From Asynchronous Games to Concurrent Games

Paul-André Melliès Samuel Mimram*

September 30, 2008

Abstract

Game semantics was introduced in order to capture the dynamic behaviour of proofs and programs. In these semantics, the interaction between a program and its environment is modeled by a series of moves exchanged between two players in a game. Every program thus induces a strategy describing how it reacts when it is provided information by its environment. Traditionally, strategies considered in game semantics are alternating: the two protagonists play a move one after the other. This property is very natural when modeling sequential programming languages, but is not desirable for programs with concurrent features, since interactions cannot be synchronized globally anymore. Extending fundamental notions of game semantics to a non-alternating setting is far from being straightforward and requires to deeply rethink the definition of strategies. Recently, a series of interactive models, such as concurrent games where strategies are closure operators, were introduced in order to give denotational semantics of programming languages or logics with concurrent features. However, these models were poorly connected with traditional game semantics. We show here that asynchronous games, which combine true concurrency and game semantics, can be used to provide a precise link between these two kind of interactive semantics, thus laying foundations for game semantics of concurrent systems.

1 Introduction

The alternating origins of game semantics. Game semantics was invented (or reinvented) at the beginning of the 1990s, in the turmoil produced by the discovery of linear logic, in order to describe the dynamics of proofs and programs. Game semantics proceeds according to the principles of trace semantics in concurrency theory: every program and proof is interpreted by the sequences of interactions, called *plays*, that it can have with its environment. The novelty of game semantics is that this set of plays defines a *strategy* which reflects the interactive behaviour of the program inside the *game* specified by the type of the program.

Game semantics was originally influenced by a pioneering work by Joyal [18] building a category of games (called Conway games) and alternating strategies. In this setting, a game is defined as a decision tree (or more precisely, a dag) in which every edge, called *move*, has a polarity indicating whether it is played by the program, called Proponent, or by the environment, called Opponent. A play is alternating when Proponent and Opponent alternate strictly – that is, when neither of them plays two moves in a row. A strategy is alternating when it contains only alternating plays.

*This work has been supported by the ANR Invariants algébriques des systèmes informatiques (INVAL). Physical address: Équipe PPS, CNRS and Université Paris 7, 2 place Jussieu, case 7017, 75251 Paris cedex 05, France. Email addresses: mellies@pps.jussieu.fr and smimram@pps.jussieu.fr.

The category of alternating strategies introduced by Joyal was later refined by Abramsky and Jagadeesan [1] and Hyland and Ong [14] in order to characterize the dynamic behaviour of proofs in (multiplicative) linear logic. The key idea is that the tensor product of linear logic, noted $\otimes$, may be distinguished from its dual, noted $\parr$, by enforcing a *switching policy* on plays – ensuring for instance that a strategy of $A \otimes B$ reacts to an Opponent move played in the subgame A by playing a Proponent move in the same subgame A.

The notion of *pointer game* was then introduced by Hyland and Ong [15], and independently by Nickau [27], in order to characterize the dynamic behaviour of programs in the programming language PCF – a simply-typed λ-calculus extended with recursion, conditional branching and arithmetical constants. The programs of PCF are characterized dynamically as particular kinds of strategies with partial memory – called *innocent* because they react to Opponent moves according to their own *view* of the play. This view is itself a play, extracted from the current play by removing all its "invisible" or "inessential" moves. This extraction is performed by induction on the length of the play, using the pointer structure of the play, and the hypothesis that Proponent and Opponent alternate strictly.

This seminal work on pointer games led to the first generation of game semantics for programming languages – based on ideas developed in the semantics of proofs. However, because Proponent and Opponent strictly alternate in the original definition of pointer games, these game semantics focus on sequential languages like Algol or ML, rather than on concurrent languages.

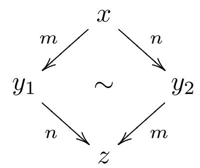
Concurrent games. This convinced a little community of researchers to work on the foundations of non-alternating games – where Proponent and Opponent are thus allowed to play several moves in a row at any point of the interaction. Abramsky and Melliès [2] introduced a new kind of game semantics to that purpose, based on *concurrent games* defined as partial orders (or more precisely, complete lattices) of *positions*, on which strategies defined as *closure operators* interact concurrently. In this approach, the image $\sigma(x)$ of a position x by the closure operator σ describes the position reached by the strategy σ after it has played all the moves it could play. Recall that a closure operator σ on a partial order D is a function $\sigma : D \longrightarrow D$ satisfying the following properties:

(1)	σ is increasing:	$\forall x \in D,$	$x \leq \sigma(x),$
(2)	σ is idempotent:	$\forall x \in D,$	$\sigma(x) = \sigma(\sigma(x)),$
(3)	σ is monotone:	$\forall x, y \in D,$	$x \leq y \Rightarrow \sigma(x) \leq \sigma(y).$

The order on positions $x \leq y$ reflects the intuition that the position y contains more information than the position x. Typically, one should think of a position x as a set of moves in a game, and $x \leq y$ as set inclusion $x \subseteq y$. Now, Property (1) expresses that a strategy σ which transports the position x to the position $\sigma(x)$ increases the amount of information. Property (2) reflects the intuition that the strategy σ delivers all its information when it transports the position x to the position $\sigma(x)$, and thus transports the position $\sigma(x)$ to itself. Property (3) is both fundamental and intuitively right, but also more subtle to justify. Note that the interaction induced by such a strategy σ is possibly non-alternating, since the strategy transports the position x to the position $\sigma(x)$ by "playing in one go" all the moves appearing in $\sigma(x)$ but not in x.

It is well-known that a closure operator σ is characterized by its set of fixpoints, that is, the positions x satisfying $x = \sigma(x)$. Hence, a strategy can be expressed either as a set of positions (the set of fixpoints of the closure operator) in concurrent games or as a set of alternating plays in pointer games.

Asynchronous transition systems. In order to understand how the two formulations of strategies are related, one should start from an obvious analogy with concurrency theory: pointer games define an *interleaving semantics* (based on sequences of transitions) whereas concurrent games define a *truly concurrent semantics* (based on sets of positions, or states) of proofs and programs. Now, Mazurkiewicz [20] taught us this important lesson: a truly concurrent semantics may be regarded as an interleaving semantics (typically a transition system) equipped with *asynchronous tiles* – represented diagrammatically as 2-dimensional tiles

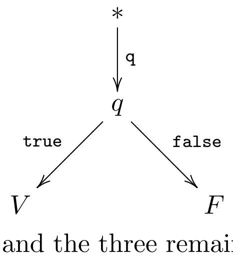


(1)

expressing that the two transitions m and n from the state x are *independent*, and consequently, that their scheduling does not matter from a truly concurrent point of view. This additional structure induces an equivalence relation on transition paths, called *homotopy*, defined as the smallest congruence relation $\sim$ identifying the two schedulings $m \cdot n$ and $n \cdot m$ for every tile of the form (1). The word *homotopy* should be understood mathematically as (directed) homotopy in the topological presentation of asynchronous transition systems as *n-cubical sets* [13]. This 2-dimensional refinement of usual 1-dimensional transition systems enables us to express simultaneously the interleaving semantics of a program as the set of transition paths it generates, and its truly concurrent semantics, as the homotopy classes of these transition paths. When the underlying 2-dimensional transition system is contractible in a suitable sense, explained later, these homotopy classes coincide in fact with the positions of the transition system.

Asynchronous games. Guided by these intuitions, Melliès introduced the notion of *asynchronous game*, which unifies in a surprisingly conceptual way the two heterogeneous notions of pointer game and concurrent game. Asynchronous games are played on asynchronous (2-dimensional) transition systems, where every transition (or move) is equipped with a *polarity*, expressing whether it is played by Proponent or by Opponent. A *play* is defined as a path starting from the root (noted $*$) of the game, and a *strategy* is defined as a well-behaved set of alternating plays, in accordance with the familiar principles of pointer games. Now, the difficulty is to understand how (and when) a strategy defined as a set of plays may be reformulated as a set of positions, in the spirit of concurrent games.

The first step in the inquiry is to observe that the asynchronous tiles (1) offer an alternative way to describe *justification pointers* between moves. For illustration, consider the boolean game $\mathbb{B}$, where Opponent starts by asking a question `q`, and Proponent answers by playing either `true` or `false`. The game is represented by the decision tree

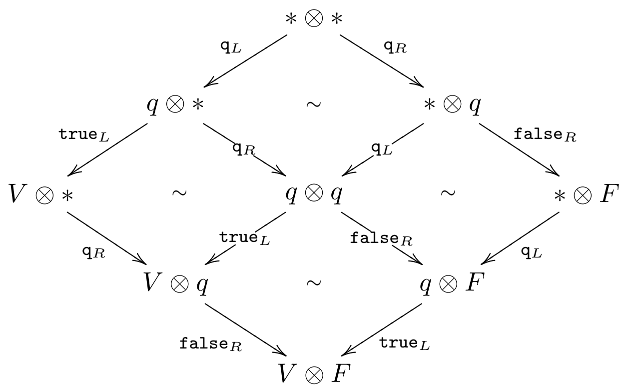


(2)

where $*$ is the root of the game, and the three remaining positions are called q, V (V for "Vrai" in French) and F. At this point, since there is no concurrency involved, the game may be seen either as an asynchronous game, or as a pointer game. Now, the game $\mathbb{B} \otimes \mathbb{B}$ is constructed by taking two boolean games "in parallel". It simulates a very simple computation device, containing two boolean memory cells. In a typical interaction, Opponent starts by asking with $\mathtt{q}_L$ the value of the left memory cell, and Proponent answers $\mathtt{true}_L$; then, Opponent asks with $\mathtt{q}_R$ the value of the right memory cell, and Proponent answers $\mathtt{false}_R$. The play is represented as follows in pointer games:

$$\mathtt{q}_L \cdot \mathtt{true}_L \cdot \mathtt{q}_R \cdot \mathtt{false}_R$$

The play contains two justification pointers, each one represented by an arrow starting from a move and leading to a previous move. Typically, the justification pointer from the move $\mathtt{true}_L$ to the move $\mathtt{q}_L$ indicates that the answer $\mathtt{true}_L$ is necessarily played after the question $\mathtt{q}_L$. The same situation is described using 2-dimensional tiles in the asynchronous game $\mathbb{B} \otimes \mathbb{B}$ below:



(3)

The justification pointer between the answer $\mathtt{true}_L$ and its question $\mathtt{q}_L$ is replaced here by a *dependency* relation between the two moves, ensuring that the move $\mathtt{true}_L$ cannot be permuted before the move $\mathtt{q}_L$. The dependency itself is expressed by a "topological" obstruction: the lack of a 2-dimensional tile permuting the transition $\mathtt{true}_L$ before the transition $\mathtt{q}_L$ in the asynchronous game $\mathbb{B} \otimes \mathbb{B}$.

This basic correspondence between justification pointers and asynchronous tiles allows a reformulation of the original definition of *innocent strategy* in pointer games (based on views) in the language of asynchronous games. Surprisingly, the reformulation leads to a purely local and diagrammatic definition of innocence in asynchronous games, which does not mention the notion of view any more. For the positions in the graph to truly be the basic notion, a strategy has to be well-behaved with respect to different but equivalent ways of getting to the same position – otherwise, instead of being able to talk about strategies as subgraphs one would have to stick to the notion of being closed under equivalent plays. This diagrammatic reformulation leads precisely to the important discovery that innocent strategies are *positional* in the following sense. Suppose that two alternating plays $s, t : * \longrightarrow x$ with the same target position x are elements of an innocent strategy σ, and that m is an Opponent move from position x. Suppose moreover that the two plays s and t are equivalent modulo homotopy. Then, the innocent strategy σ extends the play $s \cdot m$ with a Proponent move n if and only if it extends the play $t \cdot m$ with the same Proponent move n. Formally:

$$s \cdot m \cdot n \in \sigma \quad \text{and} \quad s \sim t \quad \text{and} \quad t \in \sigma \quad \text{implies} \quad t \cdot m \cdot n \in \sigma. \tag{4}$$

From this follows that every innocent strategy σ is characterized by the set of *positions* (understood here as homotopy classes of plays) reached in the asynchronous game.

This set of positions can be considered as a relation on positions. It moreover defines a closure operator, and thus a strategy in the sense of concurrent games. Asynchronous games offer in this way an elegant and unifying point of view on pointer games, concurrent games and relational models.

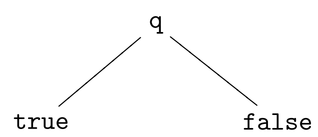
Concurrency in game semantics. There is little doubt that a new generation of game semantics is currently emerging along this foundational work on concurrent games. We see at least three converging lines of research. First, authors trained in proof theory and game semantics – Curien and Faggian – relaxed the sequentiality constraints required by Girard on *designs* in ludics, leading to the notion of L-net [7] which lives at the junction of syntax (expressed as proof nets) and game semantics (played on event structures). Then, authors trained in game semantics – Ghica, Laird and Murawski – were able to characterize the interactive behaviour of various concurrent programming languages like Parallel Algol [10] or an asynchronous variant of the π-calculus [19] using directly (and possibly too directly) the language of pointer games. Finally, and more recently, authors trained in process calculi, true concurrency and game semantics – Varacca and Yoshida – were able to extend Winskel's truly concurrent semantics of CCS, based on event structures, to a significant fragment of the π-calculus, uncovering along the way a series of nice conceptual properties of *confusion-free* event structures [33].

So, a new generation of game semantics for concurrent programming languages is currently emerging... but their various computational models are still poorly connected. We would like a regulating theory here, playing the role of Hyland and Ong pointer games in traditional (that is, alternating) game semantics. Asynchronous games are certainly a good candidate, because they combine interleaving semantics and causal semantics in a harmonious way. Unfortunately, they were limited until now to alternating strategies [23]. The key contribution of this article is thus to extend the asynchronous framework to non-alternating strategies in a smooth way, inspired by the ideas of linear logic.

Asynchronous games without alternation. One particularly simple recipe to construct an asynchronous game is to start from a *partial order* of events where, in addition, every event has a polarity, indicating whether it is played by Proponent or Opponent. This partial order $(M, \preceq)$ is then equipped with a *compatibility relation* satisfying a series of suitable properties – defining what Winskel calls an *event structure*. A *position* x of the asynchronous game is defined as a set of *compatible* events (or moves) closed under the "causality" order:

$$\forall m, n \in M, \qquad m \preceq n \quad \text{and} \quad n \in x \quad \text{implies} \quad m \in x.$$

Typically, the boolean game $\mathbb{B}$ described in (2) is generated by the event structure

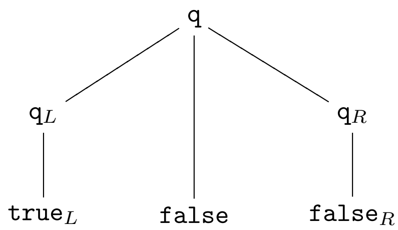


where $\mathtt{q}$ is an Opponent move, and $\mathtt{true}$ and $\mathtt{false}$ are two *incompatible* Proponent moves, with the positions q, V, F defined as $q = \{\mathtt{q}\}$, $V = \{\mathtt{q}, \mathtt{true}\}$ and $F = \{\mathtt{q}, \mathtt{false}\}$. The tensor product $\mathbb{B} \otimes \mathbb{B}$ of two boolean games is then generated by putting side by side the two event structures, in the expected way. The resulting asynchronous game looks like a flower with four petals, one of them described in (3).

More generally, every formula of linear logic defines an event structure – which generates in turn the asynchronous game associated to the formula. For instance, the event structure induced by the formula

$$(\mathbb{B} \otimes \mathbb{B}) \multimap \mathbb{B} \tag{5}$$

contains the following partial order of compatible events:



(6)

which may be seen alternatively as a (maximal) position in the asynchronous game associated to the formula.

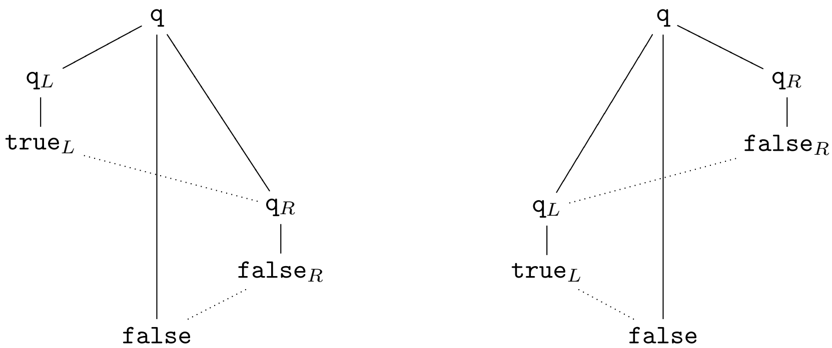
This game implements the interaction between a boolean function (the Proponent) of type (5) and its two arguments (the Opponent). In a typical play, Opponent starts by playing the move $\mathtt{q}$ asking the value of the boolean output; Proponent reacts by asking with $\mathtt{q}_L$ the value of the left input, and Opponent answers $\mathtt{true}_L$; then, Proponent asks with $\mathtt{q}_R$ the value of the right input, and Opponent answers $\mathtt{false}_R$; at this point only, using the knowledge of its two arguments, Proponent answers $\mathtt{false}$ to the initial question:

$$\mathtt{q} \cdot \mathtt{q}_L \cdot \mathtt{true}_L \cdot \mathtt{q}_R \cdot \mathtt{false}_R \cdot \mathtt{false} \tag{7}$$

Of course, Proponent could have explored its two arguments in the other order, from right to left, this inducing the play

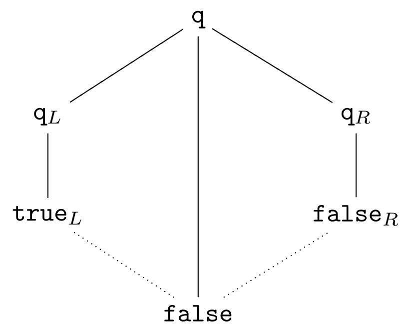
$$\mathtt{q} \cdot \mathtt{q}_R \cdot \mathtt{false}_R \cdot \mathtt{q}_L \cdot \mathtt{true}_L \cdot \mathtt{false} \tag{8}$$

The two plays start from the empty position $*$ and reach the same position of the asynchronous game. They may be seen as different linearizations (in the sense of order theory) of the partial order (6) provided by the game, that is total orders extending the partial order on the game. Each of these linearizations may be represented by adding causality (dotted) edges between moves to the original partial order (6), in the following way:



(9)

The play (7) is an element of the strategy representing the *left* implementation of the *strict conjunction*, whereas the play (8) is an element of the strategy representing its *right* implementation. Both of these strategies are alternating. Now, there is also a *parallel* implementation, where the conjunction asks the value of its two arguments at the same time. The associated strategy is not alternating anymore: it contains the play (7) and the play (8), and moreover, all the (possibly non-alternating) linearizations of the following partial order.



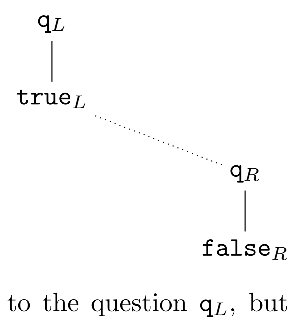
(10)

This illustrates an interesting phenomenon, of a purely concurrent nature: every play s of a concurrent strategy σ coexists with other plays t in the strategy, having the same target position x – and in fact, equivalent modulo homotopy. It is possible to reconstruct from this set of plays a *partial order* on the events of x, refining the partial order on events provided by the game. This partial order describes entirely the strategy σ under the position x: more precisely, the set of plays in σ reaching the position x coincides with the set of the linearizations of the partial order.

Our definition of innocent strategy will ensure the existence of such an underlying "causality order" for every position x reached by the strategy. Every innocent strategy will then define an event structure, obtained by putting together all the induced partial orders. The construction requires refined tools from the theory of asynchronous transition systems, and in particular the fundamental notion of *cube property*.

Composition of courteous strategies. We introduce in Section 6 the notion of *ingenuous* strategy, defined as a strategy regulated by an underlying "causality order" on moves for every reached position, and satisfying a series of suitable diagrammatic properties. This notion is strengthened in Section 7 into the notion of *courteous* strategy. We show in a precise way that these strategies induce closure operators on the positions of the game, and that these closure operators characterize the strategies. This property thus relates asynchronous games and concurrent games.

There is however a subtle mismatch between the compositions of strategies seen as sets of plays and the strategies seen as closure operators, preventing the relation between the two models from being functorial. This difficulty with composition is depicted in the following example. Consider for instance the ingenuous strategy σ of type $\mathbb{B} \otimes \mathbb{B}$ generated by the partial order:



(11)

The strategy answers $\mathtt{true}_L$ to the question $\mathtt{q}_L$, but accepts the question $\mathtt{q}_R$ only after it has given the answer $\mathtt{true}_L$. Composing the strategy σ with the *right* implementation of the strict conjunction pictured on the right-hand side of (9) induces a play $\mathtt{q}$ stopped by a *deadlock* at the position $\{\mathtt{q}\}$. On the other hand, composing the strategy with the *left* or the *parallel* implementation is fine, and leads to a complete interaction.

This dynamic phenomenon is better understood by introducing two new binary connectives $\olessthan$ and $\ogreaterthan$ called "before" and "after", describing sequential composition in asynchronous games. The game $A \olessthan B$ is defined as the 2-dimensional restriction of the game $A \otimes B$ to the plays s such that every move played before a move in A is also in A; or equivalently, every move played after a move B is also in B. The game $A \ogreaterthan B$ is simply defined as the game $B \olessthan A$, where the component B thus starts.

The ingenuous strategy σ in $\mathbb{B} \otimes \mathbb{B}$ specializes to a strategy in the subgame $\mathbb{B} \olessthan \mathbb{B}$, which reflects it, in the sense that every play $s \in \sigma$ is equivalent modulo homotopy to a play $t \in \sigma$ in the subgame $\mathbb{B} \olessthan \mathbb{B}$. This is not true anymore when one specializes the strategy σ to the subgame $\mathbb{B} \ogreaterthan \mathbb{B}$, because the play $\mathtt{q}_L \cdot \mathtt{true}_L \cdot \mathtt{q}_R \cdot \mathtt{false}_R$ is an element of σ which is not equivalent modulo homotopy to any play $t \in \sigma$ in the subgame $\mathbb{B} \ogreaterthan \mathbb{B}$. For that reason, we declare that the strategy σ is innocent in the game $\mathbb{B} \olessthan \mathbb{B}$ but *not* in the game $\mathbb{B} \otimes \mathbb{B}$.

Scheduled strategies. This leads to an interactive criterion which tests dynamically whether an ingenuous strategy σ is suitable for a given formula of linear logic. The criterion is based on *scheduling conditions*. The idea is to *switch* every tensor product $\otimes$ of the formula as $\olessthan$ or $\ogreaterthan$ and to test whether every play s in the strategy σ is equivalent modulo homotopy to a play $t \in \sigma$ in the induced subgame. Every such switching $\mathscr{S}$ reflects a choice of scheduling by the counter-strategy: a scheduled strategy is thus a strategy flexible enough to adapt to *every* scheduling of the tensor products by Opponent. Syntactically, this means that an ingenuous strategy is scheduled if and only if the underlying proof-structure satisfies a directed (and more liberal) variant of the acyclicity criterion introduced by Girard [11] and reformulated by Danos and Regnier [8].

An interesting feature of our work is that the description of scheduling is based on an orthogonality relation between strategies, in a fashion directly inspired by the ideas of linear logic. In particular, the hierarchy of games and strategies defined by tensor product and double orthogonality, is characterized interactively. In this characterization, the scheduling tests $\mathscr{S}$ define some kind of generating basis of a combinatorial nature.

We will establish that every ingenuous strategy is positional, and thus induces a set of *acceptance positions* where the strategy halts, and waits for the Opponent to select a next move. One important observation of this article, however, is that there is a subtle mismatch between the interaction of two ingenuous strategies seen as sets of plays, and seen as sets of positions (or equivalently as closure operators). Typically, the right implementation of the strict conjunction in (9) composed to the strategy σ in (11) induces two different fixpoints in the concurrent game model: the deadlock position $(* \otimes *) \multimap q = \{\mathtt{q}\}$ reached during the asynchronous interaction, and the complete position $(V \otimes F) \multimap F = \{\mathtt{q}, \mathtt{q}_L, \mathtt{true}_L, \mathtt{q}_R, \mathtt{false}_R, \mathtt{false}\}$ which is never reached interactively. The scheduling assumption is precisely what ensures that this will never occur: the fixpoint computed in the concurrent game model is unique, and coincides with the position eventually reached in the asynchronous game model. In particular, scheduled strategies compose properly in the sense that their composition correspond to concurrent strategies.

Plan of the paper. We did our best to give in this introduction an informal but detailed overview of this demanding work, which combines together ideas from several fields: game semantics, concurrency theory, linear logic, etc. We focus now on the conceptual properties of innocent strategies, expressed in the diagrammatic language of asynchronous transition systems. A more detailed presentation can be found in the second author's PhD thesis [26]. The cube property is recalled in Section 2. We define a category of 1-Player games in Section 3 and study its positional strategies in Section 4. We reformulate the definition of positionality directly on sets of traces in Section 5. The notion of ingenuous strategy is introduced for 2-Player games in Section 6. It is then refined into the notion of courteous strategy in Section 7, which enables us to statically relate asynchronous games and concurrent games. Finally, this correspondence is extended to a functorial one in Sections 9 and 10 using the notion of scheduled strategy, which is defined using a scheduling criterion capturing the essence of a directed variant of the acyclicity criterion of linear logic.

2 The cube property

The *cube property* expresses a fundamental causality principle in the diagrammatic language of asynchronous transition systems [4, 32, 34]. The property is related to stability in the sense of Berry [5]. It was first noticed by Nielsen, Plotkin and Winskel in [28], then reappeared in [29] and [12, 24] and was studied thoroughly by Kuske in his PhD thesis; see [9] for a survey.

Asynchronous graphs. The most natural way to express the property is to start from what we call an *asynchronous graph*. Recall that a *graph* $G = (V, E, \partial_0, \partial_1)$ consists of a set V of vertices (or *positions*), a set E of edges (or *transitions*), and two functions $\partial_0, \partial_1 : E \to V$ called respectively source and target functions. An *asynchronous graph* $G = (G, \diamond)$ is a graph G together with a relation $\diamond$ on coinitial and cofinal transition paths of length 2. Every relation $s \diamond t$ is represented diagrammatically as a 2-dimensional tile

$$\begin{array}{ccccc} & & x & & \\ & \overset{m}{\swarrow} & & \overset{n}{\searrow} & \\ y_1 & & \sim & & y_2 \\ & \underset{p}{\searrow} & & \underset{q}{\swarrow} & \\ & & z & & \end{array} \tag{12}$$

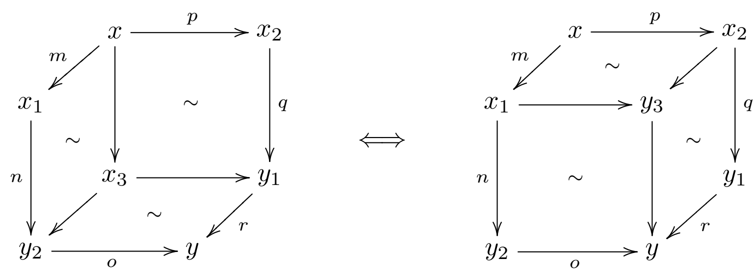
where $s = m \cdot p$ and $t = n \cdot q$. In this diagram, the transition q is intuitively the *residual* of the transition m after the transition n. One requires the two following properties for every asynchronous tile:

1. $m \neq n$ and $p \neq q$,
2. the pair of transitions (n, q) is uniquely determined by the pair of transitions (m, p), and conversely the pair of transitions (m, p) is uniquely determined by the pair of transitions (n, q).

The main difference with the asynchronous tile (1) occurring in the asynchronous transition systems defined in [34, 30] is that the transitions are not labelled by events: so, the 2-dimensional structure is purely "geometric" and not deduced from an independence relation on events. What matters is that the 2-dimensional structure enables one to define a homotopy relation $\sim$ on paths in exactly the same way.

The cube property. Every homotopy class of a path $s = m_1 \cdots m_k$ coincides with the set of linearizations of a partial order on its transitions if, and only if, the asynchronous graph satisfies the following *cube property*:

Definition 1 (cube property) *An asynchronous graph G satisfies the* cube property *when a hexagonal diagram in G induced by two coinitial and cofinal paths $m \cdot n \cdot o : x \longrightarrow\!\!\!\!\!\rightarrow y$ and $p \cdot q \cdot r : x \longrightarrow\!\!\!\!\!\rightarrow y$ is filled by 2-dimensional tiles as pictured in the left-hand side of the diagram below, if and only it is filled by 2-dimensional tiles as pictured in the right-hand side of the diagram:*



The cube property is for instance satisfied by every asynchronous transition system and every transition system with independence in the sense of [34, 30]. The correspondence between homotopy classes and sets of linearizations of a partial order adapts, in our setting, a standard result on pomsets and asynchronous transition systems with dynamic independence due to Bracho, Droste and Kuske [6].

Every asynchronous graph G equipped with a distinguished initial position (noted $*$) induces an asynchronous graph $[G]$ whose positions are the homotopy classes of paths starting from the position $*$, and whose edges $m : [s] \longrightarrow [t]$ between the homotopy classes of the paths $s : * \longrightarrow\!\!\!\!\!\rightarrow x$ and $t : * \longrightarrow\!\!\!\!\!\rightarrow y$ are the edges $m : x \longrightarrow y$ such that $s \cdot m \sim t$. When the original asynchronous graph G satisfies the cube property, the resulting asynchronous graph $[G]$ is "contractible" in the sense that every two coinitial and cofinal paths are equivalent modulo homotopy.

So, we will suppose from now on that all our asynchronous graphs satisfy the cube property and are therefore contractible. The resulting framework is very similar to the domain of configurations of an event structure. Indeed, every contractible asynchronous graph defines a partial order on its set of positions, defined by reachability: $x \leq y$ when $x \longrightarrow\!\!\!\!\!\rightarrow y$. Moreover, this order specializes to a finite distributive lattice under every position x, rephrasing – by Birkhoff representation theorem – the already mentioned property:

Proposition 1 *In an asynchronous graph G which satisfies the cube property, the homotopy class of a path $s : x \longrightarrow\!\!\!\!\!\rightarrow y$ is in one-to-one correspondence with the set of linearizations of a partial order $\leq_s$ on its transitions.*

Finally, every transition may be labelled by an "event" representing the transition modulo a "zig-zag" relation, identifying the moves m and q in every asynchronous tile (12). The idea of "zig-zag" is folklore: it appears for instance in [30] in order to translate a transition system with independence into a labelled event structure.

3 A monoidal category of 1-Player games

The categories of games and strategies considered in game semantics are usually defined on 2-Player games. However, it appears that the very construction of a category can be performed directly on 1-Player games.

Definition 2 (game) *A* 1-Player game $(G, *)$ *is defined as an asynchronous graph* G *together with a distinguished initial position* $*$*, such that for every position* x*, there exists a path* $s : * \longrightarrow\!\!\!\!\!\rightarrow x$*.*

Indeed, we will construct here a monoidal category $\mathcal{O}$ of 1-Player games and strategies, which will be self-dual – that is, compact closed – in the same way as the category of sets and relations.

Definition 3 (play) *A* play *is a path* $s : * \longrightarrow\!\!\!\!\!\rightarrow x$ *starting from the initial position* $*$ *of the game.*

Definition 4 (strategy) *A* strategy *is a set of plays of the game.*

From now on, we write $\sigma : A$ when σ is a strategy of the game A. Note that we do not require that a strategy is closed under prefix: this will play a fundamental rôle in the construction of the identity map in our category of 1-Player games.

Tensor product. Given two 1-Player games $A = (G_A, *_A)$ and $B = (G_B, *_B)$, their *tensor product* $A \otimes B = (G_{A\otimes B}, *_{A\otimes B})$ is defined as follows. Its underlying asynchronous graph is the graph whose positions are pairs (x_A, x_B) of positions of G_A and of G_B, sometimes also noted $x_A \otimes x_B$, whose transitions are of the form $m : x_A \otimes x_B \longrightarrow y_A \otimes x_B$ where $m : x_A \longrightarrow y_A$ is a transition of G_A, or of the form $m : x_A \otimes x_B \longrightarrow x_A \otimes y_B$ where $m : x_B \longrightarrow y_B$ is a transition in G_B. The tiling relation relates two paths $m \cdot n$ and $n \cdot m$ such that m and n are both transitions in G_A (resp. G_B) and $m \cdot n$ and $n \cdot m$ are related in G_A (resp. in G_B) or the transitions m and n come one from G_A and the other from G_B. The initial position $*_{A\otimes B}$ is defined as the position $*_A \otimes *_B$.

Projection. Every path s in the 1-Player game $A\otimes B$ may be seen as the interleaving of a path s_A in A, and a path s_B in B. Moreover, every two such interleavings of s_A and s_B are equivalent, modulo homotopy in $A\otimes B$. The path s_A is called the *projection* of the path s on the component A. Similarly, we write $s_{A,B}$ for the projection on the component $A \otimes B$ of a play s in the game $A \otimes B \otimes C$.

Composition. Every pair of strategies $\sigma : A\otimes B$ and $\tau : B\otimes C$ induces by interaction a strategy $\sigma \div \tau$ of the game $A \otimes B \otimes C$, defined as

$$\sigma \div \tau \quad = \quad \{\, s \in A \otimes B \otimes C \quad | \quad s_{A,B} \in \sigma \quad \text{and} \quad s_{B,C} \in \tau \,\}$$

The composite $\sigma;\tau$ is the strategy of $A \otimes C$ defined by hiding the moves in B from the interaction between σ and τ:

$$\sigma;\tau \quad = \quad \{\, s_{A,C} \quad | \quad s \in \sigma \div \tau \,\} \tag{13}$$

The category $\mathcal{O}$ has 1-Player games as objects, and strategies of $A \otimes B$ as morphisms from A to B. The composite of $\sigma : A \to B$ and $\tau : B \to C$ is the strategy $\sigma;\tau : A \to C$ defined in (13). The identity id_A on the game A is the smallest strategy containing the empty play and such that for every play $s : *_A \otimes *_A \longrightarrow\!\!\!\!\!\rightarrow x_A \otimes x_A$ in id_A and every transition $m : x_A \longrightarrow y_A$ in A, the two plays

$$*_A \otimes *_A \overset{s}{\longrightarrow\!\!\!\!\!\rightarrow} x_A \otimes x_A \overset{m}{\longrightarrow} y_A \otimes x_A \overset{m}{\longrightarrow} y_A \otimes y_A$$

and

$$*_A \otimes *_A \overset{s}{\longrightarrow\!\!\!\!\!\rightarrow} x_A \otimes x_A \overset{m}{\longrightarrow} x_A \otimes y_A \overset{m}{\longrightarrow} y_A \otimes y_A$$

are also elements of the strategy id_A. Note that the identity strategy is not closed under prefix: hence, prefix-closed strategies do not form a subcategory of the category $\mathcal{O}$. They do not even form a category, since the prefix-closed variant of our identity strategy is not idempotent.

The category $\mathcal{O}$ equipped with the tensor product $\otimes$ defines a symmetric monoidal category, whose unit is the game I with one unique position. Besides, this category is compact closed, since there is a one-to-one (and natural) relationship between the morphisms from $A \otimes B$ to C and the morphisms from A to $B \otimes C$.

4 Positionality in asynchronous games

Positional strategies (prefix-closed case). A prefix-closed strategy σ is called *positional* when

$$s \cdot u \in \sigma \quad \text{and} \quad s \sim t \quad \text{and} \quad t \in \sigma \quad \text{implies} \quad t \cdot u \in \sigma \tag{14}$$

for every three paths $s, t : * \longrightarrow\!\!\!\!\!\rightarrow x$ and $u : x \longrightarrow\!\!\!\!\!\rightarrow y$. This definition adapts the definition (4) to 1-Player games, and in fact, applies in the just same way to the non-alternating setting introduced later on in the article, in Section 6.

A positional prefix-closed strategy is the same thing as a subgraph of the 1-Player game, where every position is reachable from the initial position $*$ inside the subgraph. This subgraph inherits a 2-dimensional structure from the underlying 1-Player game. This defines an asynchronous graph, denoted G_σ, from which the strategy σ can be recovered as the set of all plays of G_σ.

It is conceptually remarkable that our notion of positional strategy is of the same nature as the notion of asynchronous game. In fact, we have just shown that a positional prefix-closed strategy $\sigma : A$ is the same thing as a *subgame* of the original game A, in the following sense:

Definition 5 (subgame) *An asynchronous game $(G_2, \diamond_2)$ is a* subgame *of an asynchronous game $(G_1, \diamond_1)$ when G_2 is a subgraph of G_1, with the same initial position, and moreover, the homotopy relation $\diamond_2$ is inherited from the homotopy relation $\diamond_1$, in the sense that*

$$m \cdot p \diamond_1 n \cdot q \quad \textit{iff} \quad m \cdot p \diamond_2 n \cdot q$$

for all transitions m, n, p, q of the graph G_2.

This generalizes the well-known principle that a prefix-closed strategy σ of a decision tree A, is the same thing as a subtree of A.

Positional strategies (general case). Now, we extend our notion of positional strategy to the general case of a non necessarily prefix-closed strategy. By definition, such a strategy σ is called *positional* when its prefix-closure $\mathrm{prefix}(\sigma)$ is positional in the previous sense, and moreover

$$s \cdot u \in \sigma \quad \text{and} \quad s \sim t \quad \text{and} \quad t \in \sigma \quad \text{implies} \quad s \in \sigma$$

for every three paths $s, t : * \longrightarrow\!\!\!\!\!\rightarrow x$ and $u : x \longrightarrow\!\!\!\!\!\rightarrow y$. From this definition, it follows that a positional strategy σ is characterized by the asynchronous graph $G_{\mathrm{prefix}(\sigma)}$ induced by its prefix-closure $\mathrm{prefix}(\sigma)$ – which will be noted G_σ from now on – together with the set $\sigma^\bullet$ of *acceptance positions* of σ defined as

$$\sigma^\bullet \quad = \quad \{\ x \quad | \quad \exists s : * \longrightarrow\!\!\!\!\!\rightarrow x, s \in \sigma\ \}.$$

The strategy σ can be reconstructed from the graph G_σ and the set of acceptance positions $\sigma^\bullet$ as follows:

$$\sigma \quad = \quad \{\, s : * \longrightarrow\!\!\!\!\!\rightarrow x \quad | \quad s \in G_\sigma \quad \text{and} \quad x \in \sigma^\bullet \,\}.$$

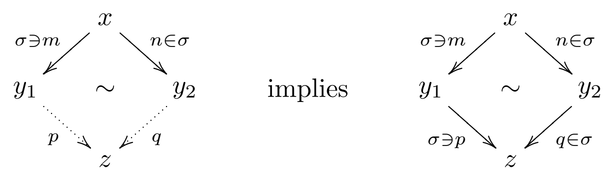
Note that every maximal position x of the graph G_σ associated to a positional strategy σ, belongs to the set $\sigma^\bullet$ of its acceptance positions. Conversely, a subgame G of 1-Player game A and a set of positions X in G is of the form $G = G_\sigma$ and $X = \sigma^\bullet$ for a positional strategy σ when every maximal position of G is element of X.

Asynchronous graphs vs. event structures. The advantage of considering asynchronous graphs instead of event structures appears at this point of the article. In our philosophy, the name of the "event" associated to a given transition should be deduced from the 2-dimensional geometry of the graph. Typically, the "event" associated to a transition m in the graph G_σ describes the "causality cascade" leading the strategy σ to play the transition m. On the other hand, the "event" associated to the same transition m, but seen this time in the 1-Player game G, is simply the name of the move in the game. This subtle difference is precisely what underlies the distinction between the names of events in the formula (5) and in the various strategies (9) and (10). For instance, there are three "events" associated to the output move `false` in the parallel implementation of the strict conjunction, each one corresponding to a particular pair of inputs (`true`, `false`), (`false`, `false`), and (`false`, `true`). This ability to name a transition by the causal cascade which produced it, follows from the cube property – an avatar of Berry stability, already noticed in [21].

5 Interlude: from sequences to positions

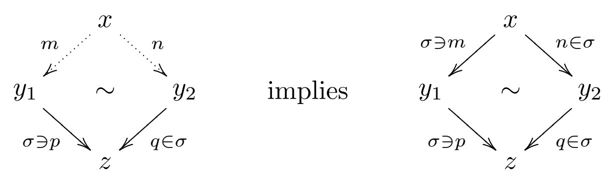
From now on, we only consider positional strategies satisfying two additional properties.

1. *Forward compatibility preservation*: every asynchronous tile of the shape (12) in the 1-Player game G belongs to the subgraph G_σ of the strategy σ when its two coinitial transitions $m : x \longrightarrow y_1$ and $n : x \longrightarrow y_2$ are transitions in the subgraph G_σ. Diagrammatically,



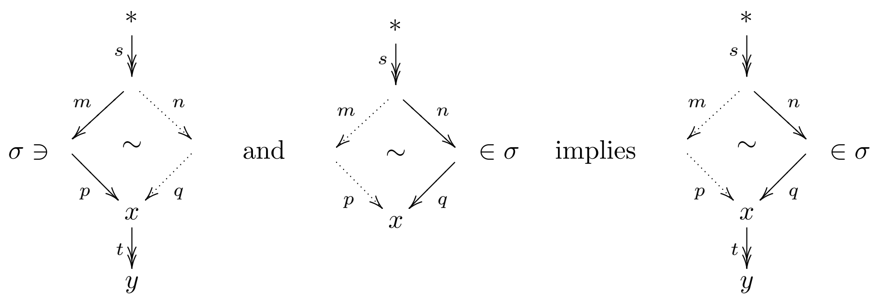
   where the dotted edges indicate edges in G.

2. *Backward compatibility preservation*: dually, every asynchronous tile of the shape (12) in the 1-Player game G belongs to the subgraph G_σ of the strategy σ when its two cofinal transitions $p : y_1 \longrightarrow z$ and $q : y_2 \longrightarrow z$ are transitions in the subgraph G_σ. Diagrammatically,

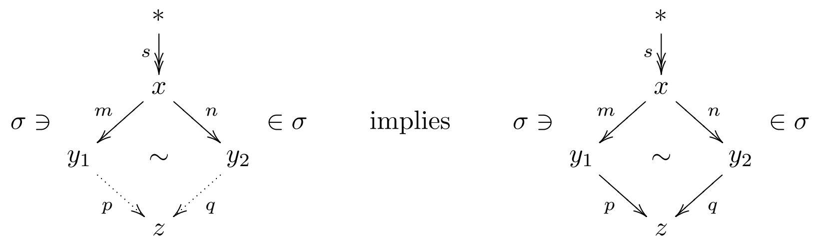


These two properties ensure that the asynchronous graph G_σ is contractible and satisfies the cube property. Contractibility means that every two cofinal plays $s, t : * \twoheadrightarrow x$ of the strategy σ are equivalent modulo homotopy *inside* the asynchronous graph G_σ – that is, every intermediate play in the homotopy relation is an element of σ. Moreover, there is a simple reformulation as a set of plays of a positional strategy. Namely, a prefix-closed strategy σ is positional and satisfies the two preservation properties if and only if it satisfies the following properties.

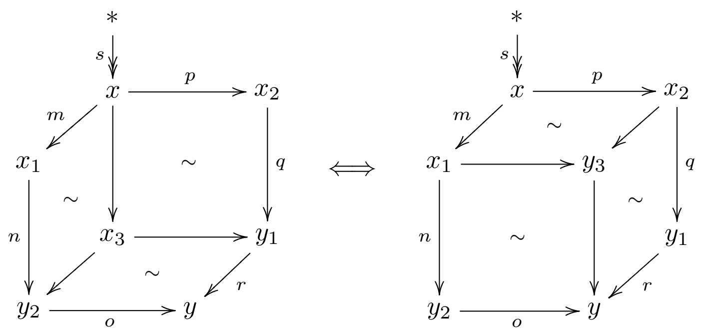
1. For every two plays $s \cdot m \cdot p : * \twoheadrightarrow x$ and $s \cdot n \cdot q : * \twoheadrightarrow x$ and for every path $t : x \twoheadrightarrow y$, we have: $s \cdot m \cdot p \cdot t \in \sigma$, $m \cdot p \sim n \cdot q$ and $s \cdot n \cdot q \in \sigma$ implies $s \cdot n \cdot q \cdot t \in \sigma$. Diagrammatically,



2. For every play $s : * \twoheadrightarrow x$ and transitions $m : x \longrightarrow y_1$ and $n : x \longrightarrow y_2$ such that there exist a position z and two transitions $p : y_1 \longrightarrow z$ and $q : y_2 \longrightarrow z$ forming a tile (12), if the paths $s \cdot m$ and $s \cdot n$ are in σ then the paths $s \cdot m \cdot p$ and $s \cdot n \cdot q$ are also in σ. Diagrammatically,



3. For every two plays $s \cdot m \cdot n \cdot o : * \twoheadrightarrow y$ and $s \cdot p \cdot q \cdot r : * \twoheadrightarrow y$ in σ, the series of homotopic paths on the left-hand side of the diagram below are in σ if and only if the series of homotopic paths on the right-hand side of the diagram are in σ:



This characterization is useful, because it enables us to manipulate a positional strategy either as a set of plays, or as an asynchronous subgraph of the game. The characterization is extended to general (that is, non prefix-closed) strategies in the obvious way: one simply requires

- a variant of property 1: for every three paths $s \cdot m \cdot p : * \twoheadrightarrow x$, $s \cdot n \cdot q : * \twoheadrightarrow x$ and $t : x \twoheadrightarrow y$, such that $s \cdot m \cdot p \cdot t \in \sigma$, $m \cdot p \sim n \cdot q$ and $s \cdot n \cdot q \in \text{prefix}(\sigma)$, one requires that $s \cdot n \cdot q \cdot t \in \sigma$.
- the properties 2. and 3. on the prefix-closure $\text{prefix}(\sigma)$ of the strategy.

Although they are very natural, the properties of positionality as well as the two additional properties of preservation of compatibility are not preserved by composition: the composite $\sigma;\tau$ of two positional strategies σ and τ is not necessarily positional, etc. This seems intimately related to the fact that our current games are 1-Player games. Hence, we introduce in the next section a category $\mathcal{G}$ of 2-Player games and ingenuous strategies, for which these properties are preserved by composition.

6 Ingenuous strategies in 2-Player games

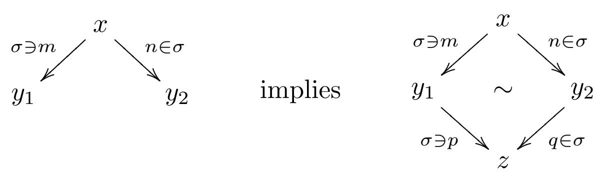
Our notion of 2-Player game is simply obtained by separating the moves of our 1-Player games in two classes: Opponent moves and Proponent moves.

Definition 6 (game) *A* 2-Player game $(G, *, \lambda)$ *is a 1-Player game* $(G, *)$ *together with a function* $\lambda : E \to \{-1, +1\}$ *which associates a polarity to every transition (or move) of the underlying graph* $G = (V, E, \diamond)$.

The convention is that a move m is played by *Proponent* when $\lambda(m) = +1$ and by *Opponent* when $\lambda(m) = -1$. Moreover, one requires that polarities match in homotopy tiles, in the sense that $\lambda(m) = \lambda(q)$ and $\lambda(n) = \lambda(p)$ for every asynchronous tile (12) of the asynchronous graph G.

Ingenuous strategies. Strategies of 2-Player games are defined as sets of plays, in the same way as in 1-Player games. However, shifting from 1-Player games to 2-Player games enables us to define the following notion of ingenuous strategy. A strategy σ is called *ingenuous* when it satisfies the following properties.

1. It is *positional*, and satisfies the backward and forward compatibility preservation properties of Section 5,
2. It is *deterministic*, in the following concurrent sense: every pair of coinitial moves $m : x \longrightarrow y_1$ and $n : x \longrightarrow y_2$ in the strategy σ where the move m is played by Proponent, induces an asynchronous tile (12) in the strategy σ. Diagrammatically,



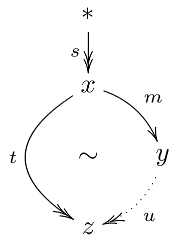
3. It is *conflict-free*: every acceptance position is halting.

A position x is *halting* in a strategy σ when it is reached by a prefix $s : * \longrightarrow\!\!\!\!\!\rightarrow x$ of the strategy, and there exists no Proponent move $m : x \longrightarrow y$ such that $s \cdot m$ is prefix of the strategy σ. These halting positions are thus positions where the strategy has played all its moves, and is either waiting for an input from the Opponent, or is not willing to interact anymore.

Note that, for simplicity, we express the condition of determinism on strategies seen as asynchronous subgames of the original game. However, this condition may be reformulated in a straightforward fashion on strategies defined as sets of plays, following the methodology of Section 5. The forward and backward compatibility preservation properties of Section 5 ensure that the set of plays of the strategy σ reaching the same position x is regulated by a "causality order" on the moves occurring in these plays – which refines the "justification order" on moves (in the sense of pointer games) provided by the asynchronous game.

Our concurrent notion of determinism is not exactly the same as the usual notion of determinism in sequential games: in particular, a strategy may play several Proponent moves from a given position, as long as it converges later.

Together with the two first conditions, the last property is equivalent to the notion of conflict-freeness introduced independently in [16] and [25]. This condition is a kind of confluence property which can be formulated as follows. A strategy σ is *conflict-free* when for every play $s : * \longrightarrow\!\!\!\!\!\rightarrow x$ prefix of a play $s \cdot t : * \longrightarrow\!\!\!\!\!\rightarrow z$ in the strategy σ, and for every Proponent transition $m : x \longrightarrow y$ such that the play $s \cdot m$ is also prefix of a play in σ, there exists a path $u : y \longrightarrow\!\!\!\!\!\rightarrow z$ such that $t \sim m \cdot u$ and the play $s \cdot m \cdot u$ is element of the strategy σ. Diagrammatically,



The idea is that when a conflict-free strategy is ready to accept a position z in the future, it should play so as to keep the ability to reach this position z at a later stage, unless Opponent decides to orient the interaction in another direction.

Composition. The tensor product $A \otimes B$ of two 2-Player games A and B is defined in the same way as for 1-Player games, with the polarity of moves preserved in each component. The dual A^* of a 2-Player game A is simply the game A with polarities reversed. The linear implication $A \multimap B$ is defined as $A^* \otimes B$, which is also equal to $(A \otimes B^*)^*$. We like to write $x_A \multimap x_B$ for the position of $A \multimap B$ consisting of the position x_A in the game A and the position x_B in the game B.

The composite $\sigma;\tau : A \multimap C$ of two strategies $\sigma : A \multimap B$ and $\tau : B \multimap C$ is defined exactly in the same way in 2-Player games as in 1-Player games. Moreover, the composite of two ingenuous strategies is also ingenuous. The proof of this statement is not entirely obvious. It is based on a subtle confluence property, stated below.

Property 1 *Suppose that $\sigma : A \multimap B$ and $\tau : B \multimap C$ are two ingenuous strategies and that*

$$u : * \longrightarrow\!\!\!\!\!\rightarrow x \multimap z$$

is a play in the composite strategy $\sigma;\tau$. Suppose also that there exist two plays

$$s_1 : * \longrightarrow\!\!\!\!\!\rightarrow x \multimap y_1 \quad \textit{and} \quad s_2 : * \longrightarrow\!\!\!\!\!\rightarrow x \multimap y_2$$

in σ *and two plays*

$$t_1 : * \twoheadrightarrow y_1 \multimap z \quad \textit{and} \quad t_2 : * \twoheadrightarrow y_2 \multimap z$$

in τ *such that*

$$(s_1)_A = (s_2)_A, \quad (s_1)_B = (t_1)_B, \quad (s_2)_B = (t_2)_B, \quad (t_1)_C = (t_2)_C$$

then there exist a position y *and two plays*

$$s : * \twoheadrightarrow x \multimap y \quad \textit{and} \quad t : * \twoheadrightarrow y \multimap z$$

respectively in σ *and in* τ *such that*

$$s_A = (s_1)_A = (s_2)_A, \quad s_B = t_B, \quad t_C = (t_1)_C = (t_2)_C$$

and both s_1 *and* s_2 *(resp.* t_1 *and* t_2*) are prefixes of* s *(resp. of* t*) modulo homotopy.*

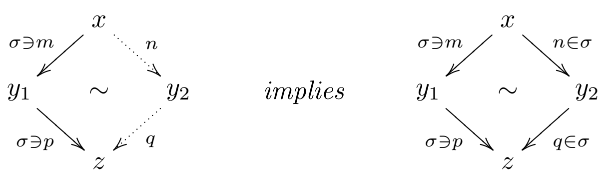
This property states that given two interactions between the strategies σ and τ leading to the same play u in the composite strategy $\sigma;\tau$, their union (wrt. the prefix order modulo homotopy) can be reached by interaction. This property of "maximal witness" is fundamental to establish, for instance, that the composite strategy $\sigma;\tau$ is deterministic. Suppose given a path $u : * \twoheadrightarrow x \multimap z$, and two Proponent transitions $m : x \multimap z \longrightarrow x \multimap z_1$ and $n : x \multimap z \longrightarrow x \multimap z_2$ in the component C such that m and n are played by the composite strategy $\sigma;\tau$. The play $u \cdot m$ (resp. $u \cdot n$) results from the interaction of two plays $s_1 \in \sigma$ and $t_1 \cdot m \in \tau$ (resp. $s_2 \in \sigma$ and $t_2 \cdot n \in \tau$). Property 1 ensures that the two plays $u \cdot m$ and $u \cdot n$ result from the interaction of essentially the same pair of plays: $s \in \sigma$ and $t \cdot m \in \tau$ for $u \cdot m$, $s \in \sigma$ and $t \cdot n \in \tau$ for $u \cdot n$. Since the strategy τ is deterministic, the transitions m and n form a tile (12) and the plays $t \cdot m \cdot p$ and $t \cdot n \cdot q$ are in τ, from which we deduce that the paths $m \cdot p$ and $n \cdot q$ are in the strategy $\sigma;\tau$. The other cases can be handled similarly.

This defines a category $\mathcal{G}$ of 2-Player games, whose morphisms from A to B are the ingenuous strategies of the game $A \multimap B$. The category is symmetric monoidal, and in fact compact closed. The obvious forgetful functor from $\mathcal{G}$ to our previous category $\mathcal{O}$ of 1-Player games is monoidal and preserves the duality.

7 Courteous strategies as closure operators

In this section, we investigate a particular class of ingenuous strategies, satisfying an additional property of *courtesy*. These strategies are remarkable, because they are completely determined by their halting positions. This additional courtesy property enables us to relate asynchronous games with concurrent games. From now on, we suppose that all the strategies we consider have their acceptance positions equal to their halting positions. The notation $\sigma^\bullet$ will thus denote the set of halting positions of a strategy σ.

Definition 7 (courtesy) *A strategy* σ *is* courteous *when every asynchronous tile (12) where* m *is a Proponent move, is in the strategy* σ *as soon as the two moves* $m : x \longrightarrow y_1$ *and* $p : y_1 \longrightarrow z$ *are in the strategy* σ*. Diagrammatically,*



This property ensures that a strategy σ which accepts an Opponent move n *after* playing an independent Proponent move m, is ready to delay its own action, and to accept the move n *before* playing the move m. Therefore, the "causality order" on moves induced by such a strategy refines the underlying "justification order" of the game, by adding only order dependencies $m \preceq n$ where m is an Opponent move. This adapts to the non-alternating setting the fact that, in alternating games, the causality order $p \preceq q$ provided by the view of an innocent strategy coincides with the justification order when p is Proponent and q is Opponent.

From positions to strategies. Given a set X of positions of a game A, we define the strategy $X^{\lightning}$ on the game A as the smallest set of plays of A such that:

- $X^{\lightning}$ contains the empty play,
- for every play $s : * \twoheadrightarrow x$ in $X^{\lightning}$, if $m : x \longrightarrow y$ is a transition such that there exists a position $z \in X$ and path $t : y \twoheadrightarrow z$ containing only Proponent moves then the play $s \cdot m : x \twoheadrightarrow y$ is in the strategy $X^{\lightning}$.

This operation provide us with a way to recover a courteous strategy from its halting positions in the following sense.

Proposition 2 *Suppose that the strategy σ is ingenuous and courteous. Then, the strategy σ is characterized by its set of halting positions:*

$$\sigma \quad = \quad (\sigma^{\bullet})^{\lightning}$$

Conversely, it is possible to give straightforward a characterization of the sets of positions which are the sets $\sigma^{\bullet}$ of halting positions of some courteous strategy σ:

Proposition 3 *Given a game A, a set X of positions of A is of the form $X = \sigma^{\bullet}$ for some courteous ingenuous strategy σ if and only if it satisfies the following properties.*

1. *The set X is closed under intersection:*

$$\forall x, y \in X, \quad x \wedge y \in X$$

2. *The set X preserves compatibility: two positions x and y of X which are compatible in A, are also compatible in X.*
3. *For every position x of A which is dominated in X, there exists a position y in X and a sequence of Proponent moves $m_1, \ldots, m_k$ forming a path*

$$x \xrightarrow{m} x_1 \cdots x_{k-1} \xrightarrow{m_k} y$$

4. *For every pair of positions $x, z \in X$ such that there exists a path $s : x \twoheadrightarrow z$, there exist an Opponent transition $m : x \longrightarrow y$ and a path $t : y \twoheadrightarrow z$ containing only Proponent moves.*

This extends to our non-alternating setting the characterization of [23].

The first property implies that the set X is a Moore family on the poset D_A of positions of the game A completed with a top element, which is a complete lattice. As such, this family X induces a closure operator on the poset D_A, defined by

$$\sigma \quad = \quad x \quad \mapsto \quad \bigwedge \{\, y \in X \quad | \quad x \leq y \,\} \tag{15}$$

Conversely, the set X can be recovered as the set

$$\mathrm{fix}(\sigma) \quad = \quad \{\, x \in D_A \quad | \quad \sigma(x) = x \,\} \tag{16}$$

of fixpoints of the closure operator. The set (16) of fixpoints of a closure operator is always a Moore family and moreover, the transformations (15) and (16) are inverse operations. These transformations thus allow us to recover the model of concurrent games [2].

A category of courteous strategies. It should be noted that the composite $\sigma;\tau : A \to C$ of two courteous and ingenuous strategies $\sigma : A \to B$ and $\tau : B \to C$ is itself a courteous strategy. In fact, courteous ingenuous strategies define a category $\mathcal{C}$ of 2-Player games and courteous ingenuous strategies. Note that this category is not a subcategory of the category $\mathcal{G}$ of ingenuous strategies. Indeed, the courteous identity on a game A is the *buffer* strategy $\mathrm{buf}_A : A \to A$ defined as the smallest courteous strategy containing the identity id_A in $\mathcal{G}$.

On the other hand, an ingenuous strategy $\sigma : A \to B$ is courteous if and only if it is invariant by composition with the buffer strategy in the sense that the equalities

$$\mathrm{buf}_A;\sigma \quad = \quad \sigma \quad = \quad \sigma;\mathrm{buf}_B$$

hold, thus adapting in our setting the notion of buffered asynchronous agent introduced by Selinger [31]. In that respect, the category of courteous strategies may be deduced conceptually from the category $\mathcal{G}$.

One could expect the correspondence between courteous strategies ingenuous strategies to extend into a functor from the category $\mathcal{C}$ of courteous strategies to the category of concurrent games – the composition of closure operators is defined in [2] by a fixpoint construction and corresponds precisely, by the transformations (15) and (16), to the relational composition of the sets fixpoints of the closure operators. However, we explain in the next section why it is not the case.

8 A lax functor to the relational model

In Section 4, we showed how a strategy $\sigma : A \to B$ in the category $\mathcal{G}$ induces a set $\sigma^\bullet$ of positions of the game $A \multimap B$. Since every such position is a couple (x_A, x_B), also noted $x_A \multimap x_B$, consisting of a position x_A of A and a position x_B of B, the set $\sigma^\bullet$ may be seen alternatively as a relation between the positions of A and the positions of B. This provides a translation from strategies to relations which is not functorial, because of a subtle mismatch between the way morphisms are composed. Hence, if $\sigma : A \to B$ and $\tau : B \to C$ denote two strategies, the equality

$$\sigma^\bullet;\tau^\bullet \quad = \quad (\sigma;\tau)^\bullet$$

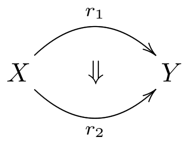
between the relational composite $\sigma^\bullet;\tau^\bullet$ of their sets of positions, and the set of positions $(\sigma;\tau)^\bullet$ of their composite does not necessarily hold.

From static to dynamic composition. However, the inclusion

$$\sigma^\bullet;\tau^\bullet \quad \supseteq \quad (\sigma;\tau)^\bullet$$

holds, for the following reason. By definition, every position $x \multimap y$ in $(\sigma;\tau)^\bullet$ is reached by a play $u : * \twoheadrightarrow x \multimap z$ in the composite strategy $\sigma;\tau$. By definition again, this play results from the interaction of a play $s : * \twoheadrightarrow x \multimap y$ in the strategy σ

and of a play $t : * \longrightarrow\!\!\!\!\rightarrow y \multimap z$ in the strategy τ. The position $x \multimap z$ is therefore in $\sigma^\bullet;\tau^\bullet$, since $x \multimap y \in \sigma^\bullet$ and $y \multimap z \in \tau^\bullet$. Technically speaking, this makes the operation $(-)^\bullet$ a lax functor from the category $\mathcal{G}$ to the 2-category of relations – where the 2-dimensional cells of the category of relations



indicate that the relation $r_2 : X \to Y$ is included in the relation $r_1 : X \to Y$.

From dynamic to static composition. In contrast, the other inclusion

$$\sigma^\bullet;\tau^\bullet \quad \subseteq \quad (\sigma;\tau)^\bullet$$

does not necessarily hold. For example, the strategy $\sigma : I \to \mathbb{B} \otimes \mathbb{B}$ pictured in (11) implements the pair (`true`, `false`) which is not willing to accept questions for its right component before it has given the value of its left component. Now, compose it with the strategy $\tau : \mathbb{B} \otimes \mathbb{B} \to \mathbb{B}$, which is the implementation of the conjunction pictured on the right-hand side of (9) which queries the value of its right argument, then the value of its left argument, and then computes the answer. The positions accepted by the strategy σ are

$$\sigma^\bullet \quad = \quad \{* \otimes *, \quad V \otimes *, \quad V \otimes F\}$$

and the set $\tau^\bullet$ of positions accepted by the strategy τ contains

$$\{(* \otimes *) \multimap *, \quad (* \otimes q) \multimap q, \quad (q \otimes F) \multimap q, \quad (V \otimes F) \multimap F\}$$

Therefore, the composite relation $\sigma^\bullet;\tau^\bullet$ contains the position F, although this position is not in $(\sigma;\tau)^\bullet$. This detects a *deadlock* during the interaction of the two strategies, preventing this position to be actually reached: at the position $* \otimes *$ in $\mathbb{B} \otimes \mathbb{B}$, the strategy σ wants to be asked a question on its left component whereas the strategy τ first asks for the right component of the pair. More conceptually, this phenomenon comes from the fact that the category $\mathcal{G}$ is compact closed: the tensor product $\otimes$ is identified with its dual $⅋$. In order to prevent such situations from happening, we impose a further criterion on strategies, which is detailed in next section.

9 Scheduling tests and orthogonality

We now strengthen the notion of ingenuous strategies by imposing a *scheduling criterion* which distinguishes the tensor product $\otimes$ from its dual $⅋$. This criterion plays the rôle in our non-alternating framework of the *switching conditions* introduced by Abramsky and Jagadeesan for alternating games [1]. As we will see, the criterion ensures that strategies do not deadlock during composition, and thus, that the sets of acceptance positions reflect properly how strategies compose – this turning the lax functor of Section 8 into a strict functor from asynchronous strategies to relations.

Scheduling. Our scheduling criterion is based on the idea that a strategy of a given type A should be able to reorganize its interactive behaviour according to the scheduling selected by the Opponent. Every such scheduling is described as a particular kind of ingenuous *counter-strategy* of the original game A, that is a strategy of the dual game A^*.

Definition 8 (scheduling) *A scheduling $\mathscr{S}$ is an ingenuous strategy $\mathscr{S}$ in which every two plays $s \in \mathscr{S}$ and $t \in \mathscr{S}$ are either equal, or incomparable wrt. the prefix ordering modulo homotopy.*

In other words, every accepted position in a scheduling $\mathscr{S}$ is maximal in the asynchronous graph associated to the scheduling.

Informally, a strategy σ adapts to the Opponent scheduling $\mathscr{S}$ when for every play $s \in \sigma$ homotopic to a play $t \in \mathscr{S}$, there exists a play u homotopic to s and t, and element of both strategies: $u \in \sigma$ and $u \in \mathscr{S}$. The main difficulty is to understand what scheduling should be allowed to test a strategy of given type A. Of course, every counter-strategy may be seen as a particularly clever kind of scheduling. Our task is thus to detect a class of basic counter-strategies – precisely what we like to call scheduling tests – generating in some appropriate sense the class of all possible counter-strategies of type A.

The solution appears to be much simpler than expected. Indeed, we will establish below that in order to test a strategy of type A, it is sufficient to "switch" every tensor product $\otimes$ of the formula into either a left-to-right or a right-to-left scheduling, noted $\olessthan$ and $\ogreaterthan$. Semantically, every such switching is interpreted as the associated connective on schedulings.

Definition 9 *Given a scheduling $\mathscr{S}$ of an asynchronous game A, and a scheduling $\mathscr{T}$ of an asynchronous game B, the scheduling $\mathscr{S} \otimes \mathscr{T}$ (called "$\mathscr{S}$ times $\mathscr{T}$") is the ingenuous strategy of $A \otimes B$ containing the plays s such that*

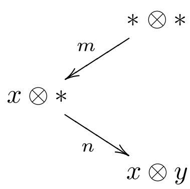
- *the projection s_A on the component A is a play of $\mathscr{S}$,*
- *the projection s_B on the component B is a play of $\mathscr{T}$.*

The scheduling $\mathscr{S} \olessthan \mathscr{T}$ (called "$\mathscr{S}$ before $\mathscr{T}$") is the ingenuous strategy of $A \otimes B$ containing the plays s of the scheduling $\mathscr{S} \otimes \mathscr{T}$ satisfying moreover:

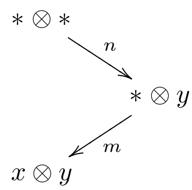
- *no move of component B is performed before a move of component A,*
- *if the play s contains a move in component B, its projection s_A on the component A is a (necessarily maximal) play of the schedule $\mathscr{S}$.*

Similarly, the scheduling $\mathscr{T} \ogreaterthan \mathscr{S}$ (called "$\mathscr{S}$ after $\mathscr{T}$") is the ingenuous strategy of $A \otimes B$ containing the plays s of the scheduling $\mathscr{S} \otimes \mathscr{T}$ such that the projection s_B on the component B is a (necessarily maximal) play of the schedule $\mathscr{T}$ when the projection s_A on the component A is non-empty.

Note that the scheduling $\mathscr{S} \olessthan \mathscr{T}$ of $A \otimes B$ is the same thing as the scheduling $\mathscr{T} \ogreaterthan \mathscr{S}$ of $B \otimes A$, after permutation of A and B. For example, if both $\mathscr{S}$ and $\mathscr{T}$ play exactly one transition noted $m : * \longrightarrow x$ in A and $n : * \longrightarrow y$ in B, the schedulings $\mathscr{S} \olessthan \mathscr{T}$ and $\mathscr{T} \ogreaterthan \mathscr{S}$ are



and



in $A \otimes B$ and $B \otimes A$ respectively. Note that x and y are not necessarily maximal positions of the games A and B – although they are maximal in $\mathscr{S}$ and $\mathscr{T}$.

Scheduling tests. From now on, we equip every game A with two non-empty sets A_P and A_O of scheduling of the asynchronous game A – called respectively the sets of Proponent and Opponent *scheduling tests*. Hence, if U and V are two sets of scheduling tests, we write $U \otimes V$ for the set

$$U \otimes V \quad = \quad \{\, \mathscr{S} \otimes \mathscr{T} \quad | \quad \mathscr{S} \in U \quad \text{and} \quad \mathscr{T} \in V \,\}$$

and similarly for $U \olessthan V$ and $U \ogreaterthan V$.

Tensor product. For every two games A and B, the scheduling tests of $A \otimes B$ are defined as

$$(A \otimes B)_P = A_P \otimes B_P \quad \text{and} \quad (A \otimes B)_O = A_O \olessthan B_O \cup A_O \ogreaterthan B_O.$$

This means that a strategy of type $A \otimes B$ should be flexible enough to adapt to a left-to-right as well as to right-to-left scheduling of the components A and B. On the other hand, a counter-strategy of type $A \otimes B$ is only required to interact properly with a scheduling of A and a scheduling of B, with arbitrary interleaving of the two components A and B.

Negation. The scheduling tests of the game A^* are obtained in the expected way, by exchanging the rôles of the two players in A:

$$(A^*)_P = A_O \quad \text{and} \quad (A^*)_O = A_P$$

Linear implication. The game $A \multimap B$ is then defined by the usual de Morgan equality $A \multimap B = (A \otimes B^*)^*$. Hence, a strategy of $A \multimap B$ should react properly to the scheduling tests in $A_P \otimes B_O$. Note moreover that $A \multimap B$ is not isomorphic in general to $A^* \otimes B$.

Strategies with interaction positions. From now on, we equip our ingenuous strategies $\sigma : A$ with a set X of positions of A, called the *interaction positions* of the strategy. We require that

1. all the acceptance positions of σ are interaction positions: $\sigma^\bullet \subseteq X$,
2. all the interaction positions are *respected* by σ, in the sense that for every prefix $s \cdot m$ of a play in the strategy σ, where m is a Proponent move, if the play s can be extended into a play $s \cdot t$ reaching the position x in the game, then the play $s \cdot m$ can also be extended into a play $s \cdot m \cdot u$ reaching the position x in the game.

Note that every acceptance position is halting, and thus respected by the ingenuous strategy σ. This follows from the property of conflict-freeness discussed in Section 6.

The category $\mathcal{H}$ has 2-Player games as objects, and pairs (σ, X) as morphisms between two games A and B – where σ is an ingenuous strategy from A to B in the category $\mathcal{G}$, and X is a set of interaction positions of σ. Composition $(\sigma, X) \bullet (\tau, Y)$ of two morphisms (σ, X) and (τ, Y) is defined by

$$(\sigma, X) \bullet (\tau, Y) \quad = \quad (\sigma; \tau, X; Y)$$

where the first component is composed in the category $\mathcal{G}$ and the second component is composed in the category of sets and relations. Similarly, the category $\mathcal{H}$ is monoidal with the tensor product of two strategies (σ, X) and (τ, Y) given by

$$(\sigma, X) \otimes (\tau, Y) \quad = \quad (\sigma \otimes \tau, X \otimes Y)$$

where the tensor product in the first component is computed in the category $\mathcal{G}$ and the tensor product in the second component is computed in the category of sets and relations – and given by the usual (set-theoretic) cartesian product. Very often, we write $\mathscr{S}$ for the pair $(\mathscr{S}, \mathscr{S}^\bullet)$ induced by a scheduling test $\mathscr{S}$.

Orthogonality. The idea of testing a strategy σ against a scheduling $\mathscr{S}$ is nicely captured (and generalized) by the following notion of *orthogonality* between a strategy (σ, X) and a counter-strategy (τ, Y) of the game A.

Definition 10 (orthogonality) *Two strategies* $(\sigma, X) : A$ *and* $(\tau, Y) : A^*$ *are* orthogonal*, what we write* $(\sigma, X) \perp (\tau, Y)$*, when*

$$X \cap Y \quad = \quad (\sigma \cap \tau)^\bullet$$

Here, the intersection $\sigma \cap \tau$ is the strategy obtained by intersecting σ and τ, seen as sets of plays. Hence, the two strategies σ and τ are orthogonal precisely when, for every position $x \in X \cap Y$, there exists a play $s : * \twoheadrightarrow x$ reaching the position x, and common element of the two strategies σ and τ.

Lemma 1 *Suppose that* (σ, X) *is a strategy and* (τ, Y) *is a counter-strategy on a game* A*. Then,*

$$(\sigma, X) \perp (\tau, Y) \quad \textit{implies} \quad (\sigma \cap \tau)^\bullet \quad \textit{is empty or singleton.}$$

Proof. The fact that two orthogonal ingenuous strategies have at most one acceptance position in common follows easily from the conflict-freeness property discussed in Section 6. □

Some properties of orthogonality. The following lemmas will be useful to prove Property 2 in the next section.

Lemma 2 *Given a counter-strategy* $\tau = (\tau, Y)$ *of* $A \otimes B$ *and two strategies* $\sigma_A = (\sigma_A, X_A)$ *in* A *and* $\sigma_B = (\sigma_B, X_B)$ *in* B*, the following statements are equivalent:*

(i) $\sigma_A \otimes \sigma_B \perp \tau$*,*

(ii) $\sigma_A \perp \tau \bullet \sigma_B$ *and* $\sigma_B \perp \sigma_A \bullet \tau$*.*

Proof. (i) ⇒ (ii). Suppose that $\sigma_A \otimes \sigma_B \perp \tau$ and x_A is a position in $X_A \cap (Y; X_B)$. Since $x_A \in (Y; X_B)$, there exists a position x_B in X_B such that $x_A \otimes x_B \in Y$. Moreover, the position $x_A \otimes x_B$ is an element of in $X_A \otimes X_B$. By our hypothesis, we deduce that there exists a play $s : * \twoheadrightarrow x_A \otimes x_B$ which is common to the strategies $\sigma_A \otimes \sigma_B$ and τ, and by projection, the play $s_A : * \twoheadrightarrow x_A$ is in the strategy σ_A and the play $s_B : * \twoheadrightarrow x_B$ is in the strategy σ_B. The play $s_B : * \twoheadrightarrow x_B$ results from the interaction of the plays s and s_A and is therefore in the strategy $\sigma_A; \tau$: the position x_A is reached by the play s_A which is common to the strategies σ_A and $\tau; \sigma_B$. Finally, we can conclude $\sigma_A \perp \tau \bullet \sigma_B$, and similarly $\sigma_B \perp \sigma_B \bullet \tau$.

(ii) ⇒ (i). Suppose that both assertions $\sigma_A \perp \tau \bullet \sigma_B$ and $\sigma_B \perp \sigma_A \bullet \tau$ are satisfied and that x is an acceptance position of both $\sigma_A \otimes \sigma_B$ and τ. The position x can be decomposed as $x = x_A \otimes x_B$ where x_A is an acceptance position of σ_A and x_B is an acceptance position of σ_B. Since the position x_A is in both X_A and $Y; X_B$, by our first hypothesis, there exists a path $s_A : * \twoheadrightarrow x_A$ common to the strategies σ_A and $\tau; \sigma_B$. This path is obtained by the interaction between τ and σ_B which means that there exists a path $s : * \twoheadrightarrow x_A \otimes y_B$ in the strategy τ whose projection on the game A is the path s_A, and whose projection $s_B : * \twoheadrightarrow y_B$ on the game B is in the strategy σ_B. The positions x_B and y_B are both in X_B and in $Y; X_A$, therefore by using our second hypothesis and Lemma 1, these positions are equal: $x_B = y_B$. Finally, the path $s : * \twoheadrightarrow x_A \otimes x_B$ is in both the strategies $\sigma_A \otimes \sigma_B$ and τ, from which we can conclude. □

Lemma 3 *Given a counter-strategy $\tau = (\tau, Y)$ of $A \otimes B$, a strategy $\sigma_A = (\sigma_A, X_A)$ in A and a scheduling test $\mathscr{S}_B$ in B, the following statements are equivalent:*

(i) $\sigma_A \otimes \mathscr{S}_B \perp \tau$,

(ii) $\sigma_A \perp \tau \bullet \mathscr{S}_B$.

Proof. The (i) ⇒ (ii) direction is proved as in previous lemma. Conversely, for the (ii) ⇒ (i) direction, the proof of previous lemma needs to be adapted a bit more. Suppose that $\sigma_A \perp \tau \bullet \mathscr{S}_B$ and $x_A \otimes x_B$ is an acceptance position of both strategies $\sigma_A \otimes \mathscr{S}_B$ and τ. As in the previous proof, we show the existence of a position y_B in B and a play $s : * \twoheadrightarrow x_A \otimes y_B$ in the strategy τ such that the play $s_B : * \twoheadrightarrow x_B$ is in the strategy $\mathscr{S}_B$. Moreover, the hypothesis that all interactive positions in $Y; \mathscr{S}_B^\bullet$ and in $\mathscr{S}_B^\bullet$ are respected by $\tau; \mathscr{S}_B$ and τ, respectively, implies that there exists a path $t : y_B \twoheadrightarrow x_B$ in the game B. Now, by definition of a scheduling test, every play in $\mathscr{S}_B$ is maximal in $\mathscr{S}_B$ wrt. the prefix ordering modulo homotopy. Moreover, the two positions x_B and y_B are acceptance positions of the scheduling test $\mathscr{S}_B$. From this, we deduce that the positions x_B and y_B are equal, and conclude as in the proof of the previous lemma. □

10 Scheduled games and strategies

From now on, given a set U of counter-strategies of A, we write $\sigma \perp U$ to mean that $\sigma \perp \tau$ for every counter-strategy τ of U. Similarly, if U is a set of strategies of A, we write $U \perp \tau$ to mean that $\sigma \perp \tau$ for every strategy σ in U.

Scheduled games. Our philosophy of testing strategies by basic scheduling requires that we establish that the following notion of *scheduled game* is closed under tensor product.

Definition 11 (scheduled game) *A game (A, A_P, A_O) is* scheduled *when*

1. *every ingenuous strategy $\sigma = (\sigma, X)$ and ingenuous counter-strategy $\tau = (\tau, Y)$ of the game satisfy:*

$$\sigma \perp A_O \quad \textit{and} \quad A_P \perp \tau \quad \textit{implies} \quad \sigma \perp \tau$$

2. *every scheduling test $\mathscr{S}$ in A_P and $\mathscr{T}$ in A_O are orthogonal: $\mathscr{S} \perp \mathscr{T}$,*
3. *for every position x of the game, there exists a Proponent scheduling test $\mathscr{S} \in A_P$ and an Opponent scheduling test $\mathscr{T} \in A_O$ such that x is an acceptance position of $\mathscr{S}$ and of $\mathscr{T}$.*

Hence, a game is scheduled when every pair of strategy σ and couter-strategy τ interact properly (they are orthogonal) as soon as they react properly to the scheduling tests; Proponent and Opponent scheduling tests are orthogonal; and every position is accepted by an Opponent and by a Proponent scheduling test.

Main property. We establish now the key property which underlies our construction:

Property 2 *The tensor product $A \otimes B$ of two scheduled games (A, A_P, A_O) and (B, B_P, B_O) is also scheduled.*

Proof. Suppose indeed that a strategy $\sigma = (\sigma, X)$ and a counter-strategy $\tau = (\tau, Y)$ of the game $A \otimes B$ satisfy:

$$\sigma \perp (A \otimes B)_O \quad \text{and} \quad (A \otimes B)_P \perp \tau. \tag{17}$$

By definition of $(A \otimes B)_O$, the first statement is equivalent to

$$\sigma \perp A_O \oslash B_O \quad \text{and} \quad \sigma \perp A_O \obslash B_O. \tag{18}$$

Every play $s \in \sigma$ reaches a position x which decomposes as $x = x_A \otimes x_B$. By hypothesis on A and B, there exists an Opponent scheduling $\mathscr{S}_A \in A_O$ accepting the position x_A, and an Opponent scheduling $\mathscr{S}_B \in B_O$ accepting the position x_B. It follows from (18) that the strategy σ is orthogonal to both scheduling tests $\mathscr{S}_A \oslash \mathscr{S}_B$ and $\mathscr{S}_A \obslash \mathscr{S}_B$. The position $x = x_A \otimes x_B$ is accepted by both σ and $\mathscr{S}_A \oslash \mathscr{S}_B$. Orthogonality of σ and $\mathscr{S}_A \oslash \mathscr{S}_B$ implies that there exists a play t homotopic to s where no move in B is played before a move in A. This demonstrates that, according to the strategy σ, no move in A is causally dependent on a move in B inside the play s. Symmetrically, no move in B is, according to the strategy σ, causally dependent on a move in A inside the play s.

Recall that s_A and s_B denote the projections of the play s on the components A and B, respectively. Causal independence in s means that every interleaving t of the two plays s_A and s_B is an element of the strategy σ. In other words, the strategy σ contains every play t satisfying $t_A = s_A$ and $t_B = s_B$. A most concise way to state this is to say that the strategy $s_A \otimes s_B$ is included in the strategy σ. This decomposition holds for every play s of the strategy σ.

Of course, every other play $t \in \sigma$ induces a strategy $t_A \otimes t_B$ included in the strategy σ. It is not clear at all that the "mixed" strategy $s_A \otimes t_B$ is also included in the strategy σ: the strategy $\sigma = (\sigma, X)$ is included in the strategy $\sigma_A \otimes \sigma_B$ where

$$\sigma_A = \{\ s_A \quad | \quad s \in \sigma\ \} \quad \text{and} \quad X_A = \{\ x_A \quad | \quad \exists x_B, x_A \otimes x_B \in X\ \}$$

$$\sigma_B = \{\ s_B \quad | \quad s \in \sigma\ \} \quad \text{and} \quad X_B = \{\ x_B \quad | \quad \exists x_A, x_A \otimes x_B \in X\ \}$$

but it is not true in general that $\sigma = \sigma_A \otimes \sigma_B$. We claim however that the strategy $t_1 \otimes t_2$ is included in σ for every play t_1 in σ_A reaching x_A, and for every play t_2 in σ_B reaching x_B. Indeed, by definition, $t_1 = u_A$ and $t_2 = v_B$ for some plays u and v of σ. From this follows that every play in $t_1 \otimes t_2$ is an element of prefix(σ). This property is established by induction (on the length of paths) using our early hypothesis that the strategy σ satisfies the *forward compatibility preservation* formulated in Section 5. Now, the strategy σ is positional, and $x_1 \otimes x_2$ is an acceptance position of the strategy. Hence, every such play in $t_1 \otimes t_2$ is element of the strategy σ.

Suppose that $x = x_A \otimes x_B$ is an acceptance position of σ. We have just shown that every play s interleaving a play s_A reaching x_A in σ_A, and a play s_B reaching x_B in σ_B, is an element of the strategy σ. Besides, it is not difficult to show that the strategies σ_A and σ_B are ingenuous, and to deduce from (18) that they satisfy

$$\sigma_A \perp A_O \quad \text{and} \quad \sigma_B \perp B_O.$$

At this stage, we may proceed with the second part of the proof and show that

$$\sigma \perp \tau.$$

We will deduce this property from the statement that

$$\sigma_A \otimes \sigma_B \perp \tau. \tag{19}$$

Suppose indeed that statement (19) holds, and that $x_A \otimes x_B \in X \cap Y$ is an interaction position shared by the two strategies $\sigma = (\sigma, X)$ and $\tau = (\tau, Y)$. By definition of $\sigma_A = (\sigma_A, X_A)$ and of $\sigma_B = (\sigma_B, X_B)$, the position $x_A \otimes x_B$ is an interaction position of $\sigma_A \otimes \sigma_B$. We deduce from our current hypothesis (19) that there exists a play $s : * \twoheadrightarrow x_A \otimes x_B$ element of the two strategies $\sigma_A \otimes \sigma_B$ and τ. By our previous discussion, we deduce from the fact that $x_A \otimes x_B \in \sigma$ that s is also an element of the strategy σ. This concludes our proof that statement (19) implies that σ and τ are orthogonal.

We start now the third part of the proof, which consists in a series of purely algebraic manipulations, establishing statement (19). By definition of the set $(A \otimes B)_P$ of scheduling tests, the strategy τ is scheduled precisely when

$$A_P \otimes B_P \perp \tau.$$

So, we know that for every Proponent scheduling test $\mathscr{S}_A \in A_P$ and every Proponent scheduling test $\mathscr{S}_B \in B_P$, the statement

$$\mathscr{S}_A \otimes \mathscr{S}_B \perp \tau$$

holds. This statement is equivalent to

$$\mathscr{S}_A \perp \tau \bullet \mathscr{S}_B$$

by Lemma 3, and holds for every Proponent scheduling $\mathscr{S}_A$ of the game A. The game A is scheduled. From this follows that

$$\sigma_A \perp \tau \bullet \mathscr{S}_B.$$

By applying Lemma 3 again, this is equivalent to

$$\sigma_A \otimes \mathscr{S}_B \perp \tau.$$

This last statement implies

$$\mathscr{S}_B \perp \sigma_A \bullet \tau$$

by Lemma 2. This holds for every Opponent scheduling test $\mathscr{S}_B$ of the game B. The game B is scheduled. From this follows that

$$\sigma_B \perp \sigma_A \bullet \tau.$$

The symmetric statement

$$\sigma_A \perp \tau \bullet \sigma_B$$

is established in exactly the same way. By Lemma 2, the conjunction of the two last statements is equivalent to

$$\sigma_A \otimes \sigma_B \perp \tau.$$

This concludes our proof that the tensor product $A \otimes B$ of two scheduled games satisfies the first (and most important) property required by the definition of scheduled games. The two other properties of scheduled games are then easily verified on the game $A \otimes B$.

□

Scheduled strategies. This leads us to the following notion of scheduled strategy.

Definition 12 (scheduled strategy) *An ingenuous strategy σ is called* scheduled *in a game (A, A_P, A_O) when it is orthogonal to every Opponent scheduling of the game, in the sense that $\sigma \perp A_O$.*

Note, in particular, that every Proponent scheduling test $\mathscr{S}$ of a scheduled game A is scheduled. A nice observation is that the interaction positions of a scheduled strategy (σ, X) coincide with its acceptance positions.

Property 3 *Every scheduled strategy (σ, X) satisfies $X = \sigma^\bullet$.*

This demonstrates that the notion of interaction position is only a technical device required for our proof that scheduled games are closed under tensor products. The notion may be removed from the very definition of scheduled games and scheduled strategies. It will not be mentioned anymore.

The orthogonality criterion in the definition of scheduled strategies requires that a scheduled strategy $\sigma : A$ interacts properly (without deadlocking) with all the Opponent scheduling tests of the game A. The fundamental property of this class of scheduling tests, is that they are sufficient to ensure that the scheduled strategy σ will interact properly with *every* scheduled counter-strategy of A. This is summarized by the following property, which follows immediately from the definition of a scheduled game:

Property 4 *Suppose that $\sigma : A$ is a scheduled strategy. Then, for every scheduled counter-strategy τ in the game A, the strategies σ and τ are orthogonal.*

A category of scheduled strategies. The category $\mathcal{I}$ has scheduled games as objects, and scheduled strategies σ of $A \multimap B$ as morphisms from (A, A_P, A_O) to (B, B_P, B_O). The identity on a scheduled game A is the identity strategy id_A in the category $\mathcal{G}$. The second defining clause of scheduled game ensures that it defines a scheduled strategy. We show below that the composite of two scheduled strategies is actually a scheduled strategy. This property is sufficient to ensure that $\mathcal{I}$ defines a category.

Property 5 *If $\sigma : A \multimap B$ and $\tau : B \multimap C$ are two scheduled strategies then the composite strategy $\sigma;\tau : A \multimap C$ is also scheduled.*

Proof. The strategy σ is scheduled. Property 4 ensures that $\sigma \perp \mathscr{S}_A \otimes \mathscr{S}_B$ for every Proponent scheduling test $\mathscr{S}_A$ of A and Opponent scheduling test $\mathscr{S}_B$. Statement $\mathscr{S}_A;\sigma \perp \mathscr{S}_B$ follows from Lemma 3. We deduce that the strategy $\mathscr{S}_A;\sigma : B$ is scheduled. We prove symmetrically that the strategy $\tau;\mathscr{S}_C : B^*$ is scheduled for every Opponent scheduling test $\mathscr{S}_C$ of C. From all this and Property 4, we deduce that

$$\mathscr{S}_A;\sigma \perp \tau;\mathscr{S}_C$$

From Lemma 3, we deduce that $(\mathscr{S}_A;\sigma) \otimes \mathscr{S}_C \perp \tau$, and from Lemma 2 that

$$\mathscr{S}_A;\sigma;\tau \perp \mathscr{S}_C.$$

We establish similarly that $\mathscr{S}_A \perp \sigma;\tau;\mathscr{S}_C$. By Lemma 2, the conjunction of the two statements implies the orthogonality relation

$$\sigma;\tau \perp \mathscr{S}_A \otimes \mathscr{S}_C.$$

This statement holds for every Proponent scheduling test $\mathscr{S}_A$ of A and every Opponent scheduling test $\mathscr{S}_C$ of C. This establishes that the composite strategy $\sigma;\tau : A \multimap C$ is orthogonal to every Opponent scheduling test $\mathscr{S}_A \otimes \mathscr{S}_C$ of $A \multimap C$ and is thus scheduled. $\square$

The operation $(-)^\bullet$ from the category $\mathcal{I}$ to the category of relations, which to every game associates its set of positions and to every strategy σ associates its set $\sigma^\bullet$ of acceptance positions, defines a functor, this rectifying the lax functor of Section 8. This functoriality property extends the programme of timeless games of [3] and the results of [23] to a non-alternating setting.

Theorem 1 *For every pair of scheduled strategies $\sigma : A \to B$ and $\tau : B \to C$ in the category $\mathcal{I}$, we have*

$$\sigma^\bullet;\tau^\bullet \quad = \quad (\sigma;\tau)^\bullet$$

Proof. The functor is lax: $\sigma^\bullet;\tau^\bullet \supseteq (\sigma;\tau)^\bullet$ for the same reasons as in Section 8. We need to show the converse inclusion

$$\sigma^\bullet;\tau^\bullet \quad \subseteq \quad (\sigma;\tau)^\bullet.$$

Suppose that $x \multimap z$ is a position of $\sigma^\bullet;\tau^\bullet$. By definition of relational composition, there exists a position y such that $x \multimap y$ is an acceptance position of σ and $y \multimap z$ is an acceptance position of τ. By the third condition in the definition of scheduled games, there exist a Proponent scheduling test $\mathscr{S}_A$ of A accepting the position x and an Opponent scheduling test $\mathscr{S}_C$ of C accepting the position z. Scheduling tests being scheduled strategies, the strategy $\mathscr{S}_A;\sigma$ in B and the counter-strategy $\tau;\mathscr{S}_C$ in B are scheduled strategies. They are orthogonal, by Property 4. Moreover, those strategies admit y as a common acceptance position. From this, we deduce that there exists a play $u : * \twoheadrightarrow y$ which is in both of those strategies. By Lemma 1, the strategies $\mathscr{S}_A$ and $\sigma;\tau;\mathscr{S}_C$ have x as only common acceptance position and similarly, the strategies $\mathscr{S}_A;\sigma;\tau$ and $\mathscr{S}_C$ have z as only common acceptance position. From this, we can finally conclude that there exist a play $s : * \twoheadrightarrow x \multimap y$ in σ and a play $t : * \twoheadrightarrow y \multimap z$ in τ such that $s_B = u = t_B$ whose interaction witnesses the fact that the position $x \multimap z$ is accepted by the strategy $\sigma;\tau$. $\square$

The scheduling tests therefore correct the mismatch described in Section 8:

Corollary 1 *The transformation which to every scheduled courteous ingenuous strategy $\sigma : A$ associates the closure operator*

$$y \quad \mapsto \quad \bigwedge\{\, y \in D_A \quad | \quad x \leq y \,\}$$

on the lattice D_A, obtained by completing the set of positions of A with a top element, extends to a functor from the subcategory of $\mathcal{I}$ of scheduled courteous ingenuous strategies to the category of concurrent games.

We have moreover described in Section 7 how to recover an asynchronous strategy from a closure operator in the image of the functor, in a bijective way.

The category $\mathcal{I}$ is $*$-autonomous, and thus provides a model of multiplicative linear logic. This fragment of logic may be extended with two modalities $\uparrow$ and $\downarrow$ lifting a game with an Opponent and a Proponent move, respectively. We are currently investigating possible extensions to the additive and exponential fragments, possibly using the group-theoretic reformulation of uniformity in Abramsky-Jagadeesan-Malacaria games, developed in [22]. The construction of scheduled games and strategies seem to be closely related to the *double glueing* construction investigated by Hyland and Schalk in [17]. We shall investigate the precise link with this abstract construction in future works.

We believe that the coincidence in the category $\mathcal{I}$ between dynamic composition (of strategies) and static composition (of relations) – expressed by Theorem 1 – is a key property towards the definition of a non-alternating generalization of the notion of innocence for strategies. However, our purely interactive notion of innocent strategy is too liberal in the sense that some strategies are not definable by proofs. The reason is that the scheduling criterion tests only for *directed* cycles in proof-structures, instead of the usual non-directed cycles of Danos and Regnier. On the other hand, it should be noted that the directed acyclicity criterion coincides with the usual non-directed acyclicity criterion in the usual situation of games semantics, treated for instance in [1]. In that case, the formula is purely multiplicative (i.e. it contains no lifting modality), every variable X and $X^\perp$ is interpreted as a game with a Proponent and an Opponent move, and every axiom link is interpreted as a "bidirectional" copycat strategy. In that case, innocent strategies coincide with proofs. Interestingly, it should be noted that the full completeness result in [2] uses a similar directed acyclicity criterion for MALL. Hence, our work provides a dynamic and interactive definition of directed acyclicity, and demonstrates that it is a fundamental, although somewhat hidden, concept of game semantics.

11 Conclusion

Extending the framework of asynchronous games to non-alternating strategies requires an exploration of the fine-grained structure of causality, using classical concepts of concurrency theory like the cube property. Interestingly, it appears that enforcing good causality properties on strategies is not sufficient to combine game semantics and concurrency theory in a harmonious way. One needs logical principles as well. Indeed, we uncover a subtle and unexpected mismatch between composition performed in asynchronous games and composition performed in concurrent games. The mismatch is resolved by strengthening the purely causal notion of *ingenuous strategy* into the more contextual notion of *scheduled strategy* by imposing a *scheduling criterion.* This criterion reformulates in a purely interactive and diagrammatic fashion a directed variant of the usual acyclicity criterion of linear logic. The criterion is sufficient to ensure the existence of a monoidal functor from the category of asynchronous games to the category of concurrent games. This functor projects the "small-step" interpretation of proofs (as strategies) to their "big-step" interpretation (as closure operators), or equivalently to their static interpretation (as relations).

Acknowledgments. We would like to thank Martin Hyland together with Pierre-Louis Curien, Claudia Faggian, Russ Harmer, Daniele Varacca, and Nobuko Yoshida for spontaneous and lively blackboard discussions.

References

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

[2] S. Abramsky and P.-A. Melliès. Concurrent games and full completeness. In *LICS*, volume 99, pages 431–442, 1999.

[3] P. Baillot, V. Danos, T. Ehrhard, and L. Regnier. Timeless Games. In *CSL*, pages 56–77, 1997.

[4] M.A. Bednarczyk. *Categories of asynchronous systems*. PhD thesis, University of Sussex, 1988.

[5] G. Berry. Modèles complètement adéquats et stables des lambda-calculs typés. *Thèse de Doctorat d'État, Université Paris VII*, 1979.

[6] F. Bracho, M. Droste, and D. Kuske. Representation of computations in concurrent automata by dependence orders. *Theoretical Computer Science*, 174(1-2):67–96, 1997.

[7] P.L. Curien and C. Faggian. L-Nets, Strategies and Proof-Nets. In *CSL*, pages 167–183. Springer, 2005.

[8] V. Danos and L. Regnier. The structure of multiplicatives. *Archive for Mathematical Logic*, 28(3):181–203, 1989.

[9] M. Droste and D. Kuske. Automata with concurrency relations – a survey. *Advances in Logic, Artificial Intelligence and Robotics*, pages 152–172, 2002.

[10] D.R. Ghica and A.S. Murawski. Angelic Semantics of Fine-Grained Concurrency. In *FoSSaCS*, pages 211–225, 2004.

[11] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[12] G. Gonthier, J.-J. Lévy, and P.-A. Melliès. An abstract standardisation theorem. In *Proc. of the 8th Annual Symposium on Logic in Computer Science*, pages 72–81, 1992.

[13] É. Goubault. Geometry and Concurrency: A User's Guide. *Mathematical Structures in Computer Science*, 10(4):411–425, 2000.

[14] J.M.E. Hyland and C.H.L. Ong. Fair games and full completeness for multiplicative linear logic without the mix-rule. *preprint*, 190, 1993.

[15] M. Hyland and L. Ong. On Full Abstraction for PCF: I, II and III. *Information and Computation*, 163(2):285–408, December 2000.

[16] M. Hyland and A. Schalk. Games on Graphs and Sequentially Realizable Functionals. *Proceedings of the 17th IEEE Symposium on Logic in Computer Science*, pages 257–264, 2002.

[17] M. Hyland and A. Schalk. Glueing and orthogonality for models of linear logic. *Theoretical Computer Science*, 294(1-2):183–231, 2003.

[18] A. Joyal. Remarques sur la théorie des jeux à deux personnes. *Gazette des Sciences Mathématiques du Quebec*, 1(4):46–52, 1977.

[19] J. Laird. A game semantics of the asynchronous π-calculus. *Proceedings of 16th CONCUR*, pages 51–65, 2005.

[20] A.W. Mazurkiewicz. Basic notions of trace theory. *Lecture Notes In Computer Science; Vol. 354*, pages 285–363, 1988.

[21] P.-A. Melliès. Axiomatic rewriting 4: A stability theorem in rewriting theory. *Logic in Computer Science*, pages 287–298, 1998.

[22] P.-A. Melliès. Asynchronous games 1: Uniformity by group invariance. 2003.

[23] P.-A. Melliès. Asynchronous games 2: the true concurrency of innocence. In *Proceedings of the 15th CONCUR*, number 3170 in LNCS, pages 448–465. Springer Verlag, 2004.

[24] P.-A. Melliès. Axiomatic Rewriting 1: A diagrammatic standardization theorem. *Lecture Notes in Computer Science*, 3838:554–638, 2005.

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

[26] S. Mimram. *Sémantique des jeux asynchrones et réécriture 2-dimensionnelle.* PhD thesis, PPS, CNRS – Université Paris Diderot, 2008.

[27] H. Nickau. Hereditarily sequential functionals. In A. Nerode and Yu. V. Matiyasevich, editors, *Proceedings of the Symposium on Logical Foundations of Computer Science: Logic at St. Petersburg*, volume 813 of *Lecture Notes in Computer Science*, pages 253–264. Springer Verlag, 1994.

[28] M. Nielsen, G. Plotkin, and G. Winskel. Petri Nets, Event Structures and Domains, Part I. *Theoretical Computer Science*, 13:85–108, 1981.

[29] P. Panangaden, V. Shanbhogue, and E.W. Stark. Stability and Sequentiality in Dataflow Networks. In *International Conference on Automates, Languages and Programming*, volume 443 of *LNCS*, pages 253–264. Springer Verlag, 1990.

[30] V. Sassone, M. Nielsen, and G. Winskel. Models for concurrency: Towards a classification. *Theoretical Computer Science*, 170(1):297–348, 1996.

[31] P. Selinger. First-order axioms for asynchrony. In *CONCUR '97*, number 1243 in Lecture Notes in Computer Science, pages 376–390. Springer, 1997.

[32] M.W. Shields. Concurrent Machines. *The Computer Journal*, 28(5):449–465, 1985.

[33] D. Varacca and N. Yoshida. Typed Event Structures and the π-calculus. MFPS, pages 373–398. Elsevier, 2006.

[34] G. Winskel and M. Nielsen. Models for concurrency. In *Handbook of Logic in Computer Science*, volume 3, pages 1–148. Oxford University Press, 1995.