

# Resource modalities in game semantics

Paul-André Melliès, Nicolas Tabareau

▶ **To cite this version:**

Paul-André Melliès, Nicolas Tabareau. Resource modalities in game semantics. 2007. hal-00144510

**HAL Id: hal-00144510**
**https://hal.science/hal-00144510**

Preprint submitted on 3 May 2007

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

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

# Resource modalities in game semantics

Paul-André Melliès Nicolas Tabareau *

hal-00144510, version 1 - 3 May 2007

## Abstract

*The description of resources in game semantics has never achieved the simplicity and precision of linear logic, because of a misleading conception: the belief that linear logic is more primitive than game semantics. We advocate the contrary here: that game semantics is conceptually more primitive than linear logic. Starting from this revised point of view, we design a categorical model of resources in game semantics, and construct an arena game model where the usual notion of bracketing is extended to multi-bracketing in order to capture various resource policies: linear, affine and exponential.*

## 1 Introduction

**Game semantics and linear logic.** Game semantics is the younger sibling of linear logic: born (or reborn) at the beginning of the 1990s, in the turmoil produced by the recent discovery of linear logic by Jean-Yves Girard [9], it remained under its spiritual influence for a very long time. This ascendancy of linear logic was extraordinarily healthy and profitable in the early days. Properly guided, game semantics developed steadily, following the idea that every *formula* of linear logic describes a *game*; and that every *proof* of the formula describes a *strategy* for playing on that game.

This correspondence between formulas of linear logic and games is supported by a series of elegant and striking analogies. One basic principle of linear logic is that every formula behaves as a resource, which disappears once consumed. In particular, a proof of the formula $A \multimap B$ is required to deduce the conclusion $B$ by using (or consuming) its hypothesis $A$ exactly once. This principle is nicely reflected in game semantics, by the idea that playing a game is just like consuming a resource, the game itself.

Another basic principle of linear logic is that negation $A \mapsto \neg A$ is involutive. This means that every formula $A$ is equal (or at least isomorphic) to the formula negated twice:

$$A \quad \cong \quad \neg\neg A. \tag{1}$$

Again, this principle is nicely reflected in game semantics by the idea that negating a game $A$ consists in permuting the rôles of the two players. Hence, negating a game twice amounts to permuting the rôle of Proponent and Opponent twice, which is just like doing nothing.

The connectives of linear logic are also nicely reflected in game semantics. For instance, the tensor product $A \otimes B$ of two formulas $A$ and $B$ is suitably interpreted as the game (or formula) $A$ played in parallel with the game (or formula) $B$, where only Opponent may switch from a component to the other one. Similarly, the sum $A \oplus B$ of two formulas $A$ and $B$ is suitably interpreted as the game where Proponent plays the first move, which consists in choosing between the game $A$ and the game $B$, before carrying on in the selected component. Finally, the exponential modality of linear logic $!A$ applied to the formula $A$ is suitably interpreted as the game where several copies of the game $A$ are played in parallel, and only Opponent is allowed (1) to switch from a copy to another one and (2) to open a fresh copy of the game $A$.

What we describe here is in essence the game semantics of linear logic defined by Andreas Blass in [6]. Simple and elegant, the model reflects the full flavour of the resource policy of linear logic. It is also remarkable that this game semantics is an early predecessor to linear logic [5].

**A schism with linear logic.** The destiny of game semantics has been to emancipate itself from linear logic in the mid–1990s, in order to comply with its own designs, inherited from denotational semantics:

1. the desire to interpret *programs* written in programming languages with effects (recursion, states, etc.) and to characterise exactly their interactive behaviour inside *fully abstract* models;
2. the desire to understand the algebraic principles of programming languages and effects, using the language of category theory.

So, a new generation of game semantics arose, propelled by (at least) two different lines of research:

*This work has been supported by the ANR Invariants algébriques des systèmes informatiques (INVAL). Postal address: Equipe PPS, Université Paris VII, 2 place Jussieu, Case 7014, 75251 Paris Cedex 05, FRANCE. Email addresses: **mellies@pps.jussieu.fr** and **tabareau@pps.jussieu.fr**

1. Samson Abramsky and Radha Jagadeesan [2] noticed that the (alternating variant of the) Blass model does not define a categorical model of linear logic. Worse: it does not even define a category, for lack of associativity. Abramsky dubs this phenomenon the *Blass problem* and describes it in [1].
2. Martin Hyland and Luke Ong [16] introduced the notion of *arena game*, and characterised the interactive behaviour of programs written in the functional language PCF — the simply-typed $\lambda$-calculus with conditional test, arithmetic and recursion.

So, the Blass problem indicates that it is difficult to construct a (sequential) game model of linear logic; and at about the same time, arena games become mainstream although they do not define a model of linear logic. These two reasons (at least) opened a schism between game semantics and linear logic: it suddenly became accepted that categories of (sequential) games and strategies would only capture *fragments* of linear logic (intuitionistic or polarised) but not the whole thing.

On the other hand, defining the resource modalities of linear logic for game semantics requires to reunify the two schismatic subjects. Since the disagreement started with category theory, this reunification should occur at the categorical level. We explain (in §2) how to achieve this by *relaxing* the involutive negation of linear logic into a less constrained tensorial negation. This negation induces in turn a *linear continuation* monad, whose unit

$$A \quad \longrightarrow \quad \neg\neg A \tag{2}$$

refines the isomorphism (1) of linear logic. Moving from an involutive to a tensorial negation means that we replace linear logic by a more general and primitive logic – which we call *tensorial logic*. As we will see, this shift to tensorial logic clarifies the Blass problem, and describes the structure of arena games. It also enables the expressions of resource modalities in game semantics, just as it is usually done in linear logic. However, because the presentation of modalities may appear difficult to readers not familiar with categorical semantics, we prefer to recall first the notion of *well-bracketing* in arena games — and explain how it can be reunderstood as a resource policy, and extended to multi-bracketing.

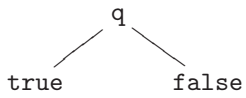
**Arena games.** Recall that an *arena* is defined as a forest of rooted trees, whose nodes are called the *moves* of the game. One writes

$$m \vdash n$$

and says that the move $m$ *enables* the move $n$ when the move $m$ is the immediate ancestor of the move $n$ in the arena. Every move $m$ is assigned a polarity $\lambda^{OP}(m) \in \{-1, +1\}$. By convention, $\lambda^{OP}(m) = +1$ when the move is Proponent, and $\lambda^{OP}(m) = -1$ when it is Opponent. Finally, one requires that the arena is alternating:

$$m \vdash n \quad \Longrightarrow \quad \lambda^{OP}(m) = -\lambda^{OP}(n)$$

and that all roots (called *opening* moves) of the arena have the same polarity. A typical example of arena is the boolean arena $\mathbb{B}$:

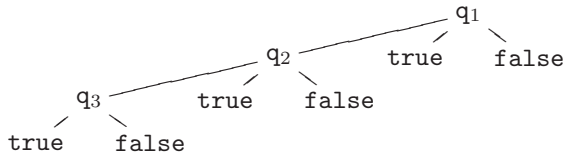



(3)

where the Opponent move `q` justifies the two Proponent moves `true` and `false`. Every arena game $A$ induces a set of *justified plays*, which are essentially sequences of moves (we will avoid discussing *pointers* here.) Typically, the PCF type

$$(\mathbb{B}_3 \Rightarrow \mathbb{B}_2) \Rightarrow \mathbb{B}_1$$

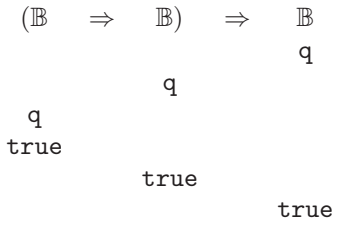
defines the arena




where the indices $1, 2, 3$ distinguish the three instances of the boolean arena $\mathbb{B}$. This arena contains the justified play

$$\mathtt{q}_1 \cdot \mathtt{q}_2 \cdot \mathtt{q}_3 \cdot \mathtt{true}_3 \cdot \mathtt{true}_2 \cdot \mathtt{true}_1 \tag{4}$$

also depicted using the convention below:




(5)

Note that the play (4-5) belongs to the strategy implemented by the PCF program $\lambda f.f(\mathtt{true})$.

**Well-bracketing.** Hyland and Ong demonstrate in their work [16] that a (finite) strategy can be implemented in PCF if and only if it satisfies two fundamental conditions, called *innocence* and *well-bracketing*. We will focus here on the well-bracketing condition, which is very similar to a *stack discipline*. The condition is usually expressed in the following way. Arenas are refined by attaching a *mode* $\lambda^{QA}(m) \in \{Q, A\}$ to every move $m$ of the arena. A move $m$ is called a *question* when $\lambda^{QA}(m) = Q$, and an *answer* when $\lambda^{QA}(m) = A$. One then requires that no answer move $m$ justifies another answer move $n$:

$$m \vdash n \quad \Longrightarrow \quad \lambda^{QA}(m) = Q \text{ or } \lambda^{QA}(n) = Q.$$

The intuition indeed is that an answer $n$ responds to the question $m$ which justifies it in the play. Note that alternation ensures that Proponent answers the questions raised by Opponent, and vice versa: hence, a player never answers his own questions. For instance, the arena game $\mathbb{B}$ is refined by declaring that the Opponent move $\mathtt{q}$ is a question, and that the two Proponent moves $\mathtt{true}$ and $\mathtt{false}$ are answers.

Now, a justified play $s$ is called *well-bracketed* when every answer $n$ appearing in the play responds to the "pending" question $m$. The terminology is supported by the intuition that (1) every question "opens" a bracket and (2) every answer "closes" a bracket, which should match the bracket opened by the answered question. Typically, the play (4-5) is well-bracketed, because every answer responds properly to the last unanswered question, thus leading to the well-bracketed sequence:

$$\begin{array}{l} \mathtt{q}_1 \cdot \mathtt{q}_2 \cdot \mathtt{q}_3 \cdot \mathtt{true}_3 \cdot \mathtt{true}_2 \cdot \mathtt{true}_1 \\ (_1 \text{———————————————} {}_1) \\ \quad (_2 \text{———————} {}_2) \\ \quad\quad (_3 \text{—} {}_3) \end{array}$$

On the other hand, the play

$$\begin{array}{lllll} (\mathbb{B} & \Rightarrow & \mathbb{B}) & \Rightarrow & \mathbb{B} \\ & & & & \mathtt{q} \\ & & \mathtt{q} & & \\ \mathtt{q} & & & & \\ & & & & \mathtt{true} \end{array} \tag{6}$$

is *not* well-bracketed, because the move $\mathtt{true}$ answers the first question of the play, whereas it should have answered the third (and pending) question. This may be depicted in the following way:

$$\begin{array}{l} \mathtt{q}_1 \cdot \mathtt{q}_2 \cdot \mathtt{q}_3 \cdot \mathtt{true}_1 \\ (_1 \text{———————} {}_1) \\ \quad (_2 \cdots\cdots\cdots\cdots \\ \quad\quad (_3 \cdots\cdots \end{array} \tag{7}$$

In fact, the play (6-7) belongs to a strategy which tests whether the function $f : \mathbb{B} \Rightarrow \mathbb{B}$ is strict, that is, interrogates its argument: this test cannot be implemented in the language PCF – although it can be implemented in PCF extended with the control operator `call-cc`, see [7, 22].

**Counting resources.** We would like to understand well-bracketing as a resource discipline, rather than simply as a stack discipline. One key step in this direction is the observation that a well-bracketed play may be detected simply by counting two specific numbers on a path:

- the number $\kappa^+$ of Proponent questions opened but left unanswered,
- the number $\kappa^-$ of Opponent questions opened but left unanswered.

Of course, it is not sufficient to count the two numbers $\kappa^+$ and $\kappa^-$ of a play $s$ to detect whether the play is well-bracketed. Typically, the well-bracketed play $(a)$ and the non well-bracketed play $(b)$ introduced in (6-7) induce the same numbers $\kappa^+$ and $\kappa^-$:

$$\begin{array}{lll} (a) & \mathtt{q}_1 \cdot \mathtt{q}_2 \cdot \mathtt{q}_3 \cdot \mathtt{true}_3 & \longmapsto \quad \kappa^+ = 1, \quad \kappa^- = 1 \\ (b) & \mathtt{q}_1 \cdot \mathtt{q}_2 \cdot \mathtt{q}_3 \cdot \mathtt{true}_1 & \longmapsto \quad \kappa^+ = 1, \quad \kappa^- = 1 \end{array}$$

In order to detect well-bracketing, one needs to apply the count to the subpaths $(c)$ and $(d)$ of these plays. This reveals a key difference:

$$\begin{array}{lll} (c) & \mathtt{q}_3 \cdot \mathtt{true}_3 & \longmapsto \quad \kappa^+ = 0, \quad \kappa^- = 0 \\ (d) & \mathtt{q}_3 \cdot \mathtt{true}_1 & \longmapsto \quad \kappa^+ = 0, \quad \kappa^- = 1 \end{array}$$

The elementary but key characterisation follows:

**Proposition 1** *A play $s$ is well-bracketed if and only if every subpath $m \cdot t \cdot n$ of the play $s$ satisfies*

$$\kappa^+(m \cdot t \cdot n) = 0 \quad \Longrightarrow \quad \kappa^-(m \cdot t \cdot n) = 0$$

*when $m$ is Opponent and $n$ is Proponent; and dually*

$$\kappa^-(m \cdot t \cdot n) = 0 \quad \Longrightarrow \quad \kappa^+(m \cdot t \cdot n) = 0$$

*when $m$ is Proponent and $n$ is Opponent.*

Let us explain this briefly. Suppose that $m \cdot t \cdot n$ is a subpath of a well-bracketed play $s$, where $m$ is Opponent and $n$ is Proponent. The first condition says that if there is an Opponent question unanswered in $m \cdot t$, then either Player answers it – in which case $\kappa^-(m \cdot t \cdot n) = 0$ – or there is a Player question unanswered in $m \cdot t \cdot n$ – in which case $\kappa^+(m \cdot t \cdot n) \neq 0$. The other condition is dual.

**A resource policy.** Reformulated in this way, the well-bracketing looks very much like a resource policy. The basic intuition is that every question $m$ emits a *query* for a *linear session*. This query is noted by a opening bracket $(_i$ and counted by $\kappa^{\pm}$ where $\pm$ is the polarity of the move $m$. The query is then complied with by a *response* emitted by an answer move $n$, and noted by a closing bracket $_i)$. In our example, the move $\mathtt{q}_3$ emits a query $(_3$ which is later complied with in the play (4-5) by the response $_3)$ emitted by the move $\mathtt{true}$ whereas it remains unanswered in the play (6-7). Hence, a play like (6-7) is not well-bracketed because it breaks the linearity policy implemented by the queries. Our game model will relate this linearity policy to the fact that the boolean formula is defined as

$$\mathbb{B} \quad = \quad \neg^O \neg^P (1 \oplus 1) \tag{8}$$

in tensorial logic. Here, the tags $O$ and $P$ are mnemonics to indicate that the external negation $\neg_O$ is interpreted as an Opponent move, whereas the internal negation $\neg_P$ is interpreted as a Proponent move. The story told by (8) goes like this: Opponent plays the external negation, followed by Proponent, who plays the internal negation and *at the same time* resolves the choice $1 \oplus 1$ between `true` and `false`. This refines the picture conveyed by the boolean arena (3) by decomposing the Player moves `true` and `false` in two compound stages: negation and choice – where negation thus encapsulates the two moves `true` and `false`. This enables to relax the well-bracketing policy by interpreting the boolean formula as

$$\mathbb{B} \quad = \quad \overset{O}{\neg} \; !_w \; \overset{P}{\neg} \; (1 \oplus 1) \tag{9}$$

where the affine modality $!_w$ of tensorial logic is inserted between the two negations. The intuitionistic hierarchy on the boolean formula (8) coincides with the well-bracketed arena game model of PCF described by Hyland and Ong in [16] whereas the intuitionistic hierarchy on the boolean formula (9) – where the affine modality $!_w$ is replaced by the exponential modality $!_e$ – coincides with the non-well-bracketed arena game model of PCF with control described by Jim Laird in [22] and Olivier Laurent in [24].

**Multi-bracketing.** This analysis leads us to the notion of *multi-bracketing* in arena games. In linear logic, every proof of the formula

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

asks the value of its two boolean arguments, and we would like to understand this as a kind of well-bracketing condition. So, the play

$$\begin{array}{ccccc} (\mathbb{B} & \otimes & \mathbb{B}) & \multimap & \mathbb{B} \\ & & & & \mathtt{q} \\ & & \mathtt{q} & & \\ & & \mathtt{true} & & \\ \mathtt{q} & & & & \\ \mathtt{true} & & & & \\ & & & & \mathtt{true} \end{array} \tag{10}$$

would be "well-bracketed" in the new setting, whereas the play

$$\begin{array}{ccccc} (\mathbb{B} & \otimes & \mathbb{B}) & \multimap & \mathbb{B} \\ & & & & \mathtt{q} \\ \mathtt{q} & & & & \\ \mathtt{true} & & & & \\ & & & & \mathtt{true} \end{array} \tag{11}$$

would not be "well-bracketed", because it does not explore the second argument of the function. This extended well-bracketing is captured by the idea that the first question emits *three* queries $(_1$ and $(_a$ and $(_b$ at the same time. Then, the play (10) appears to be "well-bracketed" if one depicts the situation in the following way:

$$\begin{array}{l} \mathtt{q}_1 \cdot \mathtt{q}_2 \cdot \mathtt{true}_2 \cdot \mathtt{q}_3 \cdot \mathtt{true}_3 \cdot \mathtt{true}_1 \\ (_1 \text{ ——————————————— } {}_1) \\ (_a - {}_a)(_2 \text{ —— } {}_2) \\ (_b \text{ ——————— } {}_b)(_3 \text{ —— } {}_3) \end{array}$$

whereas the play (11) is not "well-bracketed" because the query $(_a$ is never complied with, as can be guessed from the picture below:

$$\begin{array}{l} \mathtt{q}_1 \cdot \mathtt{q}_3 \cdot \mathtt{true}_3 \cdot \mathtt{true}_1 \\ (_1 \text{ ——————— } {}_1) \\ (_a \text{ ..... } \\ (_b - {}_b)(_3 \text{ —— } {}_3) \end{array}$$

We explain in §3 and §4 how we apply the well-bracketing criterion devised in Proposition 1 in order to generalise well-bracketing to a multi-bracketed framework.

**Plan of the paper.** We describe (§2) a categorical semantics of resources in game semantics, and explain in what sense the resulting topography refines both linear logic and polarized logic. After that, we construct (§3) a compact-closed (that is, self-dual) category of multi-bracketed Conway games and well-bracketed strategies, where the resource policy is enforced by multi-bracketing. From this, we derive (§4) a model of our categorical semantics of resources, using a family construction, and conclude (§5).

**Acknowledgements.** We would like to thank Martin Hyland together with Masahito Hasegawa, Olivier Laurent, Laurent Regnier and Peter Selinger for stimulating discussions at various stages of this work.

## 2 Categorical models of resources

We introduce now the notion of *tensorial negation* on a symmetric monoidal category; and then explain how such a category with negation may be equipped with additives and various resource modalities. The first author describes in [27] how to extract a *syntax* of proofs from a categorical semantics, using string diagrams and functorial boxes. The recipe may be applied here to extract the syntax of a logic, called *tensorial logic*. However, we provide in Appendix a sequent calculus for tensorial logic, in order to compare it to linear logic [9] or polarized linear logic [23].

**Tensorial negation.** A *tensorial negation* on a symmetric monoidal category $(\mathcal{A}, \otimes, 1)$ is defined as a functor

$$\neg \; : \; \mathcal{A} \quad \longrightarrow \quad \mathcal{A}^{op}$$

together with a family of bijections

$$\varphi_{A,B,C} \quad : \quad \mathcal{A}(A \otimes B, \neg C) \quad \cong \quad \mathcal{A}(A, \neg(B \otimes C))$$

natural in $A$, $B$ and $C$. Given a negation, it is customary to define the formula *false* as the object

$$\bot \stackrel{\text{def}}{=} \neg 1$$

obtained by "negating" the unit object $1$ of the monoidal category. Note that we use the notation $1$ (instead of $I$ or $e$) in order to remain consistent with the notations of linear logic. Note also that the bijection $\varphi_{A,B,1}$ provides then the category $\mathcal{A}$ with a one-to-one correspondence

$$\varphi_{A,B,1} \quad : \quad \mathcal{A}(A \otimes B, \bot) \quad \cong \quad \mathcal{A}(A, \neg B)$$

for all objects $A$ and $B$. For that reason, the definition of a negation $\neg$ is often replaced by the — somewhat too informal — statement that "the object $\bot$ is exponentiable" in the symmetric monoidal category $\mathcal{A}$, with negation $\neg A$ noted $\bot^A$.

**Self-adjunction.** In his PhD thesis, Hayo Thielecke [35] observes for the first time a fundamental "self-adjunction" phenomenon, related to negation. This observation plays then a key rôle in an unpublished work by Peter Selinger and the first author [30] on polar categories, a categorical semantics of polarized linear logic, continuations and games. The same idea reappears recently in a nice, comprehensive study on polarized categories (=distributors) by Robin Cockett and Robert Seely [8]. In our situation, the "self-adjunction" phenomenon amounts to the fact that every tensorial negation is left adjoint to the opposite functor

$$\neg \quad : \quad \mathcal{A}^{op} \quad \longrightarrow \quad \mathcal{A} \tag{12}$$

because of the natural bijection

$$\mathcal{A}^{op}(\neg A, B) \quad \cong \quad \mathcal{A}(A, \neg B).$$

**Continuation monad.** Every tensorial negation $\neg$ induces an adjunction, and thus a monad

$$\neg\neg \quad : \quad \mathcal{A} \longrightarrow \mathcal{A}$$

This monad is called the *continuation monad* of the negation. One fundamental fact observed by Eugenio Moggi [31] is that the continuation monad is *strong* but not commutative in general. By strong monad, we mean that the monad $\neg\neg$ is equipped with a family of morphisms:

$$t_{A,B} \quad : \quad A \otimes \neg\neg B \quad \longrightarrow \quad \neg\neg\,(A \otimes B)$$

natural in $A$ and $B$, and satisfying a series of coherence properties. By commutative monad, we mean a strong monad making the two canonical morphisms

$$\neg\neg A \otimes \neg\neg B \quad \rightrightarrows \quad \neg\neg\,(A \otimes B) \tag{13}$$

coincide. A tensorial negation $\neg$ is called *commutative* when the continuation monad induced in $\mathcal{A}$ is commutative — or equivalently, a monoidal monad in the lax sense.

**Linear implication.** A symmetric monoidal category $\mathcal{A}$ with a tensorial negation $\neg$ is not very far from being monoidal *closed*. It is possible indeed to define a *linear implication* $\multimap$ when its target $\neg B$ is a negated object:

$$A \multimap \neg B \quad \stackrel{\text{def}}{=} \quad \neg\,(A \otimes B).$$

In this way, the functor (12) defines what we call an *exponential ideal* in the category $\mathcal{A}$. When the functor is faithful on objects and morphisms, we may identify this exponential ideal with the subcategory of *negated objects* in the category $\mathcal{A}$. The exponential ideal discussed in Guy McCusker's PhD thesis [26] arises precisely in this way. This enables in particular to define the linear and intuitionistic hierarchies on the arena games (8) and (9).

**Continuation category.** Every symmetric monoidal category $\mathcal{A}$ equipped with a negation $\neg$ induces a *category of continuations* $\mathcal{A}^\neg$ with the same objects as $\mathcal{A}$, and morphisms defined as

$$\mathcal{A}^\neg(A, B) \quad \stackrel{\text{def}}{=} \quad \mathcal{A}(\neg A, \neg B).$$

Note that the category $\mathcal{A}^\neg$ is the kleisli category associated to the comonad in $\mathcal{A}^{op}$ induced by the adjunction; and that it is at the same time the opposite of the kleisli category associated to the continuation monad in $\mathcal{A}$. Because the continuation monad is strong, the category $\mathcal{A}^\neg$ is *premonoidal* in the sense of John Power and Edmund Robinson [32]. It should be noted that string diagrams in premonoidal categories are inherently related to control flow charts in software engineering, as noticed by Alan Jeffrey [18].

**Semantics of resources.** A *resource modality* on a symmetric monoidal category $(\mathcal{A}, \otimes, e)$ is defined as an adjunction:

$$\mathcal{M} \underset{F}{\overset{U}{\rightleftarrows}} \mathcal{A} \qquad (\bot) \tag{14}$$

where

- $(\mathcal{M}, \bullet, u)$ is a symmetric monoidal category,
- $U$ is a symmetric monoidal functor.

Recall that a *symmetric monoidal* functor $U$ is a functor which transports the symmetric monoidal structure of $(\mathcal{M}, \bullet, u)$ to the symmetric monoidal structure of $(\mathcal{A}, \otimes, e)$, up to isomorphisms satisfying suitable coherence properties. Another more conceptual definition of a resource modality is possible: it is an adjunction defined in the 2-category of symmetric monoidal categories, *lax* symmetric monoidal functors, and monoidal transformations. Now, the resource modality is called

- *affine* when the unit $u$ is the terminal object of the category $\mathcal{M}$,
- *exponential* when the tensor product $\bullet$ is a cartesian product, and the unit $u$ is the terminal object of the category $\mathcal{M}$.

This definition of resource modality is inspired by the categorical semantics of linear logic, and more specifically by Nick Benton's notion of Linear-Non-Linear model [4] — which may be reformulated now as a symmetric monoidal *closed* category $\mathcal{A}$ equipped with an exponential modality in our sense. Very often, we will identify the resource modality and the induced comonad $! = U \circ F$ on the category $\mathcal{A}$.

**Tensorial logic.** In our philosophy, tensorial logic is entirely described by its categorical semantics — which is defined in the following way. First, every symmetric monoidal category $\mathcal{A}$ equipped with a tensorial negation $\neg$ defines a model of *multiplicative* tensorial logic. Such a category defines a model of *multiplicative additive* tensorial logic when the category $\mathcal{A}$ has finite coproducts (noted $\oplus$) which *distribute* over the tensor product: this means that the canonical morphisms

$$\begin{array}{ccc} (A \otimes B) \oplus (A \otimes C) & \longrightarrow & A \otimes (B \oplus C) \\ 0 & \longrightarrow & A \otimes 0 \end{array}$$

are isomorphisms. Then, a model of (full) tensorial logic is defined as a model of multiplicative additive tensorial logic, equipped with an affine resource modality (with comonad noted $!_w$) as well as an exponential resource modality (with comonad noted $!_e$).

The diagrammatic syntax of tensorial logic will be readily extracted from its categorical definition, using the recipe explained in [27]. However, the reader will find a sequent calculus of tensorial logic in Appendix, written in the more familiar fashion of proof theory. Seen from that point of view, the modality-free fragment of tensorial logic describes a *linear* variant of Girard's LC [10] thus akin to ludics [11] and more precisely to what Laurent calls MALLP in his PhD thesis [23]. This convergence simply expresses the fact that these systems are all based on tensors, sums and linear continuations.

**Arena games and classical logic.** Starting from Thielecke's work, Selinger [33] designs the notion of *control category* in order to axiomatize the categorical semantics of classical logic. Then, prompted by a completeness result established by Martin Hofmann and Thomas Streicher in [15], he proves a beautiful structure theorem, stating that every control category $\mathcal{C}$ is the continuation category $\mathcal{A}^\neg$ of a *response category* $\mathcal{A}$. Now, a response category $\mathcal{A}$ — where the monic requirement on the units (2) is relaxed — is exactly the same thing as a model of multiplicative additive tensorial logic, where the tensor $\otimes$ is *cartesian* and the tensor unit $1$ is *terminal*.

A purely proof-theoretic analysis of classical logic leads exactly to the same conclusion. Starting from Girard's work on polarities in LC [10] and ludics [11], Laurent developed a comprehensive analysis of polarities in logic, incorporating classical logic, control categories and (non-well-bracketed) arena games [23, 24]. Now, it appears that Laurent's polarized logic LLP coincides with multiplicative additive tensorial logic — where the monoidal structure is *cartesian*. This is manifest in the monolateral formulation of tensorial logic, see Appendix. We sum up below the difference between tensorial logic and classical logic in a very schematic table:

| | | |
|---|---|---|
| Tensorial logic | $\otimes$ | is monoidal |
| | $\neg$ | is tensorial |
| Classical logic | $\otimes$ | is cartesian |
| | $\neg$ | is tensorial |

Note that every resource modality (14) on a category $\mathcal{A}$ equipped with a tensorial negation $\neg$ induces a tensorial negation $F^{op} \circ \neg \circ U$ on the category $\mathcal{M}$. This provides a model of polarized linear logic, and thus of classical logic, whenever $\mathcal{M}$ is cartesian. This phenomenon underlies the construction of a control category in [25], see also [12] for another construction.

**Linear logic.** The continuation monad $A \mapsto \overset{OP}{\neg\neg} A$ of game semantics lifts an Opponent-starting game $A$ with an Opponent move $\neg_O$ followed by a Player move $\neg_P$. Now, it appears that the Blass problem mentioned in §1 arises precisely from the fact that the monad is strong, but not commutative [30, 28]. Indeed, one obtains a game model of (full) propositional linear logic by *identifying* the two canonical strategies (13) — this leading to a fully complete model of linear logic expressed in the language of asynchronous games [29].

This construction in game semantics has a nice categorical counterpart. We already mentioned that the continuation category $\mathcal{A}^\neg$ inherits a premonoidal structure from the symmetric monoidal structure of $\mathcal{A}$. Now, Hasegawa Masahito shows (private communication) that the continuation category $\mathcal{A}^\neg$ equipped with this premonoidal structure is $*$-autonomous if and only if the continuation monad is commutative. The specialist will recognize here a categorification of Girard's phase space semantics [9]. Anyway, this shows that linear logic is essentially tensorial logic in which the tensorial negation is commutative.

| | | |
|---|---|---|
| Linear logic | $\otimes$ | is monoidal |
| | $\neg$ | is commutative |

In that situation, every resource modality on the category $\mathcal{A}$ induces a resource modality on the $*$-autonomous category $\mathcal{A}^\neg$, and thus a model of full linear logic.

## 3 Multi-bracketed Conway games

We define here and in §4 a game semantics with resource modalities and fixpoints, in order to interpret recursion in programming languages. We achieve this by constructing first a compact-closed category $\mathcal{B}$ of *multi-bracketed*

Conway games, inspired from André Joyal's pioneering work [19]. The compact-closed structure of $\mathcal{B}$ induces a trace operator [20] which, in turn, provides enough fixpoints in the category constructed in §4 in order to interpret the language PCF enriched with resource modalities.

**Conway games.** A Conway game is an oriented rooted graph $(V, E, \lambda)$ consisting of a set $V$ of vertices called the *positions* of the game, a set $E \subset V \times V$ of edges called the *moves* of the game, a function $\lambda : E \to \{-1, +1\}$ indicating whether a move belongs to Opponent $(-1)$ or Proponent $(+1)$. We note $\star$ the root of the underlying graph.

**Path and play.** A *play* is a path starting from the root $\star_A$ of the multi-bracketed game:

$$\star_A \xrightarrow{m_1} x_1 \xrightarrow{m_2} \dots \xrightarrow{m_{k-1}} x_{k-1} \xrightarrow{m_k} x_k \qquad (15)$$

Two paths are parallel when they have the same initial and final positions. A play (15) is *alternating* when:

$$\forall i \in \{1, \dots, k-1\}, \qquad \lambda_A(m_{i+1}) = -\lambda_A(m_i).$$

**Strategy.** A strategy $\sigma$ of a Conway game is defined as a set of alternating plays of even length such that:

- $\sigma$ contains the empty play,
- every nonempty play starts with an Opponent move,
- $\sigma$ is closed by even-length prefix: for every play $s$, and for all moves $m, n$,

$$s \cdot m \cdot n \in \sigma \quad \Longrightarrow \quad s \in \sigma,$$

- $\sigma$ is deterministic: for every play $s$, and for all moves $m, n, n'$,

$$s \cdot m \cdot n \in \sigma \text{ and } s \cdot m \cdot n' \in \sigma \quad \Longrightarrow \quad n = n'.$$

We write $\sigma : A$ when $\sigma$ is a strategy of $A$. Note that a play in a Conway game is generally non-alternating, but that alternation is required on the plays of a strategy.

**Multi-bracketed games.** A multi-bracketed game is a Conway game equipped with

- a finite set $Q_A(x)$ of *queries* for each position $x \in V$ of the game,
- a function $\lambda(x) : Q_A(x) \longrightarrow \{-1, +1\}$ which assigns to every query in $Q_A(x)$ a polarity which indicates whether the query is made by Opponent $(-1)$ or Proponent $(+1)$,
- for each move $x \xrightarrow{m} y$, a *residual* relation

$$[m] \subset Q_A(x) \times Q_A(y)$$

satisfying:

$$\begin{array}{lllll} r[m]r_1 & \text{and} & r[m]r_2 & \Longrightarrow & r_1 = r_2 \\ r_1[m]r & \text{and} & r_2[m]r & \Longrightarrow & r_1 = r_2 \end{array}$$

The definition of residuals is then extended to paths $s : x \twoheadrightarrow y$ in the usual way: by composition of relations. We then define

$$r[s] \stackrel{\text{def}}{=} \{r' \mid r[s]r'\} \quad \text{and} \quad [s]r \stackrel{\text{def}}{=} \{r' \mid r'[s]r\}.$$

We say that a path $s : x \twoheadrightarrow y$:

- *complies with a query* $r \in Q_A(x)$ when $r$ has no residual after $s$ — that is, $r[s] = \emptyset$,
- *initiates a query* $r \in Q_A(y)$ when $r$ has no ancestor before $s$ — that is, $[s]r = \emptyset$.

We require that a move $m$ only initiates queries of its own polarity, and only complies with queries of the opposite polarity. In order to formalise that a residual of a query is intuitively the query itself, we also require that two parallel paths $s$ and $t$ induce the same residual relation: $[s] = [t]$. Finally, we require that there are no queries at the root: $Q_A(\star) = \emptyset$.

**Resource function.** Extending Conway games with queries enables the definition of a resource function

$$\kappa = (\kappa^+, \kappa^-)$$

which counts, for every path $s : x \twoheadrightarrow y$, the number $\kappa^+(s)$ (respectively $\kappa^-(s)$) of Proponent (respectively Opponent) queries in $r \in Q_A(y)$ initiated by the path $s$ — that is, such that $[s]r = \emptyset$. The definition of multi-bracketed games induces three cardinal properties of $\kappa^\pm$, which will replace the very definition of $\kappa$, and will play the rôle of axioms in all our proofs – in particular, in the proof that the composite of two well-bracketed strategies is also well-bracketed.

***Property 1: accuracy.*** For all paths $s : x \twoheadrightarrow y$ and Proponent move $m : y \to z$,

$$\kappa^-(m) = 0 \quad \text{and} \quad \kappa^+(s \cdot m) = \kappa^+(s) + \kappa^+(m),$$

as well as the dual equalities for Opponent moves.

***Property 2: suffix domination.*** For all paths $s : x \twoheadrightarrow y$ and $t : y \twoheadrightarrow z$,

$$\kappa(t) \leq \kappa(s \cdot t).$$

***Property 3: sub-additivity.*** For all paths $s : x \twoheadrightarrow y$ and $t : y \twoheadrightarrow z$,

$$\kappa(s \cdot t) \leq \kappa(s) + \kappa(t).$$

Accuracy holds because Player does not initiate Opponent queries, and does not comply with Player queries. Suffix domination says that a query cannot already have been complied with. Sub-additivity expresses that composing two paths does not increase the number of queries.

**Well-bracketed plays and strategies.** Once the resource function $\kappa$ is defined on paths, it becomes possible to define a *well-bracketed play* as a play which satisfies the two conditions stated in Proposition 1 of §1. So, the property becomes a definition here. A strategy $\sigma$ is then declared

*well-bracketed* when, for every play $s \cdot m \cdot t \cdot n$ of the strategy $\sigma$ where $m$ is an Opponent move and $n$ is (necessarily) a Proponent move:

$$\kappa_A^+(m \cdot t \cdot n) = 0 \quad \Longrightarrow \quad \kappa_A^-(m \cdot t \cdot n) = 0.$$

Every well-bracketed strategy $\sigma$ then preserves well-bracketing in the following sense:

**Lemma 1** *Suppose $s \cdot m \cdot n \in \sigma$ and that $s \cdot m$ is well-bracketed. Then, $s \cdot m \cdot n$ is well-bracketed.*

Hence, when Opponent and Proponent play according to well-bracketed strategies, the resulting play is well-bracketed.

**Dual.** Every multi-bracketed game $A$ induces a *dual* game $A^*$ obtained by reversing the polarity of moves and queries. Thus, $(\kappa^+_{A^*}, \kappa^-_{A^*}) = (\kappa^-_A, \kappa^+_A)$.

**Tensor product.** The tensor product $A \otimes B$ of two multi-bracketed games $A$ and $B$ is defined as:

- its positions are the pairs $(x, y)$ noted $x \otimes y$, ie. $V_{A\otimes B} = V_A \times V_B$ with $\star_{A\otimes B} = (\star_A, \star_B)$.
- its moves are of two kinds:

$$x \otimes y \to \begin{cases} z \otimes y \text{ if } x \to z \text{ in the game } A, \\ x \otimes z \text{ if } y \to z \text{ in the game } B. \end{cases}$$

- its queries at position $x \otimes y$ are the queries at position $x$ in the game $A$ and the queries at position $y$ in the game $B$: $Q_{A\otimes B}(x \otimes y) = Q_A(x) \uplus Q_B(y)$.

The polarities of moves and queries in the game $A \otimes B$ are inherited from the games $A$ and $B$, and the residual relation of a move $m$ in the game $A \otimes B$ is defined just in the expected (pointwise) way. The unique multi-bracketed game 1 with $\{\star\}$ as underlying Conway game is the neutral element of the tensor product. As usual in game semantics, every play $s$ in the game $A \otimes B$ may be seen as the interleaving of a play $s_{|A}$ in the game $A$ and a play $s_{|B}$ in the game $B$. More interestingly, the resource function $\kappa$ is "tensorial" in the following sense:

$$\kappa_{A\otimes B}(s) = \kappa_A(s_{|A}) + \kappa_B(s_{|B}).$$

**Composition.** We proceed as in [26, 13], and say that $u$ is an interaction on three games $A, B, C$, this noted $u \in int_{ABC}$, when the projection of $u$ on each game $A^* \otimes B$, $B^* \otimes C$ and $A^* \otimes C$ is a play. Given two strategies $\sigma : A^* \otimes B, \tau : B^* \otimes C$, we define the composition of these strategies as follows:

$$\sigma;\tau = \{u_{|A^*\otimes C} \mid u \in int_{ABC}, u_{|A^*\otimes B} \in \sigma, u_{|B^*\otimes C} \in \tau\}$$

As usually, the composition of two strategies is a strategy. More interestingly, we show that our notion of well-bracketing is preserved by composition:

**Proposition 2** *The strategy $\sigma;\tau : A^* \otimes C$ is well-bracketed when the two strategies $\sigma : A^* \otimes B$ and $\tau : B^* \otimes C$ are well-bracketed.*

**Proof:** The proof is entirely based on the three cardinal properties of $\kappa$ mentioned earlier. The proof appears in the Master's thesis of the second author [34].

**The category $\mathcal{B}$ of multi-bracketed games.** The category $\mathcal{B}$ has multi-bracketed games as objects, and well-bracketed strategies $\sigma$ of $A^* \otimes B$ as morphisms $\sigma : A \to B$. The identity strategy is the usual copycat strategy, defined by André Joyal in Conway game [19]. The resulting category $\mathcal{B}$ is compact-closed in the sense of [21] and thus admits a canonical trace operator, unique up to equivalence, see [20] for details.

**Negative and positive games.** A multi-bracketed game $A$ is called *negative* when all the moves starting from the root $\star_A$ are Opponent moves; and *positive* when its dual game $A^*$ is negative. The full subcategory of negative (resp. positive) multi-bracketed games is noted $\mathcal{B}^-$ (resp. $\mathcal{B}^+$). For a multi-bracketed game $A$, we write $A^-$ for the negative game obtained by removing all the Player moves from the root.

**The exponential modality.** Every multi-bracketed game $A$ induces an *exponential* game $!A$ as follows:

- its positions are the words $w = x_1 \cdots x_k$ whose letters are positions $x_i$ of the game $A$ different from the root $\star_A$; the intuition is that the letter $x_i$ describes the current position of the $i^{th}$ copy of the game,
- its root $\star_{!A}$ is the empty word,
- its moves $w \to w'$ are either moves played in one copy:

$$w_1\, x\, w_2 \;\to\; w_1\, y\, w_2$$

  where $x \to y$ is a move in the game $A$; or moves where Opponent opens a new copy:

$$w \;\to\; w\, x$$

  where $\star_A \to x$ is an Opponent move in $A$.
- its queries at position $w = x_1 \cdots x_n$ are pairs $(i, q)$ consisting of an index $1 \le i \le n$ and a query $q$ at position $x_i$ in the game $A$.

The polarities of moves and queries are inherited from the game $A$ in the expected way, and the residual relation is defined as for the tensor product. Interestingly, the resulting multi-bracketed game $!A$ defines the free commutative comonoid associated to the well-bracketed game $A$ in the category $\mathcal{B}$. Hence, the category $\mathcal{B}$ defines a model of multiplicative exponential linear logic. This model is *degenerate* in the sense that the tensor product is equal to its dual, i.e. $(A \otimes B)^* = A^* \otimes B^*$.

**Fixpoints.** The exponential modality together with the traced symmetric monoidal structure on $\mathcal{B}$ defines a fixpoint operator in $\mathcal{B}$ as shown by Hasegawa Masahito in [14]. Remark that this construction does not require that the category $\mathcal{B}$ is cartesian.

## 4 A game model with resources

We would like to construct a model of tensorial logic based on negative multi-bracketed games. However it is meaningless to construct an affine modality on the category $\mathcal{B}^-$ itself because its unit 1 is already a terminal object in the category. So we need to introduce the notion of *pointed game.*

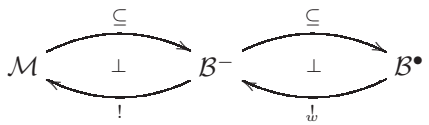
**Pointed games.** A pointed game may be seen in two different ways: (1) as a positive multi-bracketed Conway game, with a unique initial Player move, (2) as a negative multi-bracketed Conway game, except that the hypothesis that there are no queries at the root $*$ is now relaxed for Player queries. From now on, we adopt the first point of view, and thus see a pointed game as a positive game with a unique initial move. Now, a morphism $\sigma : A \longrightarrow B$ in the category $\mathcal{B}$ is called *transverse* when, for every play $mn$ of length 2 in the strategy $\sigma : A^* \otimes B$, the Opponent move $m$ is in $A$ and the Player move $n$ is in $B$. We note $\mathcal{B}^\bullet$ the subcategory of $\mathcal{B}$ with pointed games as objects, and well-bracketed transverse strategies as morphisms.

**Coalesced tensor.** Given $A, B \in \mathcal{B}^\bullet$, the coalesced tensor $A \odot B$ is the pointed game obtained from $A \otimes B$ by synchronising the two initial Player moves of $A$ and $B$. Remark that the coalesced tensor product preserves affine games, and coincides there with the tensor product of $\mathcal{B}^-$. The category $\mathcal{B}^\bullet$ equipped with $\odot$ is symmetric monoidal. It is not monoidal closed, but admits a tensorial negation. Besides, it inherits a trace operator from the category $\mathcal{B}$, which is partial, but sufficient to interpret a linear PCF with resource modalities.

**Tensorial negation.** The negation $\neg A$ of a pointed game $A$ is the pointed game obtained by lifting the dual game $A^*$ with a Proponent move $m$ which initiates one query. Then, every initial Opponent move in $A^*$ complies with this query.

**Affine modality.** A pointed game $A$ is called *affine* when its unique initial Player move does not initiate any query. Note that $\mathcal{B}^-$ is isomorphic to the full subcategory of affine games in the category $\mathcal{B}^\bullet$. The affine game $\mathop{!}_w A$ associated to a pointed game $A$ is defined by removing all the queries initiated by the first move — as well as their residuals. This defines an affine resource modality on $\mathcal{B}^\bullet$.

**Exponential modality.** The exponential modality $\mathop{!}_e$ on pointed games is obtained by composing the two adjunctions underlying the comonads $\mathop{!}_w$ and ! (defined in §3).




In particular, given a pointed game $A$, $\mathop{!}_e A$ is defined as

$$\mathop{!}_e A \quad \stackrel{\text{def}}{=} \quad !(\mathop{!}_w A)$$

**Free coproducts.** The category $\mathcal{B}^\bullet$ lacks coproducts to be a model of (full) tensorial logic. We adjust this by constructing its free completion, noted $Fam(\mathcal{B}^\bullet)$, under small coproducts [3]. Given a category $\mathcal{C}$, the objects of $Fam(\mathcal{C})$ are families $\{A_i | i \in I\}$ of objects of the category. A morphism from $\{A_i | i \in I\}$ to $\{B_j | j \in J\}$ consists of a reindexing function $f : I \to J$ together with a family of morphisms $\{f_i : A_i \to B_{f(i)} | i \in I\}$ of the category $\mathcal{C}$.

$Fam$ is a pseudo-commutative monad on $\mathcal{C}at$ [17]. Hence, the 2-monad for symmetric monoidal categories distributes over $Fam$. Consequently, (1) the category $Fam(\mathcal{C})$ inherits the symmetric monoidal structure of a symmetric monoidal category $\mathcal{C}$, and (2) the coproduct of $Fam(\mathcal{C})$ distributes over that tensor product, and (3) $Fam$ preserves monoidal adjunctions. Besides, $Fam$ preserves categories with finite products and categories with a terminal object. The construction thus preserves affine and exponential modalities in the sense of §2. Gathering all those remarks, we obtain that:

**Proposition 3** *$Fam(\mathcal{B}^\bullet)$ is a model of tensorial logic.*

Moreover, the category $Fam(\mathcal{B}^\bullet)$ has a fixpoint operator restricted on its singleton objects — that is, objects $\{A_i | i \in I\}$ where $I$ is singleton. This is sufficient to interpret a linear variant of the language PCF equipped with affine and exponential resource modalities, in the category $Fam(\mathcal{B}^\bullet)$.

## 5 Conclusion

In this paper, we integrate resource modalities in game semantics, in just the same way as they are integrated in linear logic. This is achieved by reunderstanding the very topography of the field. More specifically, linear logic is relaxed into tensorial logic, where the involutive negation of linear logic is replaced by a tensorial negation. Once this performed, it is possible to keep the best of linear logic: resource modalities, etc. but transported in the language of games and continuations. Then, linear logic coincides with tensorial logic with the additional axiom that the continuation monad is commutative. In that sense, tensorial logic is more primitive than linear logic, in the same way that groups are more primitive than abelian groups. This opens a new horizon to the subject. The whole point indeed is to understand in the future how the theory of linear logic extends to this relaxed framework. We illustrate this approach here by extending well-bracketing in arena games to the full flavour of resources in linear logic, using multi-bracketing.

## References

[1] S. Abramsky. Sequentiality vs. concurrency in games and logic. *Math. Structures Comput. Sci.*, 13:531–565, 2003.

[2] S. Abramsky and R. Jagadeesan. Games and full completeness for multiplicative linear logic. *J Sym Log*, 59(2):543–574, 1994.

[3] S. Abramsky and G. McCusker. Call-by-value games. In *CSL'97*, volume 1414 of *LNCS*. Springer-Verlag, 1998.

[4] N. Benton. A mixed linear and non-linear logic: Proofs, terms and models. In *CSL '94*, volume 933 of *LNCS*, Poland, June 1995. Springer-Verlag.

[5] A. Blass. Degrees of indeterminacy of games. *Fund. Math.*, 77:151–166, 1972.

[6] A. Blass. A games semantics for linear logic. *Ann. Pure Appl. Logic*, 56:183–220, 1992.

[7] R. Cartwright, P. Curien, and M. Felleisen. Fully Abstract Semantics for Observably Sequential Languages. *Inform. and Comput.*, 111(2):297–401, 1994.

[8] J. Cockett and R. Seely. Polarized category theory, modules, game semantics. Unpublished.

[9] J.-Y. Girard. Linear logic. *TCS*, 50:1–102, 1987.

[10] J.-Y. Girard. A new constructive logic: Classical logic. *Math. Structures Comput. Sci.*, 1(3):255–296, 1991.

[11] J.-Y. Girard. Ludics. In *this volume*. Kluwer, 2002.

[12] M. Hamano and P. Scott. A categorical semantics for polarized mall. *Ann. Pure Appl. Logic*, 145(3):276–313, 2007.

[13] R. Harmer. *Games and Full Abstraction for Nondeterministic Languages*. PhD thesis, University of London, 2000.

[14] M. Hasegawa. *Models of Sharing Graphs: A Categorical Semantics of Let and Letrec*. Springer, 1999.

[15] M. Hofmann and T. Streicher. Completeness of continuation models for $\lambda$-$\mu$-calculus. *Inform. and Comput.*, 179(2):332–355, December 2002.

[16] M. Hyland and L. Ong. On full abstraction for PCF: I, II and III. *Inform. and Comput.*, 163(2):285–408, Dec. 2000.

[17] M. Hyland and J. Power. Pseudo-commutative monads and pseudo-closed 2-categories. *J. Pure Appl. Algebra*, 175(1-3):141–185, November 2002.

[18] A. Jeffrey. Premonoidal categories and a graphical view of programs. unpublished, June 1998.

[19] A. Joyal. Remarques sur la théorie des jeux à deux personnes. *Gaz. Sc. Math. Qu.*, 1(4):46–52, 1977. English version by R. Houston available.

[20] A. Joyal, R. Street, and D. Verity. Traced monoidal categories. *Math. Proc. Cambridge Philos. Soc.*, (119(3)):447–468, 1996.

[21] M. Kelly and M. L. Laplaza. Coherence for compact closed categories. *J. Pure Appl. Algebra*, 19:193–213, 1980.

[22] J. Laird. Full abstraction for functional languages with control. In *LICS '97*, pages 58–67, 1997.

[23] O. Laurent. *Etude de la polarisation en logique*. PhD thesis, Université Aix-Marseille II, 2002.

[24] O. Laurent. Polarized games (extended abstract). In *LICS '02*, pages 265–274, 2002.

[25] O. Laurent and L. Regnier. About translations of classical logic into polarized linear logic. In *LICS '03*, 2003.

[26] G. McCusker. *Games and Full Abstraction for a Functional Metalanguage with Recursive Types*. PhD thesis, University of London, 1996.

[27] P.-A. Melliès. Functorial boxes in string diagrams. Invited talk in CSL '06.

[28] P.-A. Melliès. Asynchronous games 3: an innocent model of linear logic. In L. Birkedal, editor, *CTCS*. ENTCS, 2004.

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

[30] P.-A. Melliès and P. Selinger. Games are continuation models! Talk at Full Completeness and Full Abstraction, Satellite workshop of LICS 2001.

[31] E. Moggi. Notions of computation and monads. *Inform. and Comput.*, 93:55–92, 1991.

[32] J. Power and E. Robinson. Premonoidal categories and notions of computation. *Math. Structures Comput. Sci.*, 7:453–468, 1997.

[33] P. Selinger. Control categories and duality: on the categorical semantics of the $\lambda$-$\mu$-calculus. *Math. Structures Comput. Sci.*, 11(2):207–260, 2001.

[34] N. Tabareau. De l'opérateur de trace dans les jeux de conway. Master's thesis, ENS Cachan, 2005. Posted on HAL.

[35] H. Thielecke. *Categorical Structure of Continuation Passing Style*. PhD thesis, University of Edinburgh, 1997.


## A sequent calculus for tensorial logic

In the bilateral formulation of tensorial logic, the sequents are of two forms: $\Gamma \vdash A$ where $\Gamma$ is a context, and $A$ is a formula; $\Gamma \vdash$ where $\Gamma$ is a context (the notation $[A]$ expresses the unessential presence of $A$ in the sequent).

$$\frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}\ \otimes\text{-Right} \qquad \frac{\Gamma_1, A, B, \Gamma_2 \vdash [C]}{\Gamma_1, A \otimes B, \Gamma_2 \vdash [C]}\ \otimes\text{-Left}$$

$$\frac{}{\vdash 1}\ \text{Unit-Right} \qquad \frac{\Gamma \vdash [A]}{\Gamma, 1 \vdash [A]}\ \text{Unit-Left}$$

$$\frac{\Gamma, A \vdash}{\Gamma \vdash \neg A}\ \neg\text{-Right} \qquad \frac{\Gamma \vdash A}{\Gamma, \neg A \vdash}\ \neg\text{-Left}$$

$$\frac{}{A \vdash A}\ \text{Axiom} \qquad \frac{\Gamma \vdash A \quad A, \Delta \vdash [B]}{\Gamma, \Delta \vdash [B]}\ \text{Cut}$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \oplus B}\ \oplus\text{-Right-1} \qquad \frac{\Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma, A \oplus B \vdash C}\ \oplus\text{-Left}$$

$$\frac{\Gamma \vdash B}{\Gamma \vdash A \oplus B}\ \oplus\text{-Right-2}$$

No right introduction rule for $0$ $\qquad \frac{}{\Gamma, 0 \vdash A}\ 0\text{-Left}$

$$\frac{!_e\Gamma \vdash A}{!_e\Gamma \vdash !_e A}\ \text{Strengthening} \qquad \frac{\Gamma, A \vdash [B]}{\Gamma, !_e A \vdash [B]}\ \text{Dereliction}$$

$$\frac{\Gamma \vdash [B]}{\Gamma, !_e A \vdash [B]}\ \text{Weakening} \qquad \frac{\Gamma, !_e A, !_e A \vdash [B]}{\Gamma, !_e A \vdash [B]}\ \text{Contraction}$$

$$\frac{!_w\Gamma \vdash A}{!_w\Gamma \vdash !_w A}\ \text{Strengthening} \qquad \frac{\Gamma, A \vdash [B]}{\Gamma, !_w A \vdash [B]}\ \text{Dereliction}$$

$$\frac{\Gamma \vdash [B]}{\Gamma, !_w A \vdash [B]}\ \text{Weakening}$$

The monolateral formulation requires to polarize formulas, and to clone each construct into a negative counterpart.

| | |
|---|---|
| Positives | $0 \mid 1 \mid \downarrow L \mid P \otimes Q \mid P \oplus Q \mid !_w P \mid !_e P$ |
| Negatives | $\bot \mid \top \mid \uparrow P \mid L \parr M \mid L \& M \mid ?_w L \mid ?_e L$ |

It is then possible to reformulate all the sequent above, as illustrated below by the right and left introduction of $\otimes$.

$$\frac{\vdash \Gamma, P \quad \vdash \Delta, Q}{\vdash \Gamma, \Delta, P \otimes Q}\ \otimes \qquad \frac{\vdash \Gamma_1, L, M, \Gamma_2, [P]}{\vdash \Gamma_1, L \parr M, \Gamma_2, [P]}\ \parr$$