \mathfrak{A}[2] \rightarrow \mathfrak{A}[1] \quad e : \mathfrak{A}[0] \rightarrow \mathfrak{A}[1]$$

where the object $\mathfrak{A}[2]$ is defined as the pullback in \mathbb{S} of the diagram below:

$$\begin{array}{ccccc}
 & & \mathfrak{A}[2] & & \\
 & \pi_1 \swarrow & & \searrow \pi_2 & \\
 & \mathfrak{A}[1] & pb & \mathfrak{A}[1] & \\
 s \swarrow & & & & \searrow t \\
 \mathfrak{A}[0] & & \mathfrak{A}[0] & & \mathfrak{A}[0]
 \end{array} \tag{17}$$

The maps m and e satisfy moreover the equations of associativity and neutrality required of a multiplication and of its neutral element in a monoid. We often find convenient to call \mathbb{S} -category an internal category in the category \mathbb{S} with finite limits. We also like to call *spaces* the objects of the category \mathbb{S} . We recall the standard terminology, which we adapt only slightly:

- the object $\mathfrak{A}[0]$ is called the space of *objects*,
- the object $\mathfrak{A}[1]$ is called the space of *maps*,
- the map $m : \mathfrak{A}[2] \rightarrow \mathfrak{A}[1]$ is called the *composition law*,
- the map $e : \mathfrak{A}[0] \rightarrow \mathfrak{A}[1]$ is called the *identity law*

of the internal category or \mathbb{S} -category \mathfrak{A} .

2.3 Internal Functors

We recall now the notion of internal functor between internal categories, a notion which we often find convenient to call a \mathbb{S} -functor between \mathbb{S} -categories.

Definition 2.4 (internal functor, \mathbb{S} -functor). A \mathbb{S} -functor, or internal functor,

$$F : \mathfrak{A}_1 \longrightarrow \mathfrak{A}_2 \quad (18)$$

of \mathbb{S} -categories is a pair of maps of \mathbb{S}

$$F[0] : \mathfrak{A}_1[0] \longrightarrow \mathfrak{A}_2[0] \quad F[1] : \mathfrak{A}_1[1] \longrightarrow \mathfrak{A}_2[1]$$

making the diagram below commute

$$\begin{array}{ccccc} \mathfrak{A}_1[0] & \xleftarrow{s} & \mathfrak{A}_1[1] & \xrightarrow{t} & \mathfrak{A}_1[0] \\ F[0] \downarrow & & \downarrow F[1] & & \downarrow F[0] \\ \mathfrak{A}_2[0] & \xleftarrow{s} & \mathfrak{A}_2[1] & \xrightarrow{t} & \mathfrak{A}_2[0] \end{array}$$

and moreover compatible with the composition law and the identity law of the internal categories \mathfrak{A}_1 and \mathfrak{A}_2 , in the sense that the diagrams below commute:

$$\begin{array}{ccc} \mathfrak{A}_1[2] & \xrightarrow{m_{\mathfrak{A}_1}} & \mathfrak{A}_1[1] \\ F[2] \downarrow & & \downarrow F[1] \\ \mathfrak{A}_2[2] & \xrightarrow{m_{\mathfrak{A}_2}} & \mathfrak{A}_2[1] \end{array} \quad \begin{array}{ccc} \mathfrak{A}_1[0] & \xrightarrow{e_{\mathfrak{A}_1}} & \mathfrak{A}_1[1] \\ F[0] \downarrow & & \downarrow F[1] \\ \mathfrak{A}_2[0] & \xrightarrow{e_{\mathfrak{A}_2}} & \mathfrak{A}_2[1] \end{array}$$

where $F[2] : \mathfrak{A}_1[2] \rightarrow \mathfrak{A}_2[2]$ is the map induced from $F[0]$ and $F[1]$ by the universality property of $\mathfrak{A}_2[2] = \mathfrak{A}_2[1] \times_{\mathfrak{A}_2[0]} \mathfrak{A}_2[1]$.

We will make great usage of the well-known fact that

PROPOSITION 2.5. *The category $\text{Cat}(\mathbb{S})$ of \mathbb{S} -categories and \mathbb{S} -functors has finite limits.*

Note in particular that all the finite limits (and in particular the cartesian product) of internal categories are defined pointwise.

2.4 Every Internal Category Induces a Grothendieck Fibration

Jean Bénabou observed in [Bénabou 1985] that every internal category \mathfrak{A} in a category \mathbb{S} with finite limits induces a Grothendieck fibration

$$p_{\mathfrak{A}} : \mathbb{S}[\mathfrak{A}] \longrightarrow \mathbb{S}$$

defined in the following way. The objects of $\mathbb{S}[\pm]$ are the pairs (A, λ_A) consisting of an object A of the category \mathbb{S} together with a map

$$\lambda_A : A \longrightarrow \pm[0]$$

A map in the category $\mathbb{S}[\pm]$ between two such objects

$$(f, \lambda_f) : (A, \lambda_A) \longrightarrow (B, \lambda_B)$$

is defined as a pair of maps

$$f : A \longrightarrow B \quad \lambda_f : A \longrightarrow \pm[1]$$

of the category \mathbb{S} , making the diagram below commute

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ \lambda_A \downarrow & \searrow \lambda_f & \downarrow \lambda_B \\ \pm[0] & \xleftarrow{s} \pm[1] \xrightarrow{t} & \pm[0] \end{array}$$

The Grothendieck fibration $p_{\pm} : \mathbb{S}[\pm] \rightarrow \mathbb{S}$ transports every object (A, λ_A) to the underlying object A , and every map $(f, \lambda_f) : (A, \lambda_A) \rightarrow (B, \lambda_B)$ to the underlying map $f : A \rightarrow B$ in \mathbb{S} .

2.5 Generalization of the Idea from Maps to Spans

One main idea of the paper is to see the Grothendieck fibration $p_{\pm} : \mathbb{S}[\pm] \rightarrow \mathbb{S}$ associated to an internal category \pm as part of a wider bicategorical structure, defined in the following way. The bicategory $\mathbf{Games}(\pm)$ has the same objects (A, λ_A) as the category $\mathbb{S}[\pm]$, and its maps

$$\sigma = (S, s, t, \lambda_{\sigma}) : (A, \lambda_A) \dashrightarrow (B, \lambda_B) \quad (19)$$

are defined as the spans

$$A \xleftarrow{s} S \xrightarrow{t} B$$

with support S , together with a map $\lambda_{\sigma} : S \rightarrow \pm[1]$ making the diagram below commute:

$$\begin{array}{ccc} A & \xleftarrow{s} S \xrightarrow{t} & B \\ \lambda_A \downarrow & & \downarrow \lambda_{\sigma} \quad \downarrow \lambda_B \\ \pm[0] & \xleftarrow{s} \pm[1] \xrightarrow{t} & \pm[0] \end{array} \quad (20)$$

The 2-cells of the bicategory $\mathbf{Games}(\pm)$ are the *simulations*

$$\theta : \sigma \Longrightarrow \tau : A \dashrightarrow B \quad (21)$$

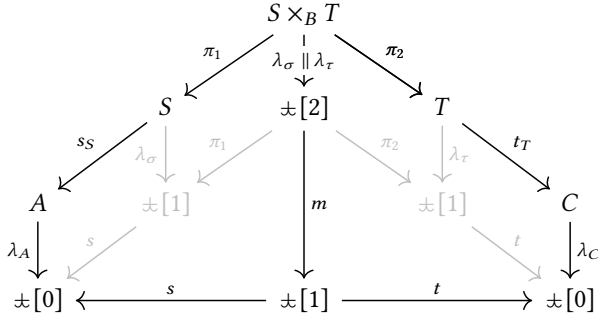
defined as maps $\theta : S \rightarrow T$ making the diagram below commute:

$$\begin{array}{ccc} S & \xrightarrow{\theta} & T \\ \swarrow s & & \searrow s \\ & \pm[0] & \end{array} \quad \begin{array}{ccc} S & \xrightarrow{\theta} & T \\ \swarrow \lambda_{\sigma} & & \searrow \lambda_{\tau} \\ & \pm[1] & \end{array} \quad \begin{array}{ccc} S & \xrightarrow{\theta} & T \\ \swarrow t & & \searrow t \\ & \pm[0] & \end{array}$$

where S is the support of σ and T is the support of τ . Two maps

$$\sigma = (S, s_S, t_S, \lambda_{\sigma}) : A \dashrightarrow B \quad \tau = (T, s_T, t_T, \lambda_{\tau}) : B \dashrightarrow C$$

of $\mathbf{Games}(\mathbb{S})$ are composed in the following way:



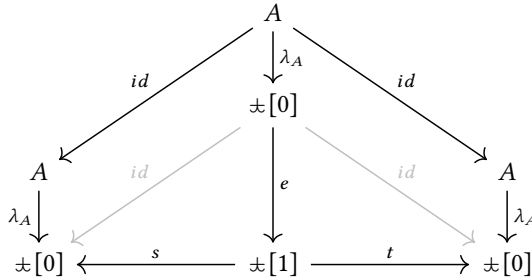
Here, $\lambda_\sigma \parallel \lambda_\tau$ denotes the map of \mathbb{S} induced by the pullback diagram (17) and uniquely determined by the equations

$$\lambda_\sigma \circ \pi_1 = \pi_1 \circ (\lambda_\sigma \parallel \lambda_\tau) \quad \lambda_\tau \circ \pi_2 = \pi_2 \circ (\lambda_\sigma \parallel \lambda_\tau)$$

The identity map

$$\mathbf{id}_A = (A, \mathbf{id}_A, \mathbf{id}_A, e \circ \lambda_A) : A \dashrightarrow A$$

is depicted as



The reader can check that the construction extends to the general setting of an internal category \mathbb{A} in a category with finite limits \mathbb{S} the construction of the bicategory \mathbf{Games} performed in the introduction for the specific internal category \mathbb{A}_{alt} .

3 HOW INTERNAL FUNCTORS ACT ON GAMES AND STRATEGIES

Now that we have associated a bicategory $\mathbf{Games}(\mathbb{A})$ of spans to every internal category \mathbb{A} , we would like to understand how internal functors act on them. To that purpose, we find convenient (and even necessary) to think of the bicategory $\mathbf{Games}(\mathbb{A})$ as part of an even larger structure, provided by a pseudo double category $\underline{\mathbf{Games}}(\mathbb{A})$ with same objects and (horizontal) maps. The reason is essentially technical, driven by the fact that pseudo double categories organise themselves into a 2-category, which is not the case unfortunately of bicategories. So, we introduce in §3.1 the pseudo double category $\underline{\mathbf{Games}}(\mathbb{A})$ and explain in §3.2 and §3.3 how to perform change of internal category along an internal functor. This leads us to the construction in §3.4 of a homomorphism from a bicategory $\mathbf{AcuteSpan}(\mathbf{Cat}(\mathbb{S}))$ of spans of internal functors in \mathbb{S} to the 2-category $\mathbf{DoubleCat}$ of double categories and vertical natural transformations.

Terminology. We find useful at this stage to instill our game-theoretic intuitions to the categorical constructions, and we will thus sometimes call *games*, *strategies* and *simulations* the objects, horizontal maps and double cells of $\underline{\mathbf{Games}}(\mathbb{A})$, respectively.

3.1 The Pseudo Double Category $\underline{\mathbf{Games}}(\pm)$ of Games, Strategies and Simulations

A pseudo double category \mathcal{V} is defined as

- a vertical category \mathcal{V}_0 of objects and vertical maps,
- a vertical category \mathcal{V}_1 of horizontal maps and double cells,
- source and target functors $s, t : \mathcal{V}_1 \rightarrow \mathcal{V}_0$,

together with horizontal unit and composition functors, associative up to special natural isomorphisms, these data satisfying moreover a number of expected coherence diagrams. We refer the reader to Garner's PhD thesis [Garner 2006] for a detailed definition of the notion of pseudo double category.

The pseudo double category $\underline{\mathbf{Games}}(\pm)$ associated to an internal category \pm in the category \mathbb{S} is defined as follows: its objects and its horizontal maps are the same as the objects and maps of the bicategory $\mathbf{Games}(\pm)$, while its vertical maps from (A, λ_A) to (B, λ_B) are the maps $u : A \rightarrow B$ of the category \mathbb{S} making the diagram below commute

$$\begin{array}{ccc} A & \xrightarrow{\theta} & B \\ \lambda_A \searrow & & \swarrow \lambda_B \\ & \pm_{\text{game}} & \end{array}$$

Given two horizontal maps defined as in (19)

$$\sigma = (S, s_S, t_S, \lambda_\sigma) : A \dashrightarrow B \quad \tau = (T, s_T, t_T, \lambda_\tau) : C \dashrightarrow D$$

the double cells of $\underline{\mathbf{Games}}(\pm)$

$$\begin{array}{ccc} A & \xrightarrow{\sigma} & B \\ u \downarrow & \Downarrow \theta & \downarrow v \\ C & \xrightarrow{\tau} & D \end{array} \quad (22)$$

are the maps $\theta : S \rightarrow T$ of the category \mathbb{S} making the two diagrams commute

$$\begin{array}{ccccc} A & \xleftarrow{s} & S & \xrightarrow{t} & B \\ u \downarrow & & \downarrow \theta & & \downarrow v \\ C & \xleftarrow{s} & T & \xrightarrow{t} & D \end{array} \quad \begin{array}{ccc} S & \xrightarrow{\theta} & T \\ \lambda_\sigma \searrow & & \swarrow \lambda_\tau \\ & \pm_{\text{strat}} & \end{array}$$

Note that a simulation (21) in the bicategory $\mathbf{Games}(\pm)$ is the same thing as a *special* double cell (or simulation) in $\underline{\mathbf{Games}}(\pm)$. Here, by special double cell, we mean a double cell (22) whose vertical maps u and v are identities. For that reason, the bicategory $\mathbf{Games}(\pm)$ coincides with the bicategory canonically obtained from the pseudo double category $\underline{\mathbf{Games}}(\pm)$ by restricting the vertical category to its identities. This shows that no information is lost in the shift from bicategories to double categories.

3.2 Homomorphisms between Double Categories

A morphism between pseudo double categories

$$(\mathcal{F}, p) : \mathcal{V} \longrightarrow \mathcal{W} \quad (23)$$

is a pair of functors

$$\mathcal{F}_0 : \mathcal{V}_0 \longrightarrow \mathcal{W}_0 \quad \mathcal{F}_1 : \mathcal{V}_1 \longrightarrow \mathcal{W}_1$$

and a family of special double cells

$$\begin{array}{ccc}
 \mathcal{F}(A) & \xrightarrow{\mathcal{F}(\sigma)} & \mathcal{F}(B) & \xrightarrow{\mathcal{F}(\tau)} & \mathcal{F}(C) & & \mathcal{F}(A) & \xrightarrow{\mathcal{F}(\text{id}_A)} & \mathcal{F}(A) \\
 \text{id} \downarrow & & \Downarrow p_{\sigma, \tau} & & \downarrow \text{id} & & \text{id} \downarrow & & \Downarrow p_A & & \downarrow \text{id} \\
 \mathcal{F}(A) & \xrightarrow{\mathcal{F}(\tau \circ \sigma)} & \mathcal{F}(C) & & \mathcal{F}(C) & & \mathcal{F}(A) & \xrightarrow{\text{id}_{\mathcal{F}(A)}} & \mathcal{F}(A)
 \end{array} \quad (24)$$

indexed by the horizontal maps σ and τ and the objects A of the double category \mathcal{V} , satisfying the expected coherence relations. The morphism (23) between double categories is called a *homomorphism* when the natural transformations $p_{\sigma, \tau}$ and p_A are isomorphisms for all horizontal maps σ and τ and objects A . Note that the definitions of *morphism* and of *homomorphism* apply also to bicategories, which are specific instances of pseudo double categories. Once again, we refer the reader to [Garner 2006] for a detailed definition of the notion morphism (and homomorphism) between pseudo double categories.

3.3 Internal Functors and their Action on the Double Category of Games

The following observation will play a central role in the paper, since it will enable us to “change basis” and relate double categories of games between them:

PROPOSITION 3.1. *Every \mathbb{S} -functor $F : \mathfrak{A}_1 \rightarrow \mathfrak{A}_2$ of \mathbb{S} -categories induces a homomorphism*

$$F^\triangleright : \underline{\mathbf{Games}}(\mathfrak{A}_1) \longrightarrow \underline{\mathbf{Games}}(\mathfrak{A}_2)$$

and a morphism between double categories

$$F^\triangleleft : \underline{\mathbf{Games}}(\mathfrak{A}_2) \longrightarrow \underline{\mathbf{Games}}(\mathfrak{A}_1)$$

The homomorphism F^\triangleright transports every game, strategy and simulation in $\underline{\mathbf{Games}}(\mathfrak{A}_1)$ into the game, strategy and simulation in $\underline{\mathbf{Games}}(\mathfrak{A}_2)$ obtained by postcomposing with $F[0]$ and $F[1]$ in the following way:

$$\begin{array}{ccccc}
 A & \xleftarrow{s} & S & \xrightarrow{t} & B \\
 \lambda_A \downarrow & & \downarrow \lambda_\sigma & & \downarrow \lambda_B \\
 \mathfrak{A}_1[0] & \xleftarrow{s} & \mathfrak{A}_1[1] & \xrightarrow{t} & \mathfrak{A}_1[0] \\
 F[0] \downarrow & & \downarrow F[1] & & \downarrow F[0] \\
 \mathfrak{A}_2[0] & \xleftarrow{s} & \mathfrak{A}_2[1] & \xrightarrow{t} & \mathfrak{A}_2[0]
 \end{array} \quad (25)$$

Conversely, the homomorphism F^\triangleleft transports every game, strategy and simulation in $\mathbf{Games}(\mathfrak{A}_2)$ into the game, strategy and simulation in $\mathbf{Games}(\mathfrak{A}_1)$ obtained by pulling back along the functors $F[0]$ and $F[1]$, as the reader can easily guess from the diagram below:

$$\begin{array}{ccccc}
 A & \xleftarrow{s} & S & \xrightarrow{t} & B \\
 \lambda_A \downarrow & & \downarrow \lambda_\sigma & & \downarrow \lambda_B \\
 \mathfrak{A}_2[0] & \xleftarrow{s} & \mathfrak{A}_2[1] & \xrightarrow{t} & \mathfrak{A}_2[0] \\
 \uparrow F[0] & & \uparrow F[1] & & \uparrow F[0] \\
 \mathfrak{A}_1[0] & \xleftarrow{s} & \mathfrak{A}_1[1] & \xrightarrow{t} & \mathfrak{A}_1[0]
 \end{array} \tag{26}$$

As indicated in Prop. 3.1, the pullback construction does not define a homomorphism of bicategories in general, but just a morphism, because composition is preserved only up to a special double cells of the form (24). We characterize in the statement below when this natural transformation is invertible, and thus defines a homomorphism of bicategory.

PROPOSITION 3.2. *The morphism of double category $F^\triangleleft : \mathbf{Games}(\mathfrak{A}_2) \rightarrow \mathbf{Games}(\mathfrak{A}_1)$ associated to a \mathbb{S} -functor $F : \mathfrak{A}_1 \rightarrow \mathfrak{A}_2$ is a homomorphism if and only if the two diagrams below are pullbacks:*

$$\begin{array}{ccc}
 \mathfrak{A}_1[2] & \xrightarrow{F[2]} & \mathfrak{A}_2[2] \\
 m_1 \downarrow & (a) & \downarrow m_2 \\
 \mathfrak{A}_1[1] & \xrightarrow{F[1]} & \mathfrak{A}_2[1]
 \end{array}
 \qquad
 \begin{array}{ccc}
 \mathfrak{A}_1[0] & \xrightarrow{F[0]} & \mathfrak{A}_2[0] \\
 e_1 \downarrow & (b) & \downarrow e_2 \\
 \mathfrak{A}_1[1] & \xrightarrow{F[1]} & \mathfrak{A}_2[1]
 \end{array} \tag{27}$$

We will use the following terminology:

Definition 3.3. A \mathbb{S} -functor $F : \mathfrak{A}_1 \rightarrow \mathfrak{A}_2$ is acute when it satisfies the pullback condition (27).

A useful observation for later usage is that

PROPOSITION 3.4. *The class of acute \mathbb{S} -functors is closed under pullbacks in the category $\mathbf{Cat}(\mathbb{S})$.*

3.4 Acute Spans of Internal Categories

The statements of Prop. 2.5 and 3.4 lead us to define the bicategory $\mathbf{AcuteSpan}(\mathbf{Cat}(\mathbb{S}))$ of acute spans in the category $\mathbf{Cat}(\mathbb{S})$ with finite limits. By *acute span*, we mean a span in $\mathbf{Cat}(\mathbb{S})$

$$\mathfrak{A}_1 \xleftarrow{F_s} \mathfrak{A}_S \xrightarrow{F_t} \mathfrak{A}_2 \tag{28}$$

where the two \mathbb{S} -functors F_s and F_t are acute in the sense of Def. 3.3. As expected, the 2-dimensional cells

$$H : \left(\mathfrak{A}_1 \xleftarrow{F_s} \mathfrak{A}_S \xrightarrow{F_t} \mathfrak{A}_2 \right) \Longrightarrow \left(\mathfrak{A}_1 \xleftarrow{G_s} \mathfrak{A}_T \xrightarrow{G_t} \mathfrak{A}_2 \right) : \mathfrak{A}_1 \dashrightarrow \mathfrak{A}_2$$

in the bicategory $\mathbf{AcuteSpan}(\mathbf{Cat}(\mathbb{S}))$ are the acute \mathbb{S} -functors $H : \mathfrak{A}_S \rightarrow \mathfrak{A}_T$ making the diagram below commute:

$$\begin{array}{ccc}
 & \mathfrak{A}_S & \\
 F_s \swarrow & \downarrow & \searrow F_t \\
 A & H & B \\
 G_t \swarrow & \downarrow & \searrow G_s \\
 & \mathfrak{A}_T &
 \end{array}$$

We can easily deduce from Prop. 3.1 that there exists a pair of functors

$$\begin{aligned}
 (-)^\triangleright &= F \mapsto F^\triangleright : \mathbf{Cat}(\mathbb{S}) \longrightarrow \mathbf{DoubleCat} \\
 (-)^\triangleleft &= F \mapsto F^\triangleleft : \mathbf{Cat}(\mathbb{S})^{op} \longrightarrow \mathbf{DoubleCat}
 \end{aligned}$$

We establish that

THEOREM 3.5. *The two functors $(-)^\triangleleft$ and $(-)^\triangleright$ combine into a homomorphism of bicategories*

$$\underline{\mathbf{Games}}(-) : \mathbf{AcuteSpan}(\mathbf{Cat}(\mathbb{S})) \longrightarrow \mathbf{DoubleCat} \quad (29)$$

Every acute span (28) is mapped to the homomorphism of double categories defined as the composite

$$\underline{\mathbf{Games}}(\mathfrak{A}_1) \xrightarrow{F_s^\triangleleft} \underline{\mathbf{Games}}(\mathfrak{A}_S) \xrightarrow{F_t^\triangleright} \underline{\mathbf{Games}}(\mathfrak{A}_2) \quad (30)$$

A Beck-Chevalley property ensures then that the two constructions behave well with respect to composition, and induce the announced homomorphism (29) of bicategories.

4 INTERMEZZO: THE TENSOR PRODUCT OF ALTERNATING GAMES

We have seen in the introduction that $\mathfrak{A}_{\text{alt}}$ defines an internal category in $\mathbb{S} = \mathbf{Cat}$. From this combinatorial data in $\mathbb{S} = \mathbf{Cat}$ follows the existence of the bicategory $\mathbf{Games}(\mathfrak{A}_{\text{alt}})$ of alternating games, strategies and simulations described in §2. In this short section, we explain how one can construct the tensor product $A \otimes B$ of two alternating games A and B using “change of internal categories” along acute spans of internal categories, following there the philosophy developed in the previous section §3.

4.1 The Internal Category of Tensor Polarities

We introduce the internal category $\mathfrak{A}_{\text{alt}}^\otimes$ of *tensor polarities*. Its first component (space of objects)

$$\mathfrak{A}_{\text{alt}}^\otimes[0] = \mathfrak{A}_{\text{game}}^\otimes$$

is defined as the category freely generated by the graph below:

$$\langle \ominus, \oplus \rangle \xleftarrow[p_l]{O_l} \langle \oplus, \oplus \rangle \xleftarrow[p_r]{O_r} \langle \oplus, \ominus \rangle$$

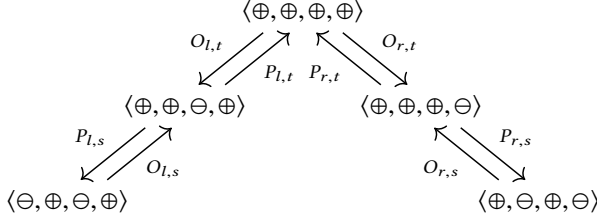
The three objects of the category $\mathfrak{A}_{\text{game}}^\otimes$ describe the three polarities

$$\langle \lambda_{A_1}(x_1), \lambda_{A_2}(x_2) \rangle$$

possibly reached by a position (x_1, x_2) in the game $A_1 \otimes A_2$ obtained (as we will see) by tensoring the games A_1 and A_2 . The next component (space of maps) of the internal category

$$\mathfrak{A}_{\text{alt}}^\otimes[1] = \mathfrak{A}_{\text{strat}}^\otimes$$

is defined as the category freely generated by the graph below:



Here, the five objects of the category describe the five possible sequences of polarities

$$\langle \lambda_{A_1}(x_1), \lambda_{A_2}(x_2), \lambda_{A_3}(x_3), \lambda_{A_4}(x_4) \rangle$$

possibly reached by a position (x_1, x_2, x_3, x_4) of the four games A_1, A_2, A_3, A_4 during a sequential and alternating execution of a strategy σ of the form

$$\sigma : A_1 \otimes A_2 \longrightarrow A_3 \otimes A_4 \quad (31)$$

Typically, the edge $O_{r,t}$ describes the polarity of an Opponent move playing on the component A_4 and thus transporting a position of polarity $\langle \oplus, \oplus, \oplus, \oplus \rangle$ into a position of polarity $\langle \oplus, \oplus, \oplus, \ominus \rangle$. The scheduling then allows Player to play on the component A_2 with a move of polarity $P_{r,s}$ which reaches a position of polarity $\langle \oplus, \ominus, \oplus, \ominus \rangle$; or to play on the component A_4 with a move of polarity $P_{r,s}$ which goes back to a position of polarity $\langle \oplus, \oplus, \oplus, \oplus \rangle$.

4.2 The Associated Span of Internal Functors

It appears that the internal category $\mathfrak{A}_{\text{alt}}^{\otimes}$ comes equipped with a span of internal functors

$$\mathfrak{A}_{\text{alt}} \times \mathfrak{A}_{\text{alt}} \xleftarrow{F_s^{\otimes}} \mathfrak{A}_{\text{alt}}^{\otimes} \xrightarrow{F_t^{\otimes}} \mathfrak{A}_{\text{alt}} \quad (32)$$

each of them defined by a pair of functors (ie. maps in $\mathbb{S} = \text{Cat}$) below:

$$\begin{array}{ccc} \mathfrak{A}_{\text{game}} \times \mathfrak{A}_{\text{game}} & \xleftarrow{F_s^{\otimes}[0]=(1)(2)} & \mathfrak{A}_{\text{game}}^{\otimes} \xrightarrow{F_t^{\otimes}[0]} \mathfrak{A}_{\text{game}} \\ \mathfrak{A}_{\text{strat}} \times \mathfrak{A}_{\text{strat}} & \xleftarrow{F_s^{\otimes}[1]=(13)(24)} & \mathfrak{A}_{\text{strat}}^{\otimes} \xrightarrow{F_t^{\otimes}[1]} \mathfrak{A}_{\text{strat}} \end{array}$$

The functor $F_s^{\otimes}[0]$ is noted (1)(2) because it projects every position $\langle \epsilon_1, \epsilon_2 \rangle$ of $\mathfrak{A}_{\text{game}}^{\otimes}$ to its first and second components ϵ_1 and ϵ_2 , with each of them transported to the first and second component of the cartesian product $\mathfrak{A}_{\text{game}} \times \mathfrak{A}_{\text{game}}$. In other words:

$$F_s^{\otimes}[0] : \langle \epsilon_1, \epsilon_2 \rangle \mapsto (\langle \epsilon_1 \rangle, \langle \epsilon_2 \rangle)$$

where $\epsilon_1, \epsilon_2 \in \{\oplus, \ominus\}$. Similarly, the functor $F_s^{\otimes}[1]$ is noted (13)(24) because it behaves as follows:

$$F_s^{\otimes}[1] : \langle \epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4 \rangle \mapsto (\langle \epsilon_1, \epsilon_3 \rangle, \langle \epsilon_2, \epsilon_4 \rangle)$$

The purpose of $F_s^{\otimes}[0]$ and $F_s^{\otimes}[1]$ is to “disentangle” the two threads of computation running on the left and the right component of the tensor product, both at the level of games for $F_s^{\otimes}[0]$ and at the level of strategies for $F_s^{\otimes}[1]$. The internal functor F_t^{\otimes} performs the complementary task of assigning a polarity in $\mathfrak{A}_{\text{alt}}$ to every polarity in $\mathfrak{A}_{\text{alt}}^{\otimes}$, according to the table of polarity expected for the tensor product of two alternating games. By this, we mean that the functor $F_t^{\otimes}[0]$ assigns a polarity in $\mathfrak{A}_{\text{game}}$ to every position and trajectory of $\mathfrak{A}_{\text{game}}^{\otimes}$ according to the following table:

$$\begin{array}{ccc} \langle \oplus, \oplus \rangle & \mapsto & \langle \oplus \rangle & O_s \mapsto O & O_t \mapsto O \\ \langle \ominus, \oplus \rangle, \langle \oplus, \ominus \rangle & \mapsto & \langle \ominus \rangle & P_s \mapsto P & P_t \mapsto P \end{array}$$

and that the functor $F_t^\otimes[1]$ assigns a polarity in $\mathfrak{A}_{\text{strat}}$ to every position of $\mathfrak{A}_{\text{strat}}^\otimes$ as follows:

$$\begin{array}{llll} \langle \oplus, \oplus, \oplus, \oplus \rangle & \mapsto & \langle \oplus, \oplus \rangle & \\ \langle \oplus, \oplus, \oplus, \ominus \rangle, \langle \oplus, \oplus, \ominus, \oplus \rangle & \mapsto & \langle \oplus, \ominus \rangle & O_{l,t}, O_{r,t} \mapsto O_t \quad P_{r,t}, P_{l,t} \mapsto P_t \\ \langle \oplus, \ominus, \oplus, \ominus \rangle, \langle \ominus, \oplus, \ominus, \oplus \rangle & \mapsto & \langle \ominus, \ominus \rangle & O_{l,s}, O_{r,s} \mapsto O_s \quad P_{r,s}, P_{l,s} \mapsto P_s \end{array}$$

Going back to the definition of acute \mathbb{S} -functors formulated in §3.3 (Def. 3.3), a nice and elementary combinatorial exercise establishes that

PROPOSITION 4.1. *The two components F_s^\otimes and F_t^\otimes of the span (32) are acute \mathbb{S} -functors.*

4.3 The Construction of the Tensor Product

We are finally ready to explain how to use the combinatorial and categorical material just introduced in order to construct the tensor product of two alternating games (A, λ_A) and (B, λ_B) . The idea is to compute the pullback diagram (a) below

$$\begin{array}{ccccc} A \times B & \xleftarrow{\text{inclusion}} & A \otimes B & & \\ \lambda_A \times \lambda_B \downarrow & & \downarrow \lambda & \searrow \lambda_{A \otimes B} & \\ \mathfrak{A}_{\text{game}} \times \mathfrak{A}_{\text{game}} & \xleftarrow{F_s^\otimes[0]=(1)(2)} & \mathfrak{A}_{\text{game}}^\otimes & \xrightarrow{F_t^\otimes[0]} & \mathfrak{A}_{\text{game}} \end{array} \quad (33)$$

in order to compute the category $A \otimes B$ as well as the polarity functor $\lambda : A \otimes B \rightarrow \mathfrak{A}_{\text{game}}^\otimes$ to the category of tensor polarities. The polarity functor $\lambda_{A \otimes B} : A \otimes B \rightarrow \mathfrak{A}_{\text{game}}^\otimes$ itself is then obtained by postcomposing the functor λ with the functor $F_t^\otimes[0]$. Note that $\mathfrak{A}_{\text{game}}^\otimes$ may be seen as a subcategory of $\mathfrak{A}_{\text{game}} \times \mathfrak{A}_{\text{game}}$ with $F_s^\otimes[0]$ as inclusion functor. For that reason, $A \otimes B$ may be defined as the subcategory of $A \times B$ whose positions and trajectories have the appropriate polarities and scheduling.

What should strike the reader at this stage is that the very concrete construction of the tensor product $A, B \mapsto A \otimes B$ of two alternating games just described can be rephrased in the more abstract and sophisticated language developed in the previous section §3. Indeed, the construction follows the recipe of Thm. 3.5 which states that the span of two acute \mathbb{S} -functors (32) induces a homomorphism of double categories

$$\mathbf{Games}(\mathfrak{A}_{\text{alt}} \times \mathfrak{A}_{\text{alt}}) \xrightarrow{(F_s^\otimes)^\triangleleft} \mathbf{Games}(\mathfrak{A}_{\text{alt}}^\otimes) \xrightarrow{(F_t^\otimes)^\triangleright} \mathbf{Games}(\mathfrak{A}_{\text{alt}}) \quad (34)$$

defined as the composite of pullback along F_s^\otimes and postcomposition along F_t^\otimes . This observation leads us to the search for a purely combinatorial axiomatics on $\mathfrak{A}_{\text{alt}}$ which would ensure that the tensor product just defined turns $\mathbf{Games}(\mathfrak{A}_{\text{alt}})$ into a symmetric monoidal closed double category $\mathbf{Games}(\mathfrak{A}_{\text{alt}})$. Note that this statement would be outrageously difficult to establish directly without the dictionary between internal categories and double categories developed in the previous two sections.

5 SPAN-MONOIDAL CLOSED \mathbb{S} -CATEGORIES

In this section, we establish that the bicategory $\mathbf{Games}(\mathfrak{A})$ is symmetric monoidal closed when the \mathbb{S} -category \mathfrak{A} in \mathbb{S} comes equipped with a *span-monoidal closed structure*, a notion which will be defined in the core of the section.

5.1 Lax Symmetric Monoidal Homomorphisms

A lax symmetric monoidal homomorphism (\mathcal{F}, p, m) between symmetric monoidal bicategories

$$(\mathcal{F}, p, m) : (\mathcal{V}, \otimes, I) \longrightarrow (\mathcal{W}, \otimes, I)$$

is a homomorphism (\mathcal{F}, p) equipped with a family of maps

$$m_{A,B} : \mathcal{F}A \otimes \mathcal{F}B \longrightarrow \mathcal{F}(A \otimes B) \quad m_I : I \longrightarrow \mathcal{F}I$$

satisfying a number of coherence relations adapting to symmetric monoidal bicategories the notion of lax symmetric monoidal functor between symmetric monoidal categories.

5.2 The Lax Symmetric Homomorphism $\mathbf{Games}(-)$

The bicategory $\mathbf{AcuteSpan}(\mathbf{Cat}(\mathbb{S}))$ defined in §3.4 is symmetric monoidal, with the cartesian product $A, B \mapsto A \times B$ of the category $\mathbf{Cat}(\mathbb{S})$ of internal categories as tensor product, and with terminal object $\mathbf{1}$ in $\mathbf{Cat}(\mathbb{S})$ as tensorial unit. We find convenient to use the notation \boxtimes for the cartesian product $A, B \mapsto A \times B$ of internal categories, when it is seen as a tensor product in bicategory $\mathbf{Span}(\mathbf{Cat}(\mathbb{S}))$. Similarly, we write $\mathbf{1}$ for the terminal internal category $\mathbf{1}$, seen as the tensor unit of \boxtimes . In comparison, the 2-category $\mathbf{DoubleCat}$ of double categories, homomorphisms and vertical natural transformations is cartesian closed, with cartesian product noted $\mathcal{V}, \mathcal{W} \mapsto \mathcal{V} \times \mathcal{W}$ and terminal double category noted $\mathbf{1}$. We establish the following result, which provides the technical foundations of all later results:

PROPOSITION 5.1. *The homomorphism of bicategories*

$$\mathbf{Games}(-) : (\mathbf{AcuteSpan}(\mathbf{Cat}(\mathbb{S})), \boxtimes, \mathbf{1}) \longrightarrow (\mathbf{DoubleCat}, \times, \mathbf{1})$$

defined in Thm. 3.5 is lax symmetric monoidal.

As indicated in §5.1, this means that $\mathbf{Games}(-)$ comes equipped with a family of homomorphisms of double categories

$$\begin{array}{ccc} m_{\mathfrak{s}_1, \mathfrak{s}_2} & : & \mathbf{Games}(\mathfrak{s}_1) \times \mathbf{Games}(\mathfrak{s}_2) \longrightarrow \mathbf{Games}(\mathfrak{s}_1 \boxtimes \mathfrak{s}_2) \\ m_{\mathbf{1}} & : & \mathbf{1} \longrightarrow \mathbf{Games}(\mathbf{1}) \end{array}$$

indexed by the internal categories \mathfrak{s}_1 and \mathfrak{s}_2 in \mathbb{S} , and satisfying a number of coherence relations. It is worth stressing that the homomorphisms $m_{\mathfrak{s}_1, \mathfrak{s}_2}$ and $m_{\mathbf{1}}$ of bicategories are as simple as they should be. Typically, the homomorphism $m_{\mathfrak{s}_1, \mathfrak{s}_2}$ transports every pair of objects

$$\lambda_A : A \longrightarrow \mathfrak{s}_1[0] \quad \lambda_B : B \longrightarrow \mathfrak{s}_2[0] \tag{35}$$

of the double categories $\mathbf{Games}(\mathfrak{s}_1)$ and $\mathbf{Games}(\mathfrak{s}_2)$ to the object

$$\lambda_A \times \lambda_B : A \times B \longrightarrow (\mathfrak{s}_1 \times \mathfrak{s}_2)[0]$$

of the category $\mathbf{Games}(\mathfrak{s}_1 \boxtimes \mathfrak{s}_2)$ defined by taking the cartesian product of (35) in \mathbb{S} .

5.3 Span-Monoidal Internal Categories

We are now ready to introduce the central notion of the paper, and to establish our main theorem.

Definition 5.2. A symmetric span-monoidal internal category \mathfrak{s} is defined as a symmetric pseudomonoid object $(\mathfrak{s}, \otimes, J)$ in the symmetric monoidal bicategory $\mathbf{AcuteSpan}(\mathbf{Cat}(\mathbb{S}))$.

In other words, a symmetric span-monoidal internal category \mathfrak{s} is an internal category which comes equipped with two spans of acute \mathbb{S} -functors

$$\mathfrak{s} \boxtimes \mathfrak{s} \xrightarrow{\otimes_{\mathfrak{s}}} \mathfrak{s} \qquad \mathbf{1} \xrightarrow{J_{\mathfrak{s}}} \mathfrak{s} \quad (36)$$

satisfying the coherence diagrams of a symmetric pseudomonoid object $(\mathfrak{s}, \otimes, J)$. We are ready now to state our first structure theorem:

THEOREM 5.3. *Every symmetric span-monoidal \mathbb{S} -category \mathfrak{s} induces a symmetric monoidal bicategory $\mathbf{Games}(\mathfrak{s})$.*

The tensor product and unit

$$\mathbf{Games}(\mathfrak{s}) \times \mathbf{Games}(\mathfrak{s}) \xrightarrow{\otimes} \mathbf{Games}(\mathfrak{s}) \qquad \mathbf{1} \xrightarrow{J} \mathbf{Games}(\mathfrak{s})$$

of the double category $\mathbf{Games}(\mathfrak{s})$ are defined as the composite homomorphisms:

$$\begin{array}{c} \mathbf{Games}(\mathfrak{s}) \times \mathbf{Games}(\mathfrak{s}) \xrightarrow{m_{\mathfrak{s}, \mathfrak{s}}} \mathbf{Games}(\mathfrak{s} \boxtimes \mathfrak{s}) \xrightarrow{\mathbf{Games}(\otimes_{\mathfrak{s}})} \mathbf{Games}(\mathfrak{s}) \\ \mathbf{1} \xrightarrow{m_1} \mathbf{Games}(\mathbf{1}) \xrightarrow{\mathbf{Games}(J_{\mathfrak{s}})} \mathbf{Games}(\mathfrak{s}) \end{array}$$

5.4 Span-Monoidal Closed \mathbb{S} -categories

Now that we have established in Thm. 5.3 that every symmetric span-monoidal \mathbb{S} -category \mathfrak{s} induces a symmetric monoidal bicategory $\mathbf{Games}(\mathfrak{s})$, we are interested to find sufficient conditions on \mathfrak{s} in order to ensure that the bicategory $\mathbf{Games}(\mathfrak{s})$ is also closed. Before giving the definition, we observe the following interesting phenomenon: given a \mathbb{S} -category \mathfrak{s} , consider the “opposite” \mathbb{S} -category \mathfrak{s}^{op} obtained by permuting the source and target maps s and t in the definition of \mathfrak{s} . It is essentially immediate that

PROPOSITION 5.4. *The bicategory $\mathbf{Games}(\mathfrak{s}^{op})$ is isomorphic to the bicategory $\mathbf{Games}(\mathfrak{s})^{op(1)}$ obtained by reversing the 1-dimensional cells in the bicategory $\mathbf{Games}(\mathfrak{s})$.*

This observation leads us to the following definition:

Definition 5.5. A span-monoidal \mathbb{S} -category \mathfrak{s} is *closed* when there exists an acute span

$$\mathfrak{s}^{op} \boxtimes \mathfrak{s} \xrightarrow{\overset{\circ}{\otimes}_{\mathfrak{s}}} \mathfrak{s}$$

in the sense of §3.4. This means that there exists a \mathbb{S} -category \mathfrak{s}° equipped with a span of acute \mathbb{S} -functors

$$\mathfrak{s}^{op} \times \mathfrak{s} \xleftarrow{F_s^{\circ}} \mathfrak{s}^{\circ} \xrightarrow{F_t^{\circ}} \mathfrak{s}$$

One requires moreover that the span computed in the category \mathbb{S}

$$\begin{array}{ccccc} & & \mathfrak{s}^{1\otimes 2,3} & & \\ & (12) \swarrow & & \searrow (3) & \\ F_s^{\otimes}[0] & \mathfrak{s}^{\otimes}[0] & pb & \mathfrak{s}[1] & \\ \swarrow & \searrow & \swarrow s & \searrow t & \\ \mathfrak{s}[0] \times \mathfrak{s}[0] & F_t^{\otimes}[0] & \mathfrak{s}[0] & & \mathfrak{s}[0] \end{array} \quad (37)$$

coincides with the span also computed in the category \mathbb{S}

$$\begin{array}{c}
 \begin{array}{ccccc}
 & & \mathbb{I}^{1,2 \dashv 3} & & \\
 & (1) \swarrow & & \searrow (23) & \\
 & \mathbb{I}[1] & pb & \mathbb{I}^{\dashv}[0] & \\
 s \swarrow & & \searrow & \swarrow F_s^{\dashv}[0] & \\
 \mathbb{I}[0] & & \mathbb{I}[0] & \mathbb{I}[0] \times \mathbb{I}[0] & \\
 & \searrow t & \swarrow F_t^{\dashv}[0] & & \\
 & & \mathbb{I}[0] & &
 \end{array}
 \end{array} \tag{38}$$

up to isomorphism, and up to appropriate renaming of the components. Typically, in concrete situations, the objects $\mathbb{I}^{1 \otimes 2,3}$ and $\mathbb{I}^{1,2 \dashv 3}$ would be required to coincide, and the three maps

$$f_1 = \pi_1 \circ F_s^{\otimes}[0] \circ (12) \quad f_2 = \pi_2 \circ F_s^{\otimes}[0] \circ (12) \quad f_3 = t \circ (3)$$

induced by the span (37) would be required to coincide with the three maps induced by the span (38)

$$g_1 = s \circ (1) \quad g_2 = \pi_1 \circ F_s^{\dashv}[0] \circ (23) \quad g_3 = \pi_2 \circ F_s^{\dashv}[0] \circ (23)$$

in the expected sense that $f_1 = g_1$, $f_2 = g_2$ and $f_3 = g_3$.

We establish that

THEOREM 5.6. *Every symmetric span-monoidal closed \mathbb{S} -category \mathbb{I} induces a symmetric monoidal closed bicategory $\mathbf{Games}(\mathbb{I})$.*

The tensorial implication of the pseudo double category $\mathbf{Games}(\mathbb{I})$

$$\dashv : \mathbf{Games}(\mathbb{I})^{op(1)} \times \mathbf{Games}(\mathbb{I}) \longrightarrow \mathbf{Games}(\mathbb{I})$$

is defined as the composite homomorphism:

$$\begin{array}{ccc}
 \mathbf{Games}(\mathbb{I})^{op(1)} \times \mathbf{Games}(\mathbb{I}) & & \mathbf{Games}(\mathbb{I}) \\
 \text{iso} \downarrow & & \uparrow \mathbf{Games}(\dashv_{\mathbb{I}}) \\
 \mathbf{Games}(\mathbb{I}^{op}) \times \mathbf{Games}(\mathbb{I}) & \xrightarrow{m_{\mathbb{I}^{op}, \mathbb{I}}} & \mathbf{Games}(\mathbb{I}^{op} \boxtimes \mathbb{I})
 \end{array}$$

6 BACK TO SEQUENTIAL ALTERNATING GAMES AND STRATEGIES

At this point, we are ready to take all the benefit of the abstract constructions performed in the previous sections, and to illustrate our foundational and principled approach to game semantics by deriving a sequential and alternating game model of linear logic $\mathbf{Games}(\mathbb{I}_{\text{alt}})$ from our main result (Thm. 5.6). The construction is simple, since it boils down to establishing that the span-monoidal structure on \mathbb{I}_{alt} defined in §4 is closed in the sense of Def. 5.5. An easy computation shows that the pullback diagram

$$\begin{array}{c}
 \begin{array}{ccccc}
 & & \mathbb{I}^{1 \otimes 2,3} & & \\
 & \swarrow & & \searrow & \\
 & \mathbb{I}^{\otimes}_{\text{game}} & pb & \mathbb{I}_{\text{strat}} & \\
 F_s^{\otimes}[0] \swarrow & & \searrow & \swarrow s=(1) & \\
 \mathbb{I}_{\text{game}} \times \mathbb{I}_{\text{game}} & & \mathbb{I}_{\text{game}} & & \mathbb{I}_{\text{game}} \\
 & \searrow F_t^{\otimes}[0] & & \swarrow t & \\
 & & \mathbb{I}_{\text{game}} & &
 \end{array}
 \end{array} \tag{39}$$

defines the category $\mathfrak{A}^{1 \otimes 2, 3}$ freely generated by the graph below:

$$\begin{array}{ccc}
 & \langle \oplus, \oplus, \oplus \rangle & \\
 & \begin{array}{c} \downarrow O_3 \\ \uparrow P_3 \end{array} & \\
 & \langle \oplus, \oplus, \ominus \rangle & \\
 \begin{array}{c} \swarrow O_1 \\ \searrow P_1 \end{array} & & \begin{array}{c} \swarrow O_2 \\ \searrow P_2 \end{array} \\
 \langle \ominus, \oplus, \oplus \rangle & & \langle \oplus, \ominus, \oplus \rangle
 \end{array} \tag{40}$$

The four positions of the graph describe the four polarities of a position (x_1, x_2, x_3) possibly reached by a strategy σ of the form

$$\sigma : A_1 \otimes A_2 \longrightarrow A_3 \tag{41}$$

Similarly, one introduces the \mathbb{S} -category $\mathfrak{A}_{\text{alt}}^{\circ}$ defined as follows. Its first component (space of objects) is the category $\mathfrak{A}_{\text{alt}}^{\circ}[0]$ freely generated by the graph

$$\langle \ominus, \ominus \rangle \xrightleftharpoons[O_l]{P_l} \langle \oplus, \ominus \rangle \xrightleftharpoons[O_r]{P_r} \langle \oplus, \oplus \rangle$$

Its second component (space of maps) is the category $\mathfrak{A}_{\text{alt}}^{\circ}[1]$ freely generated by the graph

$$\begin{array}{ccc}
 & \langle \oplus, \ominus, \oplus, \ominus \rangle & \\
 & \begin{array}{c} \swarrow O_{l,t} \\ \searrow P_{l,s} \end{array} & \begin{array}{c} \swarrow O_{r,t} \\ \searrow P_{r,s} \end{array} \\
 & \langle \ominus, \ominus, \oplus, \oplus \rangle & \langle \oplus, \oplus, \oplus, \ominus \rangle \\
 \begin{array}{c} \swarrow P_{l,t} \\ \searrow O_{l,t} \end{array} & & \begin{array}{c} \swarrow P_{r,t} \\ \searrow O_{r,t} \end{array} \\
 \langle \ominus, \ominus, \ominus, \ominus \rangle & & \langle \oplus, \oplus, \oplus, \oplus \rangle
 \end{array}$$

The span of acute \mathbb{S} -functors

$$\mathfrak{A}_{\text{alt}} \xleftarrow{F_s^{\circ}} \mathfrak{A}_{\text{alt}}^{\circ} \xrightarrow{F_t^{\circ}} \mathfrak{A}_{\text{alt}}^{\text{op}} \times \mathfrak{A}_{\text{alt}}$$

is defined in the same way as was done for the tensor product in §4, except that the polarity table of \otimes is replaced by the polarity table of \multimap . Finally, one checks that the pullback (38) coincides with the pullback (37) or (39) computed in (40). This establishes that the four positions of the graph (41) describe the four polarities of a position (x_1, x_2, x_3) possibly reached by a strategy σ of the form

$$\sigma : A_1 \longrightarrow A_2 \multimap A_3 \tag{42}$$

This concludes the proof that the internal \mathbb{S} -category $\mathfrak{A}_{\text{alt}}$ is symmetric span-monoidal closed in the sense of Def. 5.5. From this elementary combinatorial property, it follows by Thm 5.6 that

THEOREM 6.1. *The bicategory $\mathbf{Games} = \mathbf{Games}(\mathfrak{A}_{\text{alt}})$ of sequential alternating games, strategies and simulations is symmetric monoidal closed.*

It should be noted that the tensorial implication of two games (A, λ_A) and (B, λ_B) is defined using the same recipe as the tensor product in (33) as

$$(A, \lambda_A) \multimap (B, \lambda_B) := (A \multimap B, F_t^{\circ}[0] \circ (\lambda_A \multimap \lambda_B))$$

where the category $A \multimap B$ and the functor $\lambda_A \multimap \lambda_B$ are defined by the pullback diagram (a) below, living in $\mathbb{S} = \mathbf{Cat}$:

$$\begin{array}{ccc}
 A \times B & \xleftarrow{\text{inclusion}} & A \multimap B \\
 \lambda_A \times \lambda_B \downarrow & & \downarrow \lambda \\
 \mathfrak{J}_{\text{game}}^{\text{op}} \times \mathfrak{J}_{\text{game}} & \xleftarrow{F_S^{-\circ}[0]=(1)(2)} & \mathfrak{J}_{\text{alt}}^{-\circ}[0] \xrightarrow{F_t^{-\circ}[0]} \mathfrak{J}_{\text{game}}
 \end{array}
 \quad (a) \quad (43)$$

7 CONCURRENT NON-ALTERNATING GAMES AND STRATEGIES

Here, we illustrate again, this time with concurrent non-alternating games and strategies, the fact that our main result (Thm. 5.6) is extremely simple and useful in practice, since it enables us to derive sophisticated bicategories of games, strategies and simulations from basic combinatorial properties of their scheduling policy. Here, we illustrate the usage and benefit of the theorem by deriving a bicategory $\mathbf{Games}(\mathfrak{J}_{\text{conc}})$ of concurrent non-alternating games and strategies from the construction of a specific internal \mathbb{S} -category $\mathfrak{J}_{\text{conc}}$ in the category $\mathbb{S} = \mathbf{Cat}$ of small categories. We start by defining the internal category $\mathfrak{J}_{\text{conc}}$. Its first component is the space of objects and *template of concurrent non-alternating games* defined as the category

$$\mathfrak{J}_{\text{conc}}[0]$$

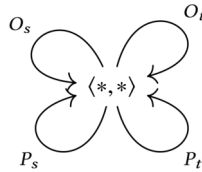
with one object $\langle * \rangle$ generated by the graph



and the additional equation $O \cdot P = P \cdot O$. Its second component is the space of maps and *template of concurrent non-alternating strategies* defined as the category

$$\mathfrak{J}_{\text{conc}}[1]$$

with one object $\langle *, * \rangle$ generated by the graph



and the six elementary equations

$$\begin{array}{lll}
 O_s \cdot P_s = P_s \cdot O_s & O_s \cdot P_t = P_t \cdot O_s & O_s \cdot O_t = O_t \cdot O_s \\
 O_t \cdot P_s = P_s \cdot O_t & O_t \cdot P_t = P_t \cdot O_t & P_s \cdot P_t = P_t \cdot O_s
 \end{array}$$

The categories $\mathfrak{J}_{\text{conc}}[0]$ and $\mathfrak{J}_{\text{conc}}[1]$ are constructed along the same lines as the categories $\mathfrak{J}_{\text{alt}}[0]$ and $\mathfrak{J}_{\text{alt}}[1]$ except that all the polarity positions appearing in the alternating case of $\mathfrak{J}_{\text{alt}}$ have been “depolarized” and identified to the very same object, noted $*$ or $\langle *, * \rangle$. Since a category with one object is the same thing as a monoid, the two categories $\mathfrak{J}_{\text{conc}}[0]$ and $\mathfrak{J}_{\text{conc}}[1]$ may be seen as free commutative monoids generated by the two elements O and P in the case of $\mathfrak{J}_{\text{conc}}[0]$ and by the four elements O_s, O_t, P_s and P_t in the case of $\mathfrak{J}_{\text{conc}}[1]$. For that reason, we find convenient to describe each of the two categories by the set of elements, or edges, which generate it:

$$\mathfrak{J}_{\text{conc}}[0] = \{ \ominus, \oplus \} \quad \mathfrak{J}_{\text{conc}}[1] = \{ \ominus, \oplus, \ominus, \oplus \}$$

The edges O and P generating $\mathfrak{A}_{\text{conc}}[0]$ are represented as red \ominus (for O) and blue \oplus (for P) while the edges O_s, O_t, P_s, P_t generating $\mathfrak{A}_{\text{conc}}[1]$ are represented as red \ominus on the left (for O_s), red \ominus on the right (for O_t), blue \oplus on the left (for P_s) and blue \oplus on the right (for P_t). The functors $s, t : \mathfrak{A}_{\text{conc}}[1] \rightarrow \mathfrak{A}_{\text{conc}}[0]$ are defined as expected, and in essentially the same way as in the case of alternating polarities (4) expect for the absence of polarities on positions:

$$s : \begin{array}{l} O_s \mapsto P \quad P_s \mapsto O \\ O_t, P_t \mapsto \text{id} \end{array} \qquad t : \begin{array}{l} O_t \mapsto O \quad O_t \mapsto P \\ O_s, P_s \mapsto \text{id} \end{array}$$

The diagrammatic representation is particularly convenient to describe the source and target functors s and t :



At this stage, it appears that the category $\mathfrak{A}_{\text{conc}}[2]$ of concurrent and non-alternating interaction defined by the pullback diagram (17) is once again a free commutative monoid generated this time by six elements depicted as follows:

$$\mathfrak{A}_{\text{conc}}[2] = \begin{array}{c} \text{---} \ominus \text{---} \oplus \text{---} \\ \text{---} \ominus \text{---} \oplus \text{---} \\ \text{---} \ominus \text{---} \oplus \text{---} \end{array}$$

More generally, all the categories involved to establish that $\mathfrak{A}_{\text{conc}}$ is span-monoidal closed in $\mathbb{S} = \mathbf{Cat}$ are defined as free commutative monoids of that form. The coherence diagrams is thus even easier to check than in the case of the template $\mathfrak{A}_{\text{alt}}$ of sequential alternating games. Typically, the important categories of concurrent and non-alternating polarities involved in the proof are generated by the following elements, or edges:

$$\mathfrak{A}_{\text{conc}}^{\otimes}[0] = \begin{array}{c} \text{---} \ominus \text{---} \oplus \text{---} \\ \text{---} \ominus \text{---} \oplus \text{---} \end{array} \quad \mathfrak{A}_{\text{conc}}^{\otimes}[1] = \begin{array}{c} \text{---} \ominus \text{---} \oplus \text{---} \\ \text{---} \ominus \text{---} \oplus \text{---} \\ \text{---} \ominus \text{---} \oplus \text{---} \\ \text{---} \ominus \text{---} \oplus \text{---} \end{array} \quad \mathfrak{A}_{\text{conc}}^{1 \otimes 2, 3}[0] = \begin{array}{c} \text{---} \ominus \text{---} \oplus \text{---} \\ \text{---} \ominus \text{---} \oplus \text{---} \\ \text{---} \ominus \text{---} \oplus \text{---} \end{array}$$

So, we may apply Theorem 5.6 and deduce from this purely combinatorial description of the internal category $\mathfrak{A}_{\text{conc}}$ that

THEOREM 7.1. *The bicategory $\mathbf{Games}(\mathfrak{A}_{\text{conc}})$ of concurrent and non-alternating games, strategies and simulations is symmetric monoidal closed.*

It should be stressed that this last theorem (and its companion Thm. 6.1) would be tremendously difficult to establish in full rigor without the conceptual tools introduced in the paper.

8 RELATIONAL SEMANTICS AND FUNCTORIAL SPANS

The relational semantics of linear logic was originally formulated by Girard as a mild variation of the coherence space model [Girard 1987]. The model relies on the existence of the $*$ -autonomous (and in fact compact closed) category \mathbf{Rel} of sets and relations. The model was more recently “categorified” into a 2-dimensional model of linear logic [Fiore et al. 2008] based this time on the bicategory \mathbf{Dist} of small categories and distributors (also called bimodules or profunctors) between them. Here, we consider another variation and categorification of the relational semantics of linear logic, based this time on the bicategory $\mathbf{Span}(\mathbf{Cat})$ of small categories and functorial spans between them. Interestingly, this bicategorical model of linear logic is an instance (although degenerate) of our general construction since $\mathbf{Span}(\mathbf{Cat}) = \mathbf{Games}(\mathfrak{A}_{\text{span}})$ for the *terminal* internal category $\mathfrak{A}_{\text{span}}$ in $\mathbb{S} = \mathbf{Cat}$, whose components $\mathfrak{A}_{\text{span}}[0]$ and $\mathfrak{A}_{\text{span}}[1]$ are thus equal to the terminal category $\mathbb{1}$. The internal category $\mathfrak{A}_{\text{span}}$ is span-monoidal closed for obvious reasons, and we may thus conclude from our main result (Thm. 5.6) that

THEOREM 8.1. *The bicategory $\mathbf{Span}(\mathbf{Cat}) = \mathbf{Games}(\mathbb{1}_{\text{conc}})$ of small categories, functorial spans and simulations is symmetric monoidal closed.*

9 STAR-AUTONOMOUS SPAN-MONOIDAL \mathbb{S} -CATEGORIES

In some situations of interest, we will see that the closed structure (Def. 5.5) of a symmetric span-monoidal \mathbb{S} -category $\mathbb{1}$ comes from a more primitive structure which we describe now.

Definition 9.1. A symmetric span-monoidal \mathbb{S} -category is **-autonomous* when there exists a pair of inverse \mathbb{S} -functors between \mathbb{S} -categories

$$*(-) : \mathbb{1}^{op} \longrightarrow \mathbb{1} \quad (-)^* : \mathbb{1} \longrightarrow \mathbb{1}^{op} \quad (44)$$

such that the acute \mathbb{S} -functor obtained by composition

$$\mathbb{1}^{op} \boxtimes \mathbb{1} \xrightarrow{\mathbb{1}^{op} \boxtimes (-)^*} \mathbb{1}^{op} \boxtimes \mathbb{1}^{op} \xrightarrow{\otimes_{\mathbb{1}}^{op}} \mathbb{1}^{op} \xrightarrow{*(-)} \mathbb{1} \quad (45)$$

makes the two spans (37) and (38) coincide up to isomorphism, and up to appropriate renaming of the components.

Every *-autonomous span-monoidal category is closed by definition, with tensorial implication $\multimap_{\mathbb{1}}$ defined as the acute span (45). As a matter of fact, we establish a variant of Thm. 5.6:

THEOREM 9.2. *Every *-autonomous span-monoidal \mathbb{S} -category $\mathbb{1}$ induces a *-autonomous bicategory $\mathbf{Games}(\mathbb{1})$.*

In particular, the *-autonomous bicategory $\mathbf{Games}(\mathbb{1})$ comes equipped with an isomorphism of bicategory induced by the pair of inverse \mathbb{S} -functors (44).

$$*(-) : \mathbf{Games}(\mathbb{1})^{op(1)} \longrightarrow \mathbf{Games}(\mathbb{1})$$

As mentioned in the introduction, the three \mathbb{S} -categories $\mathbb{1}_{\text{alt}}$, $\mathbb{1}_{\text{conc}}$ and $\mathbb{1}_{\text{span}}$ are *-autonomous. The inverse \mathbb{S} -functors (44) are defined by reversing the polarities of positions and moves in the case of $\mathbb{1}_{\text{alt}}$, by reversing the polarities of moves in the case of $\mathbb{1}_{\text{conc}}$ and by doing nothing in the case of $\mathbb{1}_{\text{span}}$. From this, we conclude by Thm. 9.2 that the bicategories $\mathbf{Games}(\mathbb{1}_{\text{alt}})$, $\mathbf{Games}(\mathbb{1}_{\text{conc}})$ and $\mathbf{Games}(\mathbb{1}_{\text{span}})$ are not just symmetric monoidal closed, but also *-autonomous.

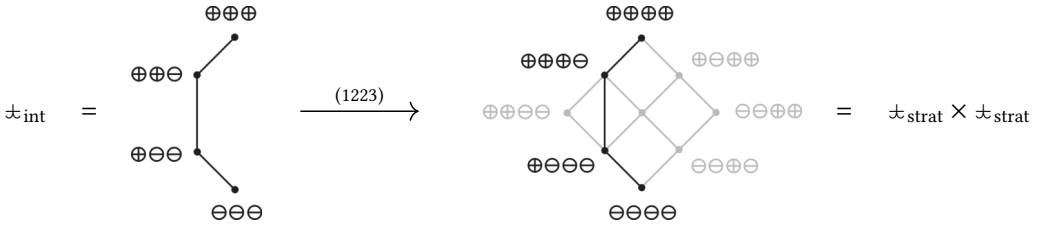
10 CONCLUSION AND FUTURE WORKS

Given a category \mathbb{S} with finite limits, we have shown how to derive a monoidal closed bicategory $\mathbf{Games}(\mathbb{1})$ of games, strategies and simulations from any span-monoidal closed category $\mathbb{1}$ internal in the category \mathbb{S} . We have also illustrated the construction with three monoidal closed bicategories $\mathbf{Games}(\mathbb{1}_{\text{alt}})$, $\mathbf{Games}(\mathbb{1}_{\text{conc}})$ and $\mathbf{Games}(\mathbb{1}_{\text{span}})$ corresponding to three different flavors of game semantics (alternating or non-alternating) and of relational semantics (functorial spans). Although the categorical material required to establish our main theorem (Thm. 5.6) is conceptually demanding, the construction of the span-monoidal closed category $\mathbb{1}$ is quite elementary in each case, and the verification of the expected coherence axioms is instructive and pleasant. The framework of template games thus provides a new foundation for game semantics, based on a very flexible and compelling way to construct new bicategories of games and strategies, with more sophisticated and interesting synchronization mechanisms than we ever had access to in the past. In particular, we are currently working on a formulation of the asynchronous game model of concurrent separation logic [Melliès and Stefanescu 2018] in the language of synchronization templates developed in the present paper. A fascinating aspect of the exercise is that the synchronization templates $\mathbb{1}$ are provided in that case by (variants of) the stateful or stateless machine models $\mathbb{1}_L$ and $\mathbb{1}_S$ in the

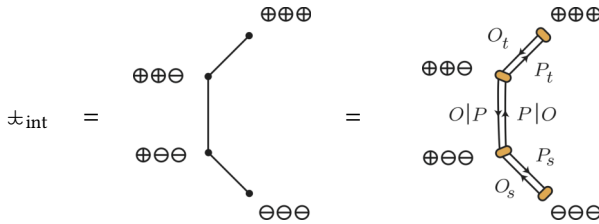
category \mathbb{S} of asynchronous graphs — thus quite far in apparence from the internal categories of game polarities $\mathfrak{A}_{\text{alt}}$ and $\mathfrak{A}_{\text{conc}}$ considered in the present paper. For all these reasons, we believe that this work provides a unifying and long-awaited framework for a new generation of integrated game semantics, at ease both with high-level and low-level description of programming languages.

A COMPUTING THE PULLBACK DEFINING THE CATEGORY OF INTERACTIONS

We give a detailed proof that the category $\mathfrak{A}_{\text{int}}$ defined by the pullback diagram (9) is freely generated by the graph (10). The computation relies on the existence of a faithful functor noted (1223) from the category $\mathfrak{A}_{\text{int}}$ computed by the pullback diagram (9) to the category $\mathfrak{A}_{\text{strat}} \times \mathfrak{A}_{\text{strat}}$. We find convenient to depict the functor (1223) in the following way:



In order to recover the oriented graph (10) from this representation, one needs to “fatten” every edge of the non-oriented graph by a ribbon whose two borders define a pair of oriented edges, going in opposite directions. The two edges O_t and P_t are then defined as the borders of the upper ribbon, the two edges $O|P$ and $P|O$ as the borders of the middle ribbon, and the two edges O_s and P_s as the borders of the lower ribbon. One obtains the following translation:



Note the functor $\pi_1 : \mathfrak{A}_{\text{int}} \rightarrow \mathfrak{A}_{\text{strat}}$ is characterized by the fact that it behaves in the following way on objects: $\langle \epsilon_1, \epsilon_2, \epsilon_3 \rangle \mapsto \langle \epsilon_1, \epsilon_2 \rangle$ and that it transports every edge of the graph (10) to an edge of the graph (3) or to the identity. For that reason, we find convenient to use the notation (12) for the first projection π_1 . Similarly, the functor $\pi_2 : \mathfrak{A}_{\text{int}} \rightarrow \mathfrak{A}_{\text{strat}}$ is characterized by the fact that it behaves in the following way on objects $\langle \epsilon_1, \epsilon_2, \epsilon_3 \rangle \mapsto \langle \epsilon_2, \epsilon_3 \rangle$ and that it transports every edge of the graph (10) to an edge of the graph (3) or to the identity. We thus proceed accordingly, and use the notation (23) for the second projection $\pi_2 : \mathfrak{A}_{\text{int}} \rightarrow \mathfrak{A}_{\text{strat}}$. Note that putting the functors (12) and (23) together and side by side using the universality property of the cartesian product, one recovers the functor (1223) : $\mathfrak{A}_{\text{int}} \rightarrow \mathfrak{A}_{\text{strat}} \times \mathfrak{A}_{\text{strat}}$. The notation is thus a useful guide in the categorical combinatorics of synchronization we are exploring.

ACKNOWLEDGMENTS

The research underlying this article was partially supported by the ERC Advanced Grant DualL, number 670624. The author would like to thank Thomas Ehrhard, Tom Hirschowitz, Léo Stefanescu, Thomas Streicher and Noam Zeilberger for enlightening discussions on this work, as well as the POPL reviewers for their useful feedback on the final version of the paper.

REFERENCES

- Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. 2000. Full Abstraction for PCF. *Information and Computation* 163, 2 (2000), 409–470.
- Jean Bénabou. 1967. *Introduction to bicategories*. Number 40 in Lecture Notes in Mathematics. Springer Verlag, 1–77.
- Jean Bénabou. 1985. Fibered Categories and the Foundations of Naive Category Theory. *Journal of Symbolic Logic* 50, 1 (1985), 10–37.
- Simon Castellan and Nobuko Yoshida. 2019. The two sides of the same coin: Session Types and Game Semantics. In *Accepted for publication at POPL'19*.
- Marcelo Fiore, Nicola Gambino, Martin Hyland, and Glynn Winskel. 2008. The cartesian closed bicategory of generalised species of structures. *Journal of the London Mathematical Society* 77 (2008), 203–220.
- Richard Garner. 2006. *Polycategories*. Ph.D. Dissertation. University of Cambridge.
- Dan R. Ghica and Mohamed N. Mena. 2011. Synchronous Game Semantics via Round Abstraction. In *Foundations of Software Science and Computational Structures*, Martin Hofmann (Ed.). Springer Verlag, 350–364.
- Jean-Yves Girard. 1987. Linear Logic. *Theoretical Computer Science* 50 (1987), 1–101.
- Russell Harmer. 1999. *Games and Full Abstraction for Nondeterministic Languages*. Ph.D. Dissertation. Imperial College of Science, London.
- Russell Harmer, Martin Hyland, and Paul-André Mellies. 2007. Categorical Combinatorics for Innocent Strategies. In *Proceedings of the 22nd IEEE Symposium on Logic in Computer Science (LICS)*. IEEE Computer Society, 379–388.
- André Hirschowitz, Michel Hirschowitz, and Tom Hirschowitz. 2007. A Theory for Game Theories. In *Proceedings of the 27th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS) (LNCS)*, Vol. 4855. Springer Verlag, 192–203.
- Martin Hyland and Luke Ong. 2000. On Full Abstraction for PCF. *Information and Computation* 163, 2 (2000), 285–408.
- Martin Hyland and Andrea Schalk. 1999. Abstract Games for Linear Logic. In *Proceedings of Category Theory in Computer Science (CTCS) (Electronic Notes in Theoretical Computer Science)*, Vol. 29.
- Martin Hyland and Andrea Schalk. 2003. Glueing and Orthogonality for Models of Linear Logic. *Theoretical Computer Science* 294, 1-2 (2003), 183–231.
- Paul-André Mellies. 2006. Asynchronous games 2: the true concurrency of innocence. *Theoretical Computer Science* 358, 2-3 (2006), 200–228.
- Paul-André Mellies and Samuel Mimram. 2007. Asynchronous Games: Innocence Without Alternation. In *18th International Conference on Concurrency Theory (CONCUR 2007)*. 395–411.
- Paul-André Mellies and Samuel Mimram. 2008. From Asynchronous Games to Concurrent Games. (2008). Unpublished manuscript. Available on the author's webpage.
- Paul-André Mellies and Léo Stefanescu. 2018. An Asynchronous Soundness Theorem for Concurrent Separation Logic. In *Proceedings of LICS 2018*.
- Sylvain Rideau and Glynn Winskel. 2011. Asynchronous Strategies. In *Proceedings of the 26th Annual Symposium on Logic in Computer Science (LICS)*. IEEE, 409–418.
- Peter Selinger. 1999. Categorical Structure of Asynchrony. In *Proceedings of Mathematical Foundations of Programming Semantics (MFPS 15) (Electronic Notes in Theoretical Computer Science)*, Vol. 20. Elsevier, 158–181.
- Glynn Winskel. 1986. Event Structures. In *Petri Nets: Applications and Relationships to Other Models of Concurrency (Lecture Notes in Computer Science)*, Vol. 255. Springer Verlag, 325–392.