Focusing in Asynchronous Games

Samuel Mimram

Focusing in Asynchronous Games

Samuel Mimram[*]

CEA LIST / École Polytechnique

Abstract. Game semantics provides an interactive point of view on proofs, which enables one to describe precisely their dynamical behavior during cut elimination, by considering formulas as games on which proofs induce strategies. We are specifically interested here in relating two such semantics of linear logic, of very different flavor, which both take in account concurrent features of the proofs: asynchronous games and concurrent games. Interestingly, we show that associating a concurrent strategy to an asynchronous strategy can be seen as a semantical counterpart of the focusing property of linear logic.

A cut-free proof in sequent calculus, when read from bottom up, progressively introduces the connectives of the formula that it proves, in the order specified by the syntactic tree constituting the formula, following the conventions induced by the logical rules. In this sense, a formula can be considered as a playground that the proof will explore. The formula describes the rules that this exploration should obey, it can thus be abstractly considered as a *game*, whose moves are its connectives, and a proof as a *strategy* to play on this game. If we follow the principle given by the Curry-Howard correspondence, and see a proof as some sort of program, this way of considering proof theory is particularly interesting because the strategies induced by proofs describe very precisely the interactive behavior of the corresponding program in front of its environment.

This point of view is at the heart of *game semantics* and has proved to be very successful in order to provide denotational semantics which is able to describe precisely the dynamics of proofs and programs. In this interactive perspective, two players are involved: the *Proponent*, which represents the proof, and the *Opponent*, which represents its environment. A formula induces a game which is to be played by the two players, consisting of a set of moves together with the rules of the game, which are formalized by the polarity of the moves (the player that should play a move) and the order in which the moves should be played. The interaction between the two players is formalized by a *play*, which is a sequence of moves corresponding to the part of the formula being explored during the cut-elimination of the proof with another proof. A proof is thus described in this setting by a *strategy* which corresponds to the set of interactions that the proof is willing to have with its environment.

This approach has been fruitful for modeling a wide variety of logics and programming languages. By refining Joyal's category of Conway games [12] and

Blass' games [7], Abramsky and Jagadeesan were able to give the first fully complete game model of the multiplicative fragment of linear logic extended with the MIX rule [3], which was later refined into a fully abstract model of PCF (Programming Language of Computable Functions) [4]. Here, "fully complete" and "fully abstract" essentially mean that the model is very precise, in the sense that every strategy is *definable* (i.e. is the interpretation of a proof or a program); more details can be found in Curien's survey on the subject [9]. Giving such a precise model of this language, introduced by Plotkin [18], was considered as a corner stone in computer science because it is a prototypical programming language, consisting of the λ-calculus extended with base data types and a fixpoint operator. At exactly the same time, Hyland and Ong gave another fully abstract model of PCF based on a variant of game semantics called *pointer games* [11]. In this model, definable strategies are characterized by two conditions imposed on strategies (well-bracketing and innocence). This setting was shown to be extremely expressive: relaxing in various ways these conditions gave rise to fully abstract models of a wide variety of programming languages with diverse features such as references, control, etc.

Game semantics is thus helpful to understand how logic and typing regulate computational processes. It also provides ways to analyze them (for example by doing model checking [2]) or to properly extend them with new features [9], and this methodology should be helpful to understand better concurrent programs. Namely, concurrency theory being relatively recent, there is no consensus about what a good process calculus should be (there are dozens of variants of the π-calculus and only one λ-calculus) and what a good typing system for process calculus should be: we believe that the study of denotational semantics of those languages is necessary in order to reveal their fundamental structures, with a view to possibly extending the Curry-Howard correspondence to programming languages with concurrent features. A few game models of concurrent programming languages have been constructed and studied. In particular, Ghica and Murawski have built a fully abstract model of Idealized Algol (an imperative programming language with references) extended with parallel composition and mutexes [10] and Laird a game semantics of a typed asynchronous π-calculus [13].

In this paper, we take a more logical point of view and are specifically interested in concurrent denotational models of linear logic. The idea that multiplicative connectives express concurrent behaviors is present since the beginnings of linear logic: it is namely very natural to see a proof of $A \,\invamp\, B$ or $A \otimes B$ as a proof of A in "parallel" with a proof of B, the corresponding introduction rules being

$$\frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \,\invamp\, B}[\invamp] \qquad \text{and} \qquad \frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B}[\otimes]$$

with the additional restriction that the two proofs should be "independent" in the case of the tensor, since the corresponding derivations in premise of the rule are two disjoint subproofs. Linear logic is inherently even more parallel: it has the *focusing* property [6] which implies that every proof can be reorganized into one in which all the connectives of the same polarity at the root of a formula are introduced at once (this is sometimes also formulated using *synthetic con-*

nectives). This property, originally discovered in order to ease proof-search has later on revealed to be fundamental in semantics and type theory. Two game models of linear logic have been developed in order to capture this intuition. The first, by Abramsky and Melliès, called *concurrent games*, models strategies as closure operators [5] following the domain-theoretic principle that computations add information to the current state of the program (by playing moves). It can be considered as a big-step semantics because concurrency is modeled by the ability that strategies have to play multiple moves at once. The other one is the model of *asynchronous games* introduced by Melliès [14] where, in the spirit of "true concurrency", playing moves in parallel is modeled by the possibility for strategies to play any interleaving of those moves and these interleavings are considered to be equivalent. We recall here these two models and explain here that concurrent games can be related to asynchronous games using a semantical counterpart of focusing. A detailed presentation of these models together with the proofs of many properties evoked in this paper can be found in [16,17].

1 Asynchronous games

Recall that a *graph* $G = (V, E, s, t)$ consists of a set V of vertices (or *positions*), a set E of edges (or *transitions*) and two functions $s, t : E \rightarrow V$ which to every transition associate a position which is called respectively its *source* and its *target*. We write $m : x \longrightarrow y$ to indicate that m is a transition with x as source and y as target. A *path* is a sequence of consecutive transitions and we write $t : x \longrightarrow\!\!\!\!\rightarrow y$ to indicate that t is a path whose source is x and target is y. The concatenation of two consecutive paths $s : x \longrightarrow\!\!\!\!\rightarrow y$ and $t : y \longrightarrow\!\!\!\!\rightarrow z$ is denoted $s \cdot t$. An *asynchronous graph* $G = (G, \diamond)$ is a graph G together with a *tiling* relation $\diamond$, which relates paths of length two with the same source and the same target. If $m : x \longrightarrow y_1$, $n : x \longrightarrow y_2$, $p : y_1 \longrightarrow z$ and $q : y_2 \longrightarrow z$ are four transitions, we write

$$
\begin{array}{ccc}
 & z & \\
{\scriptstyle p}\nearrow & & \nwarrow{\scriptstyle q} \\
y_1 & \sim & y_2 \\
{\scriptstyle m}\nwarrow & & \nearrow{\scriptstyle n} \\
 & x &
\end{array}
\tag{1}
$$

to diagrammatically indicate that $m \cdot p \diamond n \cdot q$. We write $\sim$ for the smallest congruence (wrt concatenation) containing the tiling relation. This relation is called *homotopy* because it should be thought as the possibility, when s and t are two homotopic paths, to deform "continuously" the path s into t. From the concurrency point of view, a homotopy between two paths indicates that these paths are the same up to reordering of independent events, as in Mazurkiewicz traces. In the diagram (1), the transition q is the *residual* (in the sense of rewriting theory) of the transition m after the transition n, and similarly p is the residual of n after m; the *event* (also called *move*) associated to a transition is therefore its equivalence class under the relation identifying a transition with its residuals. In the asynchronous graphs we consider, we suppose that given a path $m \cdot p$ there is at most one path $n \cdot q$ forming a tile (1). We moreover require that a transition should have at most one residual after another transition.

We consider formulas of the multiplicative and additive fragment of linear logic (MALL), which are generated by the grammar

$$A \quad ::= \quad A \,⅋\, A \quad | \quad A \otimes A \quad | \quad A \,\&\, A \quad | \quad A \oplus A \quad | \quad X \quad | \quad X^*$$

where X is a variable (for brevity, we don't consider units). The $⅋$ and $\&$ (resp. $\otimes$ and $\oplus$) connectives are sometimes called *negative* or *asynchronous* (resp. *positive* or *synchronous*). A *position* is a term generated by the following grammar

$$x \quad ::= \quad † \quad | \quad x \,⅋\, x \quad | \quad x \otimes x \quad | \quad \&_L\, x \quad | \quad \&_R\, x \quad | \quad \oplus_L\, x \quad | \quad \oplus_R\, x$$

The de Morgan dual A^* of a formula is defined as usual, for example $(A \otimes B)^* = A^* \,⅋\, B^*$, and the dual of a position is defined similarly. Given a formula A, we write $\mathrm{pos}(A)$ for the set of valid positions of the formula which are defined inductively by $† \in \mathrm{pos}(A)$ and if $x \in \mathrm{pos}(A)$ and $y \in \mathrm{pos}(B)$ then $x \,⅋\, y \in \mathrm{pos}(A \,⅋\, B)$, $x \otimes y \in \mathrm{pos}(A \otimes B)$, $\&_L x \in \mathrm{pos}(A \,\&\, B)$, $\&_R y \in \mathrm{pos}(A \,\&\, B)$, $\oplus_L x \in \mathrm{pos}(A \oplus B)$ and $\oplus_R y \in \mathrm{pos}(A \oplus B)$.

An *asynchronous game* $G = (G, *, \lambda)$ is an asynchronous graph $G = (V, E, s, t)$ together with a distinguished *initial position* $* \in V$ and a function $\lambda : E \to \{O, P\}$ which to every transition associates a *polarity*: either O for Opponent or P for Proponent. A transition is supposed to have the same polarity as its residuals, polarity is therefore well-defined on moves. We also suppose that every position x is *reachable* from the initial position, i.e. that there exists a path $* \longrightarrow\!\!\!\!\to x$. Given a game G, we write G^* for the game G with polarities inverted. Given two games G and H, we define their asynchronous product $G\|H$ as the game whose positions are $V_{G\|H} = V_G \times V_H$, whose transitions are $E_{G\|H} = E_G \times V_H + V_G \times E_H$ (by abuse of language we say that a transition is "in G" when it is in the first component of the sum or "in H" otherwise) with the source of $(m, x) \in E_G \times V_H$ being $(s_G(m), x)$ and its target being $(t_G(m), x)$, and similarly for transitions in $V_G \times E_H$, two transitions are related by a tile whenever they are all in G (resp. in H) and the corresponding transitions in G (resp. in H) are related by a tile or when two of them are an instance of a transition in G and the two other are instances of a transition in H, the initial position is $(*_G, *_H)$ and the polarities of transitions are those induced by G and H.

To every formula A, we associate an asynchronous game G_A whose vertices are the positions of A as follows. We suppose fixed the interpretation of the free variables of A. The game $G_{A⅋B}$ is obtained from the game $G_A\|G_B$ by replacing every pair of positions (x, y) by $x \,⅋\, y$, and adding a position $†$ and an Opponent transition $† \longrightarrow † \,⅋\, †$. The game $G_{A\&B}$ is obtained from the disjoint union $G_A + G_B$ by replacing every position x of G_A (resp. G_B) by $\&_L x$ (resp. $\&_R x$), and adding a position $†$ and two Opponent transitions $† \longrightarrow \&_L†$ and $† \longrightarrow \&_R†$. The games associated to the other formulas are deduced by de Morgan duality: $G_{A^*} = G_A^*$. This operation is very similar to the natural embedding of event structures into asynchronous transition systems [19]. For example, if we interpret the variable X (resp. Y) as the game with two positions $†$ and x (resp. $†$ and y) and one transition between them, the interpretation of the formula $(X \otimes X^*) \,\&\, Y$ is depicted in (2). We have made explicit the positions of the games in order to underline the fact that they correspond to partial explorations of formulas, but

the naming of a position won't play any role in the definition of asynchronous strategies.

$$
\begin{array}{ccc}
& \&_L(x \otimes x^*) & \\
\nearrow & & \nwarrow \\
\&_L(x \otimes \dagger) \quad\quad \sim \quad\quad & & \&_L(\dagger \otimes x^*) \\
\searrow & & \nearrow \\
& \&_L(\dagger \otimes \dagger) \quad\quad\quad\quad \&_R y \\
& \uparrow \quad\quad\quad\quad\quad\quad \uparrow \\
& \&_L\dagger \quad\quad\quad\quad\quad \&_R\dagger \\
& \nwarrow \quad\quad\quad \nearrow \\
& \dagger &
\end{array}
\tag{2}
$$

A *strategy* σ on a game G is a prefix-closed set of *plays*, which are paths whose source is the initial position of the game. To every proof π of a formula A, we associate a strategy, defined inductively on the structure of the proof. Intuitively, these plays are the explorations of formulas allowed by the proof. For example, the strategies interpreting the proofs

$$
\dfrac{\dfrac{\pi}{\vdash \Gamma, A, B}}{\vdash \Gamma, A \, \mathfrak{N} \, B}[\mathfrak{N}] \quad\quad \text{and} \quad\quad \dfrac{\dfrac{\pi}{\vdash \Gamma, A}}{\vdash \Gamma, A \oplus B}[\oplus_L]
$$

will contain plays which are either empty or start with a transition $\dagger \longrightarrow \dagger \, \mathfrak{N} \, \dagger$ (resp. $\dagger \longrightarrow \oplus_L \dagger$) followed by a play in the strategy interpreting π. The other rules are interpreted in a similar way. To be more precise, since the interpretation of a proof depends on the interpretation of its free variables, the interpretation of a proof will be an uniform family of strategies indexed by the interpretation of the free variables in the formula (as in e.g. [3]) and axioms proving $\vdash A, A^*$ will be interpreted by copy-cat strategies on the game interpreting A. For the lack of space, we will omit details about variables and axioms.

Properties characterizing definable strategies were studied in the case of alternating strategies (where Opponent and Proponent moves should alternate strictly in plays) in [15] and generalized to the non-alternating setting that we are considering here in [16,17]. We recall here the basic properties of definable strategies. One of the interest of these is that they allow one to reason about strategies in a purely local and diagrammatic fashion. It can be shown that every definable strategy σ is

- *positional*: for every three paths $s, t : * \longrightarrow x$ and $u : x \longrightarrow y$, if $s \cdot u \in \sigma$, $s \sim t$ and $t \in \sigma$ then $t \cdot u \in \sigma$. This property essentially means that a strategy is a subgraph of the game: a strategy σ induces a subgraph G_σ of the game which consists of all the positions and transitions contained in at least one play in σ, conversely every play in this subgraph belongs to the strategy when the strategy is positional. In fact, this graph G_σ may be seen itself as an asynchronous graph by equipping it with the tiling relation induced by the underlying game.
- *deterministic*: if the graph G_σ of the strategy contains a transition $n : x \longrightarrow y_2$ and a Proponent transition $m : x \longrightarrow y_1$ then it also contains the residual of m along n, this defining a tile of the form (1).
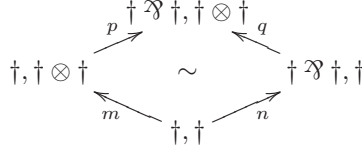
– *receptive*: if σ contains a play $s : * \longrightarrow\!\!\!\!\!\rightarrow x$ and there exists an Opponent move $m : x \longrightarrow y$ in the game then the play $s \cdot m : * \longrightarrow\!\!\!\!\!\rightarrow y$ is also in σ.

– *total*: if σ contains a play $s : * \longrightarrow\!\!\!\!\!\rightarrow x$ and there is no Opponent transition $m : x \longrightarrow y$ in the game then either the position x is terminal (there is no transition with x as source in the game) or there exists a Proponent transition $m : x \longrightarrow y$ such that $s \cdot m$ is also in σ.

2 Focusing in linear logic

In linear logic, a proof of the form depicted on the left-hand side of

$$
\cfrac{\cfrac{\cfrac{\pi_1}{\vdash A, B, C}}{\vdash A \,\mathregular{⅋}\, B, C}[\mathregular{⅋}] \qquad \cfrac{\pi_2}{\vdash D}}{\vdash A \,\mathregular{⅋}\, B, C \otimes D}[\otimes]
\qquad\qquad
\cfrac{\cfrac{\cfrac{\pi_1}{\vdash A, B, C} \qquad \cfrac{\pi_2}{\vdash D}}{\vdash A, B, C \otimes D}[\otimes]}{\vdash A \,\mathregular{⅋}\, B, C \otimes D}[\mathregular{⅋}]
$$

can always be reorganized into the proof depicted on the right-hand side. This proof transformation can be seen as "permuting" the introduction of $\otimes$ after the introduction of $\mathregular{⅋}$ (when looking at proofs bottom-up). From the point of view of the strategies associated to the proofs, the game corresponding to the proven sequent contains

and the transformation corresponds to replacing the path $m \cdot p$ by the path $n \cdot q$ in the strategy associated to the proof. More generally, the introduction rules of two negative connectives can always be permuted, as well as the introduction of two positive connectives, and the introduction rule of a positive connective can always be permuted after the introduction rule of a negative one. Informally, a negative (resp. positive) can always be "done earlier" (resp. "postponed"). We write $\pi \prec \pi'$ when a proof π' can be obtained from a proof π by a series of such permutations of rules.

These permutations of rules are at the heart of Andreoli's work [6] which reveals that if a formula is provable then it can be found using a *focusing* proof search, which satisfies the following discipline: if the sequent contains a negative formula then a negative formula should be decomposed (*negative phase*), otherwise a positive formula should be chosen and decomposed repeatedly until a (necessarily unique) formula is produced (*positive phase*) – this can be formalized using a variant of the usual sequent rules for linear logic. From the point of view of game semantics, this says informally that every strategy can be reorganized into one playing alternatively a "bunch" of Opponent moves and a "bunch" of Proponent moves.

All this suggests that proofs in sequent calculus are too sequential: they contain inessential information about the ordering of rules, and we would like to work with proofs modulo the congruence generated by the $\prec$ relation. Semantically, this can be expressed as follows. A strategy σ is *courteous* when for every

tile of the form (1) of the game, such that the path $m \cdot p$ is in (the graph G_σ of) the strategy σ, and either m is a Proponent transition or p is an Opponent transition, the path $n \cdot q$ is also in σ. We write $\tilde{\sigma}$ for the smallest courteous strategy containing σ. Courteous strategies are less sequential than usual strategies: suppose that σ is the strategy interpreting a proof π of a formula A, then a play s is in $\tilde{\sigma}$ if and only if it is a play in the strategy interpreting some proof π' such that $\pi \prec \pi'$.

Strategies which are positional, deterministic, receptive, total, courteous, are closed under residuals of transitions and satisfy some other properties such as the *cube property* (enforcing a variant of the domain-theoretic stability property) are called *ingenuous* and are very well behaved: they form a compact closed category, with games as objects and ingenuous strategies σ on $A^* \| B$ as morphisms $\sigma : A \to B$, which is a denotational model of MLL, which can be refined into a model of MALL by suitably quotienting morphisms. Composition of strategies $\sigma : A \to B$ and $\tau : B \to C$ is defined as usual in game semantics by "parallel composition and hiding": the plays in $\tau \circ \sigma$ are obtained from *interactions* of σ and τ, which are the plays on the game $A \| B \| C$ whose projection on $A^* \| B$ (resp. $B^* \| C$) is in σ (resp. τ) up to polarities of moves, by restricting them to $A^* \| C$. Associativity of the composition is not trivial and relies mainly on the determinism property, which implies that if a play in $\tau \circ \sigma$ comes from two different interactions s and t then there it also comes from a third interaction u which is greater than both wrt the prefix modulo homotopy order.

3 Concurrent games

We recall here briefly the model of concurrent games [5]. A *concurrent strategy* ς on a complete lattice $(D, \leqslant)$ is a continuous closure operator on this lattice. Recall that a closure operator is a function $\varsigma : D \to D$ which is

1. *increasing*: $\forall x \in D, x \leqslant \varsigma(x)$
2. *idempotent*: $\forall x \in D, \varsigma \circ \varsigma(x) = \varsigma(x)$
3. *monotone*: $\forall x, y \in D, x \leqslant y \Rightarrow \varsigma(x) \leqslant \varsigma(y)$

Such a function is *continuous* when it preserves joins of directed subsets. Informally, an order relation $x \leqslant y$ means that the position y contains more information than x. With this intuition in mind, the first property expresses the fact that playing a strategy increases the amount of information in the game, the second that a strategy gives all the information it can given its knowledge in the current position (so that if it is asked to play again it does not have anything to add), and the third that the more information the strategy has from the current position the more information it has to deliver when playing.

Every such concurrent strategy ς induces a set of fixpoints defined as the set $\mathrm{fix}(\varsigma) = \{ x \in D \mid \varsigma(x) = x \}$. This set is (M) closed under arbitrary meets and (J) closed under joins of directed subsets and conversely, every set $X \subseteq D$ of positions which satisfies these two properties (M) and (J) induces a concurrent strategy $X^\bullet$ defined by $X^\bullet(x) = \bigwedge \{ y \in X \mid x \leqslant y \}$, whose set of fixpoints is precisely X.
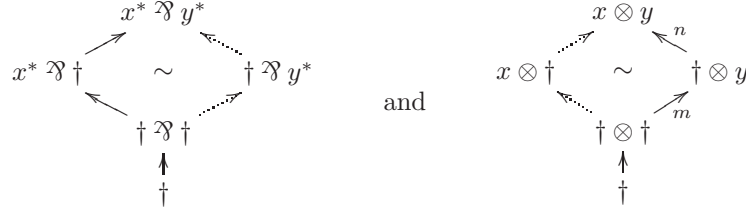
Suppose that G is a game. Without loss of generality, we can suppose that G is *simply connected*, meaning that every position is reachable from the initial

position $*$ and two plays $s, t : * \longrightarrow\!\!\!\!\rightarrow x$ with the same target are homotopic. This game induces a partial order on its set of positions, defined by $x \leqslant y$ iff there exists a path $x \longrightarrow\!\!\!\!\rightarrow y$, which can be completed into a complete lattice D by formally adding a top element $\top$. Now, consider a strategy σ on the game G. A position x of the graph G_σ induced by σ is *halting* when there is no Proponent move $m : x \longrightarrow y$ in σ: in such a position, the strategy is either done or is waiting for its Opponent to play. It can be shown that the set σ° of halting positions of an ingenuous strategy σ satisfies the properties (M) and (J) and thus induces a concurrent strategy $(\sigma^\circ)^\bullet$. Conversely, if for every positions $x, y \in D$ we write $x \leqslant_P y$ when $y \neq \top$ and there exists a path $x \longrightarrow\!\!\!\!\rightarrow y$ containing only Proponent moves, then every concurrent strategy ς induces a strategy $\varsigma^\natural$ defined as the set of plays in G whose intermediate positions x satisfy $x \leqslant_P \varsigma(x)$ – and these can be shown to be ingenuous. This thus establishes a precise relation between the two models:

Theorem 1. *The two operations above describe a retraction of the ingenuous strategies on a game G into the concurrent strategies on the domain D induced by the game G.*

Moreover, the concurrent strategies which correspond to ingenuous strategies can be characterized directly. In this sense, concurrent strategies are close to the intuition of focused proofs: given a position x, they play at once many Proponent moves in order to reach the position which is the image of x.

However, the correspondence described above is not functorial: it does not preserve composition. This comes essentially from the fact that the category of ingenuous strategies is compact closed, which means that it has the same strategies on the games interpreting the formulas $A \otimes B$ and $A \,⅋\, B$ (up to the polarity of the first move). For example, if X (resp. Y) is interpreted by the game with one Proponent transition $\dagger \longrightarrow x$ (resp. $\dagger \longrightarrow y$), the interpretations of $X^* \,⅋\, Y^*$ and $X \otimes Y$ are respectively

Now, consider the strategy $\sigma : X^* \,⅋\, Y^*$ which contains only the prefixes of the bold path $\dagger \longrightarrow\!\!\!\!\rightarrow (x^* \,⅋\, y^*)$ and the strategy $\tau : X \otimes Y$ which contains only the prefixes of bold path $\dagger \longrightarrow\!\!\!\!\rightarrow (x \otimes y)$. The fixpoints of the corresponding concurrent games are respectively $\sigma^\circ = \{\ \dagger,\ \dagger\,⅋\,\dagger,\ x^*\,⅋\,\dagger,\ x^*\,⅋\,y^*\ \}$ and $\tau^\circ = \{\ x \otimes y\ \}$. From the point of view of asynchronous strategies, the only reachable positions by both of the strategies in $X \otimes Y$ are $\dagger$ and $\dagger \otimes \dagger$. However, from the point of view of the associated concurrent strategies, they admit the position $x \otimes y$ as a common position in $X \otimes Y$. From this observation, it is easy to build two strategies $\sigma : A \to B$ and $\tau : B \to C$ such that $((\tau \circ \sigma)^\circ)^\bullet \neq (\tau^\circ)^\bullet \circ (\sigma^\circ)^\bullet$ (we refer the reader to [5] for the definition of the composition of closure operators). In the example

above, the strategy τ is the culprit: as mentioned in the introduction, the two strategies on X and Y should be independent in a proof of $X \otimes Y$, whereas here the strategy τ makes the move n depends on the move m. Formally, this dependence expresses the fact that the move m occurs after the move n in every play of the strategy τ. In [16], we have introduced a *scheduling criterion* which dynamically enforces this independence between the components of a tensor: a strategy satisfying this criterion is essentially a strategy such that in a sub-strategy on a formula of the form $A \otimes B$ no move of A depends on a move of B and vice versa. Every definable strategy satisfies this criterion and moreover,

Theorem 2. *Strategies satisfying the scheduling criterion form a subcategory of the category of ingenuous strategies and the operation $\sigma \mapsto (\sigma^\circ)^\bullet$ extends into a functor from this category to the category of concurrent games.*

This property enables us to recover more precisely the focusing property directly at the level of strategies as follows. Suppose that σ is an ingenuous strategy interpreting a proof π of a sequent $\vdash \Gamma$. Suppose moreover that $s : x \twoheadrightarrow y$ is a maximal play in σ. By receptivity and courtesy of the strategy, this play is homotopic in the graph G_σ to the concatenation of a path $s_1 : x \twoheadrightarrow x_1$ containing only Opponent moves, where x_1 is a position such that there exists no Opponent transition $m : x_1 \longrightarrow x_1'$, and a path $s_2 : x_1 \twoheadrightarrow y$. Similarly, by totality of the strategy, the path s_2 is homotopic to the concatenation of a path $s_2' : x_1 \twoheadrightarrow y_1$ containing only Proponent moves, where y_1 is a position which is either terminal or such that there exists an Opponent transition $m : y_1 \longrightarrow y_1'$, and a path $s_2'' : y_1 \twoheadrightarrow y$. The path s_2' consists in the partial exploration of positive formulas, one of them being explored until a negative subformula is reached. By courtesy of the strategy, Proponent moves permute in a strategy and we can suppose that s_2' consists only in such a maximal exploration of one of the formulas available at the position x. If at some point a branch of a tensor formula is explored, then by the scheduling criterion it must be able to also explore the other branch of the formula. By repeating this construction on the play s_2'', every play of σ can be transformed into one which alternatively explores all the negative formulas and explores one positive formula until negative formulas are reached. By formalizing further this reasoning, one can therefore show that

Theorem 3. *In every asynchronous strategy interpreting a proof in MALL is included a strategy interpreting a focusing proof of the same sequent.*

A motivation for introducing concurrent games was to solve the well-known *Blass problem* which reveals that the "obvious" game model for the multiplicative fragment of linear logic has a non-associative composition. Abramsky explains in [1] that there are two ways to solve this: either syntactically by considering a focused proof system or semantically by using concurrent games. Thanks to asynchronous games, we understand here the link between the two points of view: every proof in linear logic can be reorganized into a focused one, which is semantically understood as a strategy playing multiple moves of the same polarity at once, and is thus naturally described by a concurrent strategy. In

this sense, concurrent strategies can be seen as a semantical generalization of focusing, where the negative connectives are not necessarily all introduced at first during proof-search. It should be noted that some concurrent strategies are less sequential than focused ones, for example the strategies interpreting the multi-focused proofs [8], where the focus can be done on multiple positives formulas in the positive phase of the focused proof-search. Those multi-focused proofs were shown to provide canonical representatives of focused proofs (the interpretation of the canonical multi-focused proof can be recovered by generalizing Theorem 3). We are currently investigating a generalization of this result by finding canonical representatives for concurrent strategies.

The author is much indebted to Paul-André Melliès and Martin Hyland.

References

1. S. Abramsky. Sequentiality vs. concurrency in games and logic. *Mathematical Structures in Computer Science*, 13(04):531–565, 2003.
2. S. Abramsky, D.R. Ghica, A.S. Murawski, and L. Ong. Applying game semantics to compositional software modeling and verification. *LNCS*, pages 421–435, 2004.
3. S. Abramsky and R. Jagadeesan. Games and Full Completeness for Multiplicative Linear Logic. *The Journ. of Symb. Logic*, 59(2):543–574, 1994.
4. S. Abramsky, R. Jagadeesan, and P. Malacaria. Full abstraction for PCF. *Information and Computation*, 163(2):409–470, 2000.
5. S. Abramsky and P.-A. Melliès. Concurrent games and full completeness. In *LICS*, volume 99, pages 431–442, 1999.
6. J.M. Andreoli. Logic Programming with Focusing Proofs in Linear Logic. *Journal of Logic and Computation*, 2(3):297–347, 1992.
7. A. Blass. Degrees of indeterminacy of games. *Fund. Math*, 77(2):151–166, 1972.
8. K. Chaudhuri, D. Miller, and A. Saurin. Canonical Sequent Proofs via Multi-Focusing. In *International Conference on Theoretical Computer Science*, 2008.
9. P.L. Curien. Definability and Full Abstraction. *Electronic Notes in Theoretical Computer Science*, 172:301–310, 2007.
10. D.R. Ghica and A.S. Murawski. Angelic Semantics of Fine-Grained Concurrency. In *FoSSaCS*, pages 211–225, 2004.
11. M. Hyland and L. Ong. On Full Abstraction for PCF: I, II and III. *Information and Computation*, 163(2):285–408, December 2000.
12. A. Joyal. Remarques sur la théorie des jeux à deux personnes. *Gazette des Sciences Mathématiques du Quebec*, 1(4):46–52, 1977.
13. J. Laird. A game semantics of the asynchronous π-calculus. *Proceedings of 16th CONCUR*, pages 51–65, 2005.
14. P.-A. Melliès. Asynchronous games 2: the true concurrency of innocence. In *Proc. of the 15th CONCUR*, number 3170 in LNCS, pages 448–465, 2004.
15. P.-A. Melliès. Asynchronous games 4: a fully complete model of propositional linear logic. *Proceedings of 20th LICS*, pages 386–395, 2005.
16. P.-A. Melliès and S. Mimram. Asynchronous Games: Innocence without Alternation. In *Proc. of the 18th CONCUR*, volume 4703 of *LNCS*, pages 395–411, 2007.
17. S. Mimram. *Sémantique des jeux asynchrones et réécriture 2-dimensionnelle*. PhD thesis, PPS, CNRS–Univ. Paris Diderot, 2008.
18. G.D. Plotkin. LCF considered as a programming language. *Theoretical Computer Science*, 5(3):223–255, 1977.
19. G. Winskel and M. Nielsen. Models for concurrency. In *Handbook of Logic in Computer Science*, volume 3, pages 1–148. Oxford University Press, 1995.