Games and Full Completeness for Multiplicative Linear Logic

Samson Abramsky and Radha Jagadeesan
Department of Computing
Imperial College of Science, Technology and Medicine

Technical Report DoC 92/24

September 25, 1992

Abstract

We present a game semantics for Linear Logic, in which formulas denote games and proofs denote winning strategies. We show that our semantics yields a categorical model of Linear Logic and prove *full completeness* for Multiplicative Linear Logic with the MIX rule: every winning strategy is the denotation of a unique cut-free proof net. A key role is played by the notion of *history-free* strategy; strong connections are made between history-free strategies and the Geometry of Interaction. Our semantics incorporates a natural notion of polarity, leading to a refined treatment of the additives. We make comparisons with related work by Joyal, Blass *et al.*

Contents

1 Introduction

We present a Game Semantics for Linear Logic [Gir87], in which formulas denote games, and proofs denote winning strategies. We also prove a novel kind of Completeness Theorem for this semantics, which says that every strategy in the model is the denotation of some proof.

Our motivation is threefold:

- We believe that the Game Semantics captures the dynamical intuitions behind Linear Logic better than any other extant semantics.
- We see Game Semantics as potentially providing a very powerful unifying framework for the semantics of computation, allowing typed functional languages, concurrent processes and complexity to be handled in an integrated fashion.
- Game Semantics mediates between traditional operational and denotational semantics, combining the good structural properties of one with the ability to model computational fine structure of the other. This is similar to the motivation for the Geometry of Interaction programme [Gir89b, Gir89a, AJ92a]; indeed, we shall exhibit strong connections between our semantics and the Geometry of Interaction.

1.1 Overview of Results

Blass has recently described a Game semantics for Linear Logic [Bla92b]. This has good claims to be the most intuitively appealing semantics for Linear Logic presented so far. However, there is a considerable gap between Blass' semantics and Linear Logic:

1. The semantics validates Weakening, so he is actually modelling Affine logic.
2. Blass characterises validity in his interpretation for the multiplicative fragment: a formula is game semantically valid if and only if it is an instance of a binary classical propositional tautology (where tensor, par, linear negation are read as classical conjunction, disjunction and negation). Thus there is a big gap even between provability in Affine logic and validity in his semantics.

This leaves open the challenge of refining Blass' interpretation to get a closer fit with Linear Logic, while retaining its intuitive appeal.

On the other hand, there is the challenge of obtaining a *full completeness theorem.* The usual completeness theorems are stated with respect to provability; a full completeness theorem is with respect to proofs. This is best formulated in terms of a categorical model of the logic, in which formulas denote objects, and proofs denote morphisms. One is looking for a model $\mathbb{C}$ such that:

Completeness: $\mathbb{C}(A, B)$ is non-empty only if $A \vdash B$ is provable in the logic.

Full Completeness: Any $f : A \rightarrow B$ is the denotation of a proof of $A \vdash B$. (This amounts to asking that the unique functor from the relevant free category to $\mathbb{C}$ be full, whence our terminology). One may even ask for there to be a *unique* cut-free such proof, *i.e.* that the above functor be faithful.

With full completeness, one has the tightest possible connection between syntax and semantics. We are not aware of any previously published results of this type; however, the idea is related to representation theorems in category theory [FS91]; to full abstraction theorems in programming language semantics [Mil75, Plo77]; to studies of parametric polymorphism [BFSS90, HRR89]; and to the completeness conjecture in [Gir91a].

We now make a first statement in broad terms of our results. We have refined Blass' game semantics for Linear Logic. This refinement is not a complication; on the contrary, it makes the definitions smoother and more symmetric. Thus, we get a categorical model of the logic, while Blass does not. Then, we prove a Full Completeness Theorem for this semantics, with respect to MLL + MIX (Multiplicative Linear Logic plus the Mix Rule). Recall that the MIX rule [Gir87] has the form

$$\frac{\vdash \Gamma \quad \vdash \Delta}{\vdash \Gamma, \Delta}$$

There is a notion of proof net for this logic: this uses the Danos/Regnier criterion [DR89], simply omitting the connectedness part. Thus, a proof structure will be a valid proof net for MLL + MIX just if, for every switching, the corresponding graph is acyclic. This criterion was studied by Fleury and Retoré [FR90], used by Blute in his work on coherence theorems [Blu92], and adapted by Lafont for his work on interaction nets [Laf90].

Now we can state our result in more precise terms.

Theorem 1 *Every proof net in MLL + MIX denotes a uniform, history independent winning strategy for Player in our game interpretation. Conversely, every such strategy is the denotation of a unique cut-free proof net.*

Of course, we now have to explain uniform, history independent strategies. Note that a formula in MLL + MIX is built from atomic formulas and the binary connectives tensor and par. Its denotation will then be a *variable type*. We construe this as a functor over a category of games and embeddings, in the fashion of domain theoretic semantics of polymorphism [Gir86, CGW87]. (In fact, this interpretation of variable types is part of our game theoretic semantics of polymorphism). An element of variable type, the denotation of a proof of $\Gamma(\vec{\alpha})$, where $\vec{\alpha}$ enumerates the atoms occurring in Γ, will then be a family of strategies $\{\sigma_{\vec{A}}\}$, one for each tuple of games $\vec{A}$ instantiating $\vec{\alpha}$. The uniformity of this family is expressed by the condition that it is a natural transformation $\sigma : F^- \rightarrow F^+$, where F^-, F^+ are functors derived from Γ as explained in Section 3.6.

A history independent strategy is one in which the player's move is a function only of the last move of the opponent and not of the preceding history of the play. Thus such a strategy is induced by a partial function on the set of moves in the game. The interpretation of proofs in MLL + MIX by strategies, when analysed in terms of these underlying functions on moves, turns out to be very closely related to the Geometry of Interaction interpretation [Gir89b, Gir89a, Gir88].

The contents of the reminder of this paper are as follows. Section 2 reviews MLL + MIX. Section 3 describes our game semantics for MLL + MIX. Section 4 is devoted to the proof of the Full Completeness Theorem. Section 5 outlines how our semantics can be extended to full Classical Linear Logic. Section 6 makes comparisons with related work.

2 MLL+MIX

The formulas A, B, $C, \ldots$ of MLL + MIX are built up from propositional atoms α, β, $\gamma, \ldots$ and their linear negations $\alpha^\perp$, $\beta^\perp$, $\gamma^\perp, \ldots$ by tensor ($\otimes$) and par ($\mathbin{⅋}$). The sequent calculus presentation of MLL + MIX is as follows.

Identity Group	$\dfrac{}{\vdash \alpha^{\perp}, \alpha}$ Identity	$\dfrac{\vdash \Gamma, A \qquad \vdash \Delta, A^{\perp}}{\vdash \Gamma, \Delta}$ Cut
Structural Group	$\dfrac{\vdash \Gamma}{\vdash \sigma\Gamma}$ Exchange	$\dfrac{\vdash \Gamma \qquad \vdash \Delta}{\vdash \Gamma, \Delta}$ Mix
Multiplicatives	$\dfrac{\vdash \Gamma, A \qquad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B}$ Tensor	$\dfrac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B}$ Par

We have restricted the Identity axioms to propositional atoms; this does not affect provability.

2.1 An aside: Units

Our presentation has not included the *units* $\mathbf{1}$ for Tensor and $\perp$ for Par. The rules for these, *together* with the nullary version of MIX, would be as follows.

Tensor Unit	Par Unit	Mix0
$\dfrac{}{\vdash \mathbf{1}}$	$\dfrac{\Gamma}{\vdash \Gamma, \perp}$	$\dfrac{}{\vdash}$

In fact, in the presence of the units, MIX can equivalently be expressed by declaring $\mathbf{1} = \perp$. It is easily checked that MIX and MIX0 are derivable from this, and conversely that $\vdash \mathbf{1}, \mathbf{1}$ and $\vdash \perp, \perp$ are derivable from MIX and MIX0. But with $\mathbf{1} = \perp$, clearly any sequent will be equivalent to one in which the units do not occur. Thus, we prefer to omit the units from our system.

2.2 Proof nets for MLL+MIX

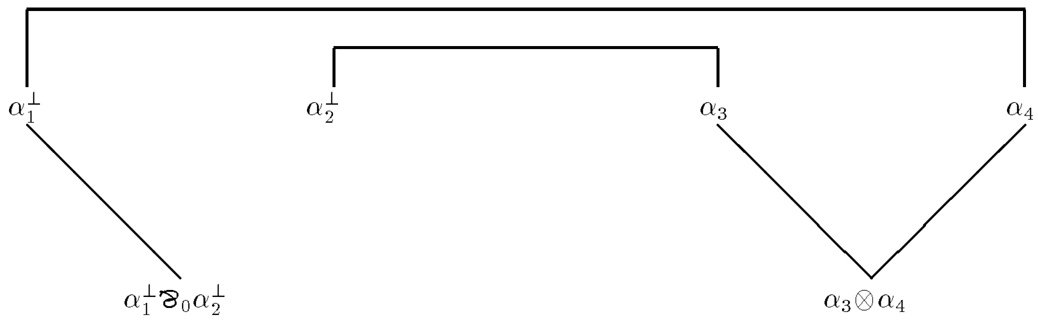
Proof structures can be defined for MLL + MIX just as for MLL [Gir87, DR89]. Alternatively, since we only allow atomic instances of identity axioms, we can define a proof structure to be a pair (Γ, ϕ), where Γ is a sequent and ϕ is a fixpoint free involution on the set of occurrences of literals in Γ, such that, if o is an occurrence of l, $\phi(o)$ is an occurrence of $l^{\perp}$. Thus, ϕ specifies the axiom links of the proof structure; all the other information is already conveyed by Γ.

A *switching* S for a proof structure (Γ, ϕ) is an assignment of L or R to each occurrence of $\parr$ in Γ. We then obtain a graph $G(\Gamma, \phi, S)$ from the formation trees of the formulas of Γ, together with the axiom links specified by ϕ, with unswitched arcs as specified by S deleted.

Example:

$$\begin{array}{rcll} \Gamma & = & \alpha_1^{\perp} \parr_0 \alpha_2^{\perp}, \alpha_3 \otimes \alpha_4 & \text{(subscripts are used to label occurrences)} \\ \phi & = & 1 \leftrightarrow 4, 2 \leftrightarrow 3 & \\ S & = & 0 \mapsto \mathrm{L} & \end{array}$$

Then $G(\Gamma, \phi, S)$ is:



Definition 1 *A (cut-free) proof net for MLL+MIX is a proof structure* (Γ, ϕ) *such that, for all switchings* S*,* $G(\Gamma, \phi, S)$ *is acyclic.*

Fleury and Retoré [FR90] make a detailed study of this criterion, which is of course just a modification of the Danos-Regnier criterion [DR89], to accomodate the MIX rule by dropping the connectedness condition. We can regard proof nets as the canonical representations of (cut-free) proofs in MLL + MIX.

3 The Game Semantics

3.1 Basic Notions on Games

This section describes the basic notions of Game and Strategy and relates these ideas to Domain Theory and Processes.

We begin by fixing some notation. If X is a set, we write $X^\star$ for the set of finite sequences (words, strings) on X and X^ω for the set of infinite sequences. If $f : X \to Y$, then $f^\star : X^\star \to Y^\star$ is the unique monoid homomorphism extending f. We write $|s|$ for the length of a finite sequence. If $Y \subseteq X$ and $s \in X^\star$, we write $s{\restriction}Y$ for the result of deleting all occurrences of symbols not in Y from s. If $a \in X$ and $s \in X^\star$, we write $a \cdot s$ ($s \cdot a$) for the result of prefixing (postfixing) s with a. We write $s \sqsubseteq t$ if s is a prefix of t, *i.e.* for some u $su = t$. We always consider sequences under this prefix ordering and use order-theoretic notions [DP90] without further comment.

3.1.1 Games

The games we consider are between Player and Opponent. A *play* or *run* of the game consists of an alternating sequence of moves, which may be finite or infinite. Each play has a determinate outcome; one player wins and the other loses. Our plays are always with Opponent to move first.

Definition 2 *A game is a structure* $A = (M_A, \lambda_A, P_A, W_A)$*, where*

- M_A *is the set of moves.*
- $\lambda_A : M_A \to \{P, O\}$ *is the labelling function to indicate if a move is by Player or Opponent. We write* $M_A^+ = \lambda_A^{-1}(\{P\})$*,* $M_A^- = \lambda_A^{-1}(\{O\})$ *and* $\overline{P} = O$*,* $\overline{O} = P$*.*
- *Let* $M_A^\circledast$ *be the set of all alternately-labelled finite sequences of moves,* i.e.

$$M_A^\circledast = \{s \in M_A^\star \mid (\forall i : 1 \leq i < |s|)\,[\lambda_A(s_{i+1}) = \overline{\lambda_A(s_i)}]\}$$

Then P_A*, the set of valid* positions *of the game, is a non-empty prefix closed subset of* $M_A^\circledast$*.*

- *Let P_A^∞ be the set of all infinite sequences of moves, all of whose finite prefixes are in P_A. W_A is a subset of P_A^∞, indicating which infinite plays are won by Player.*

An Important Remark: Note that P_A may contain positions in which the opening move is by Player, even though all *plays* in A must be started by Opponent. This becomes significant when games are combined, *e.g.* with tensor. Sections 5 and 6 discuss this point in detail.

3.1.2 Strategies

A *strategy* for Player (with Opponent to start) in A is usually defined to be a partial function from positions (with Player to move) to moves (by Player). We prefer the following definition, which leads to a more elegant treatment of composition.

Definition 3 *A* strategy *is a non-empty prefix closed subset $\sigma \subseteq P_A$ satisfying*

(s1) $a \cdot s \in \sigma \Rightarrow \lambda_A(a) = O$.

(s2) *If $s \cdot a,\ s \cdot b \in \sigma$, Player to move at s, then $a = b$.*

(s3) *If $s \in \sigma$, Opponent to move at s, $s \cdot a \in P_A$, then $s \cdot a \in \sigma$.*

Of these conditions, the first incorporates the convention that Opponent is to start; and the second enforces that strategies are *deterministic*. Note that any strategy σ does indeed determine a partial function $\hat{\sigma}$ on positions with Player to move.

We can readily define the notion of a strategy for Opponent (with Opponent to start) in A, by interchanging Player and Opponent in conditions **(s2)** and **(s3)**. Such a strategy is called a *counter-strategy*. Given a strategy σ and a counter-strategy τ, we can define the play that results when Player follows σ and Opponent follows τ:

$$\langle \sigma \mid \tau \rangle \ = \bigsqcup (\sigma \cap \tau)$$

Here $\sigma \cap \tau$ is an ideal of the poset P_A, in fact a down-closed chain. Its join s, taken in the directed completion of P_A, $P_A \cup P_A^\infty$, is a finite or infinite play. In the former case, the player who is to play at s loses; in the latter case, Player wins if and only if $s \in W_A$. A strategy is *winning* if it beats all counter-strategies.

3.1.3 Games and Domain theory

The following table draws an analogy between games and Domain theory.

Game	Information System
Strategy	Domain Element
Winning Strategy	Total Element

3.1.4 Games and Processes

The following table draws a much richer analogy between games and concurrent processes.

Game	Process Specification
Moves	Alphabet or Sort of actions
Player	System
Opponent	Environment
P_A	Safety specification
W_A	Liveness specification
Strategy	Process
Strategy in A	Process satisfying safety specification "Partial correctness"
Winning Strategy	Deadlock-free process satisfying liveness specification "Total correctness"

3.2 The Game interpretation of the Multiplicatives

Linear Negation

$$A^{\perp} = (M_A, \overline{\lambda_A}, P_A, P_A^{\infty} \setminus W_A)$$

where $\overline{\lambda_A}(a) = \overline{\lambda_A(a)}$. Clearly $A^{\perp\perp} = A$.

Tensor

The game $A \otimes B$ is defined as follows.

- $M_{A\otimes B} = M_A + M_B$, the disjoint union of the two move sets.
- $\lambda_{A\otimes B} = [\lambda_A, \lambda_B]$, the source tupling.
- $P_{A\otimes B}$ is the set of all alternately labelled finite sequences of moves such that:
  1. The restriction to the moves in M_A (resp. M_B) is in P_A (resp. P_B)
  2. If two successive moves are in different components, (*i.e.* one is in A and the other is in B), it is the Opponent who has switched components.
- $W_{A\otimes B}$ is the set of infinite plays of the game, such that the restriction to each component is either finite or is a win for Player in that component.

The tensor unit is given by

$$\mathbf{1} = (\varnothing, \varnothing, \{\epsilon\}, \varnothing)$$

Note that $\perp = \mathbf{1}^{\perp} = \mathbf{1}$.

Other Connectives

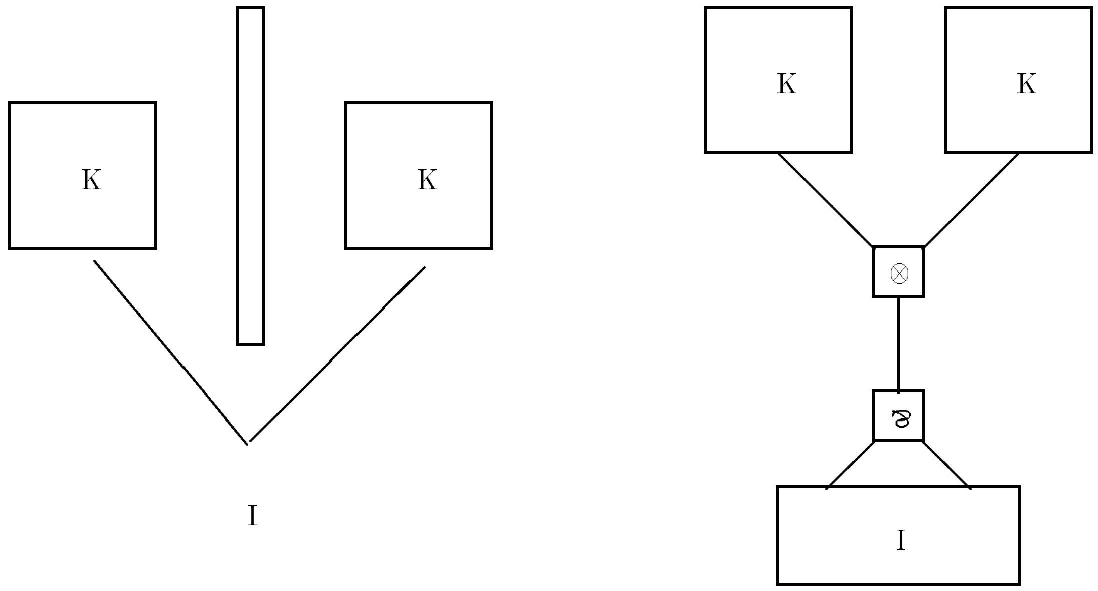
The other multiplicative connectives can be defined from Tensor and Linear negation:

$$\begin{array}{rcl} A \parr B & = & (A^{\perp} \otimes B^{\perp})^{\perp} \\ A \multimap B & = & A^{\perp} \parr B \end{array}$$

Comment on the definitions

Note that positions in A with first move by Player can indeed be significant for plays in $A^\perp, A\otimes B$ etc. This will be more fully discussed in relation to Blass' definitions in Section 6. The main point that we wish to make here is that there are clear intuitions behind our definition of $P_{A\otimes B}$ (and similarly of $P_{A \parr B}$, $P_{A\multimap B}$).

The first condition on $P_{A\otimes B}$ says that a play in $A\otimes B$ consists of (an interleaved representation of) concurrent plays in A and B. (Compare this with the definition of composition without communication in the trace model of CSP [Hoa85]). The second condition, that Player must move in the same component in which Opponent last moved, while Opponent is free to switch components, reflects the fundamental meaning of, and difference between Tensor and Par. Tensor is *disjoint* concurrency; Par is *connected* concurrency. That is, Tensor combines two processes in parallel with no flow of information between them; while Par allows flow of information. (More precisely, in MLL flow is *required* for Par; this is the content of the connectedness part of the proof-net criterion. In MLL + MIX, flow is *permitted* but not obligatory, so that Tensor becomes a special case of Par.) These constraints on the flow of information are reflected in game-theoretic terms as follows. The Player for Tensor (or Opponent for Par) must respond in the component in which his adversary moved; while Opponent for Tensor (or Player for Par) is allowed to use the moves of his adversary in one component to influence his play in the other component. In this way we get the chess game strategy by which I can defeat Karpov or Kasparov if I play against them in the following configuration[1]:



and I play white in one game and black in the other. (The vertical rectangle represents a screen between Karpov and Kasparov that prevents each from seeing the other's game board, while I can see both games). This "copy-cat" strategy is the game-theoretic content of the Identity axiom $\vdash A^\perp, A$ (or equivalently $\vdash A^\perp \parr A$).

These ideas can also be related to the trip condition for proof nets [Gir87]: the difference between Tensor and Par is expressed thus in terms of the trip condition ([Gir87] Introduction, Section III.4.3):

- "In the case of $\otimes$ there is no cooperation: if we start with $A^\wedge$, then we come back through $A_\vee$ before entering $B^\wedge$ after which we come back through $B_\vee$.

[1]This example is taken from [LS91], but the same idea can be found in [Con76].

- in the case of ⅋ there is cooperation: if we start again with $A^\wedge$, then we are expected through $B_\vee$, from which we go to $B^\wedge$ and eventually come back through $A_\vee$."

Thus we get the following possible transitions in trips:

$A{\otimes}B$: $A^\wedge A_\vee B^\wedge B_\vee$ or $B^\wedge B_\vee\ A^\wedge A_\vee$

$A⅋B$: $A^\wedge B_\vee B^\wedge A_\vee$ or $B^\wedge A_\vee A^\wedge B_\vee$

If we correlate "questions", in the terminology of [Gir87], with moves by Opponent and "answers" with moves by Player, this says exactly that only Opponent (Player) may switch between components in a Tensor (Par) game.

3.3 The Category of Games

We build a category $\mathcal{G}$ with games as objects and winning strategies as morphisms. The objects of $\mathcal{G}$ are games; the morphisms $\sigma : A \rightarrow B$ are the winning strategies in $A \multimap B = A^\perp ⅋ B$.

The composition of strategies can be defined elegantly in terms of the set representation. Firstly, a preliminary definition. Given a sequence of games $A_1, \ldots, A_n$, we define $\mathcal{L}(A_1, \ldots A_n)$, the *local* strings on $A_1, \ldots, A_n$, to be the set of all $s \in (M_{A_1} + \cdots + M_{A_n})^\star$ such that, for all i with $1 \leqslant i < |s|$, $s_i \in M_{A_j}$ and $s_{i+1} \in M_{A_k}$ implies that j is adjacent to k, *i.e.* $|j-k| \leqslant 1$. Now, given $\sigma : A \rightarrow B$, $\tau : B \rightarrow C$, define

$$\sigma;\tau = \{s{\upharpoonright}A, C \mid s \in \mathcal{L}(A,B,C),\ s{\upharpoonright}A,B \in \sigma,\ s{\upharpoonright}B,C \in \tau\}$$

Here, $s{\upharpoonright}X,Y$ means the result of deleting all moves in s not in M_X or M_Y. Note that this definition clearly exhibits the "Cut = Parallel Composition + Hiding" paradigm proposed by the first author [Abr91] as the correct computational interpretation of Cut in Classical Linear Logic, with respect to the CSP-style trace semantics for parallel composition and hiding [Hoa85]. What makes the game semantics so much richer than trace semantics is the explicit representation of the environment as the Opponent.

Proposition 1 *If $\sigma : A \rightarrow B$, $\tau : B \rightarrow C$ are winning strategies, so is $\sigma;\tau$.*

Proof: Let $S = \{s \in \mathcal{L}(A,B,C) \mid s{\upharpoonright}A,B \in \sigma,\ s{\upharpoonright}B,C \in \tau\}$ so that $\sigma;\tau = \{s{\upharpoonright}A,C \mid\ s \in S\}$. Firstly, note that $\sigma;\tau$ is non-empty and prefix closed because S is.

Since $s \in S$ implies $s{\upharpoonright}A,B \in \sigma$, $(s{\upharpoonright}A,C){\upharpoonright}A = (s{\upharpoonright}A,B){\upharpoonright}A \in P_A$ and similarly, $(s{\upharpoonright}A,C){\upharpoonright}C \in P_C$.

Now, suppose $s{\upharpoonright}A,C = t \cdot a \cdot c \in \sigma;\tau$ with $s \in S$, $a \in M_A$, $c \in M_C$. Since $s \in \mathcal{L}(A,B,C)$, we must have $s = s' \cdot a \cdot b_1 \cdot \ldots \cdot b_k \cdot c$, for some $b_1, \ldots, b_k \in M_B$ with $k \geq 1$. Moreover,

$$\begin{array}{rcl} (s'{\upharpoonright}A,B) \cdot a \cdot b_1 \cdot \ldots \cdot b_k & \in & \sigma \\ (s'{\upharpoonright}B,C) \cdot b_1 \cdot \ldots \cdot b_k \cdot c & \in & \tau \end{array}$$

Hence, a must be an O-move and c must be a P-move. A symmetric argument applies when $t \cdot c \cdot a \in \sigma;\tau$. We have shown that $\sigma;\tau \subseteq P_{A \multimap C}$.

Next, note that if $s \in S$, s cannot start with a move in B since this would violate **(s1)** either for $s{\upharpoonright}A,B \in \sigma$, or for $s{\upharpoonright}B,C \in \tau$. If $s = a \cdot s'$ with $a \in M_A$, then $a \cdot (s'{\upharpoonright}A,B) \in \sigma$, so a is an O-move by **(s1)** applied to σ; and similarly if $s = c \cdot s'$ with $c \in M_C$. Thus, $\sigma;\tau$ satisfies **(s1)**.

Given $t \in \sigma;\tau$ we say that s *covers* t if

- $s \in \mathcal{L}(A,B,C)$

- $s \upharpoonright A, C = s$
- $s \upharpoonright A, B \in \sigma,\ s \upharpoonright B, C \in \tau$

We claim that for each $t \in \sigma;\tau$ there is a *least* s covering t; we write $s \succ t$ in this case. Moreover, we claim that if $t \in \sigma;\tau$ with Opponent to move at t, then for any d such that $t \cdot d \in P_{A \multimap C}$, there is a unique e such that $t \cdot d \cdot e \in \sigma;\tau$. We will prove these claims by simultaneous induction on $|t|$.

- $\epsilon \succ \epsilon$
- If $t = t' \cdot d$, where d is an O-move, then by induction we have $s' \succ t'$, and then $s = s' \cdot d \succ t' \cdot d = t$. Note that this is well defined: since $t' \cdot d$ is in $P_{A \multimap C}$, either $t' = \epsilon$ or d is in the same component as the previous P-move. By minimality of s', either $s' = \epsilon$ or $s' = s'' \cdot e \cdot d$, where e is the previous P-move in t. In either case, $s \upharpoonright A, B \in \sigma,\ s \upharpoonright B, C \in \tau$ as required.
- If $t = t' \cdot d$, where d is a O-move, then by induction hypothesis, we have $s = s' \cdot d \succ t$. Suppose $d \in M_A$ (the case of $d \in M_C$ is symmetrical).

  Since σ is a winning strategy in $A \multimap B$, it has a unique response e to $(s \upharpoonright A, B) \cdot d$, which is either $e = a' \in A$, or $e = b_1 \in B$. Moreover, e is the unique move such that $s' \cdot d \cdot e \in S$, by the requirements that e is in A or B and that $(s \cdot d \cdot e) \upharpoonright A, B \in \sigma$. If $e = b_1$, then b_1 is an O-move in $B^{\perp}$, and since τ is a winning strategy in $B \multimap C$, it has a unique response to $(s \cdot d \cdot b_1) \upharpoonright B, C$, which will be either $b_2 \in B$ or $c' \in C$. Continuing in this way, we obtain a uniquely determined sequence of extensions of s in S. Either this sequence culminates in $s \cdot d \cdot b_1 \cdot \ldots \cdot b_k \cdot e$, where e lies in A or C, or the sequence of "internal" moves in B is infinite. We claim that the latter situation cannot in fact apply; for if it did, we would have infinite plays $u = (s \cdot d \cdot b_1 \cdot b_2 \cdots) \upharpoonright A, B$ in $A \multimap B$ following σ and $v = (s \cdot d \cdot b_1 \cdot b_2 \cdots) \upharpoonright B, C$ in $B \multimap C$ following τ. Since $u \upharpoonright A$ and $v \upharpoonright C$ are finite, and $u \upharpoonright B = v \upharpoonright B^{\perp}$, Player must lose in one of these plays, contradicting the hypothesis that σ and τ are both winning. It is clear that $s \cdot d \cdot b_1 \cdot \ldots \cdot b_k \cdot e \succ t \cdot d \cdot e$.

Thus $\sigma;\tau$ satisfies **(s2)**, and moreover has a well defined response at all positions with Player to move. It remains to be shown that if Player follows $\sigma;\tau$ he wins all infinite plays. Let s be such a play; we must show that if $s \upharpoonright A \in W_A$ then $s \upharpoonright C \in W_C$. Let $\{s_k\}$ be the increasing sequence of finite prefixes of s. Let $\{t_k\}$ be the corresponding increasing sequence where $t_k \succ s_k$. Let $t = \bigsqcup t_k$. Then $t \upharpoonright A, B$ is an infinite play following σ and $t \upharpoonright B, C$ is an infinite play following τ. If $s \upharpoonright A = t \upharpoonright A \in W_A$, then since σ is winning, $t \upharpoonright B \in W_B$; and then since τ is winning, $t \upharpoonright C = s \upharpoonright C \in W_C$, as required. ∎

Note that part of what we proved is that when two winning strategies are composed, we cannot get infinite "chattering" (*i.e.* internal communication) in the terminology of CSP [Hoa85].

Proposition 2 *$\mathcal{G}$ is a category.*

Proof: We define the identity morphism $\mathtt{id}_A : A \rightarrow A$ as

$$\mathtt{id}_A = \{s \in P_{A \multimap A} \mid \ s \text{ begins with an } O\text{-move, } (\forall t \sqsubseteq s)\,(|t| \text{ even} \Rightarrow t \upharpoonright A = t \upharpoonright A^{\perp})\}$$

In process terms, this is a bidirectional one place buffer [Abr91]. In game terms, this is the copy-cat strategy discussed previously.

Next, we prove associativity. Given $\sigma : A \rightarrow B,\ \tau : B \rightarrow C,\ \upsilon : C \rightarrow D$, we will show that $(\sigma;\tau);\upsilon = S$, where

$$S = \{t \upharpoonright A, D \mid t \in \mathcal{L}(A, B, C, D),\ t \upharpoonright A, B \in \sigma,\ t \upharpoonright B, C \in \tau,\ t \upharpoonright C, D \in \upsilon\}$$

A symmetrical argument shows that $\sigma;(\tau;\upsilon) = S$, whence we get the required result.

The inclusion $S \subseteq (\sigma;\tau);\upsilon$ is straightforward. Write

$$(\sigma;\tau);\upsilon \;=\; \{s\restriction A, D \mid s \in \mathcal{L}(A,C,D),\ s\restriction C, D \in \upsilon,\ (\exists t \in \mathcal{L}(A,B,C))\,[t\restriction A, B \in \sigma,\ t\restriction B, C \in \tau,\ t\restriction A, C = s\restriction A, C]\}$$

Given $u\restriction A, D \in S$, $u\restriction A, B, C$ witnesses that $u\restriction A, C \in \sigma;\tau$, while $u\restriction C, D \in \upsilon$ by assumption. Hence, $u\restriction A, D \in (\sigma;\tau);\upsilon$.

For the converse, a witness t such that $t\restriction A, D \in S$ may be constructed from $s \in (\sigma;\tau);\upsilon$ by the same argument used to construct $t \succ s$ in Proposition 1. ∎

3.4 History-free strategies

We will be interested in a restricted class of strategies, the history-free (or history independent, or history insensitive) ones. A strategy for Player is history-free if there is some partial function $f : M_A^- \to M_A^+$, such that at any position $s \cdot a$, with Player to move,

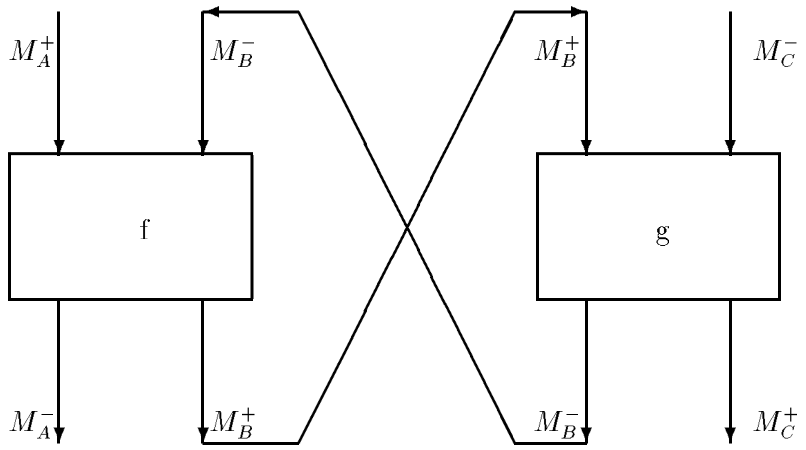
$$\hat{\sigma}(s \cdot a) = \begin{cases} f(a), & f(a) \text{ defined and } s \cdot a \cdot f(a) \in P_A \\ \text{undefined}, & \text{otherwise} \end{cases}$$

Clearly, in this case, there is a *least* partial function inducing σ; we write $\sigma = \sigma_f$, always meaning this least f. It is important to note that the category $\mathcal{G}$ described in subsection 3.3 also forms a model of MLL + MIX. However, to obtain a precise correspondence with the logic, we will focus our attention on the sub-category $\mathcal{G}_{\mathtt{hf}}$ of history-free strategies.

A history-free strategy $\sigma = \sigma_f$ is uniquely determined by the underlying function f on moves. In particular, all the morphisms witnessing the $\star$-autonomous structure in $\mathcal{G}_{\mathtt{hf}}$, or equivalently the interpretations of proofs in MLL + MIX [See89], can be defined directly in terms of these functions. When we do so, we find that the interpretation coincides exactly with the Geometry of Interaction interpretation [Gir89b, Gir89a, Gir88]. More precisely, it corresponds to a reformulation of the Geometry of Interaction, due to the present authors, in a typed version based on sets and partial functions, in the same spirit as the $\mathcal{GI}(\mathbb{C})$ construction of [AJ92a].

3.4.1 Games and the Geometry of Interaction

As a first illustration, we consider composition again. Say we have $\sigma_f : A \to B$, $\tau_g : B \to C$. We want to find h such that $\sigma_f;\tau_g = (\sigma;\tau)_h$. We shall compute h by the "execution formula" [Gir89b, Gir89a, Gir88], cut down to its actual content, which is adequately described in terms of sets and partial functions. Before giving the formal definition, let us explain the idea, which is rather simple. We want to hook the strategies up so that Player's moves in B under σ get turned into Opponent's moves in $B^\perp$ for τ, and vice versa. Consider the following picture:



Assume that the Opponent starts in A. There are two possible cases:

- The move is mapped by f to a response in A: In this case, this is the response of the function h.
- The move is mapped by f to a response in B. In this case, this response is interpreted as a move of the Opponent in $B^\perp$ and fed as input to g. In turn, if g responds in C, this is the response of the function h. Otherwise, if g responds in $B^\perp$, this is fed back to f. In this way, we get an internal dialogue between the strategies f and g; this dialogue *cannot be infinite*, because σ, τ are both *winning* strategies.

Thus, "termination of Cut-elimination", or nilpotency in terms of the Geometry of Interaction, corresponds to "no infinite internal chattering" in process-algebra terms.

It remains to give a formula for computing h according to these ideas. This is the execution formula:

$$h = \bigvee_{k \in \omega} m_k$$

The join in the definition of h can be interpreted concretely as union of graphs. It is well-defined because it is being applied to a family of partial functions with pairwise disjoint domains of definition. The functions $m_k : M_A^+ + M_C^- \rightharpoonup M_A^- + M_C^+$ are defined by

$$m_k = \pi^\star \circ ((f+g) \circ \mu)^k \circ (f+g) \circ \pi$$

The idea is that m_k is the function which, when defined, feeds an input from M_A^+ or M_C^- exactly k times around the internal feedback loop and then exits from M_A^- or M_C^+. The retraction

$$\pi : M_A + M_C \lhd M_A + M_B + M_B + M_C : \pi^\star$$

is defined by

$$\pi^\star = [\mathtt{inl}, 0, 0, \mathtt{inr}] \qquad \pi = [\mathtt{in}_1, \mathtt{in}_4]$$

and the "message exchange" function $\mu : M_A^- + M_B^+ + M_B^- + M_C^+ \rightharpoonup M_A^+ + M_B^- + M_B^+ + M_C^-$ is defined by

$$\mu = 0 + [\mathtt{inr}, \mathtt{inl}] + 0$$

Here, 0 is the everywhere undefined partial function.

3.4.2 The Category of Games and History-free strategies

We build a category $\mathcal{G}_{\mathtt{hf}}$ with games as objects and history-free winning strategies as morphisms. The objects of $\mathcal{G}_{\mathtt{hf}}$ are games; the morphisms $\sigma : A \to B$ are the history-free winning strategies in $A \multimap B = A^{\perp} \invamp B$.

Proposition 3 *$\mathcal{G}_{\mathtt{hf}}$ is a sub-category of $\mathcal{G}$.*

Proof: Note that the identity morphism $\mathtt{id}_A : A \to A$ is history-free. Thus, it suffices to prove that $\mathcal{G}_{\mathtt{hf}}$ is closed under composition.

Let $\sigma_f : A \to B$ and $\tau_g : B \to C$ be history-free winning strategies. Then, with notation as above, we need to show that: $\sigma_f ; \tau_g = \delta_h$. We prove that:

$$d \cdot e \in \delta_h \Longleftrightarrow d \cdot e \in \sigma_f ; \tau_g$$

The required result follows by a straightforward induction.

Consider

$$m'_k = ((f+g) \circ \mu)^k \circ (f+g) \circ \pi$$

The domains of definition of the partial functions m'_k are disjoint. Furthermore, we have

1. $d \cdot e \in \delta_h \Longleftrightarrow (\exists k)\, [m'_k(d) = e]$
2. $m'_{k+1}(d) = e \Longleftrightarrow (\exists e')\, [m'_k(d) = e' \wedge\ [(f+g) \circ \mu](e') = e]$

Let

$$T' = \{d \cdot b_1 \cdot b_2 \dots \cdot b_k \cdot e \mid (\exists d \cdot e \in \sigma_f ; \tau_g)\, [d \cdot b_1 \cdot b_2 \dots \cdot b_k \cdot e \succ d \cdot e]\}$$

The existence of T' is guaranteed by the proof of Proposition 1. Let T be the prefix closure of T'. We show by induction on k that

$$d_0 \cdot d_1 \cdot d_2 \dots \cdot d_k \in T \Longleftrightarrow m'_{k-1}(d_0) = d_k$$

For the base case, $k = 0$, let d_0 be an O-move in the component A (the case when d_0 is an O-move in the component C is symmetric)

$$\begin{aligned} m'_0(d_0) = d_1 &\Longleftrightarrow d_1 = f(d_0) \\ &\Longleftrightarrow d_0 \cdot d_1 \in \sigma_f \\ &\Longleftrightarrow d_0 \cdot d_1 \in T \end{aligned}$$

For the inductive case, suppose that d_k is a move in the B-component of $A \multimap B$ (the case when d_k is a move in the B component in $B \multimap C$ is symmetric). Then,

$$\begin{aligned} d_0 \cdot \dots \cdot d_{k+1} \in T &\Longleftrightarrow d \cdot \dots \cdot d_k \in T \wedge d_k \cdot d_{k+1} \in \tau_g \\ &\Longleftrightarrow m'_{k-1}(d_0) = d_k \wedge g(d_k) = d_{k+1} \\ &\Longleftrightarrow m'_k(d_0) = d_{k+1} \end{aligned}$$

We now prove the main result. Consider $d \cdot e \in P_{A \multimap C}$.

$$\begin{aligned} d_0 \cdot e \in \sigma_f ; \tau_g &\Longleftrightarrow (\exists k)\, (\exists d_1, \dots, d_k)\, [d_0 \cdot \dots \cdot d_k \cdot e \in T] \\ &\Longleftrightarrow (\exists k)\, [m'_k(d_0) = e] \\ &\Longleftrightarrow d \cdot e \in \delta_h . \end{aligned}$$

∎

3.5 $\star$-autonomous categories of games

3.5.1 $\mathcal{G}_{\mathtt{hf}}$ as a $\star$-autonomous category

We show that $\mathcal{G}_{\mathtt{hf}}$ is a $\star$-autonomous category, and thus yields an interpretation of the formulas and proofs of MLL + MIX. (For background, see [See89, Bar91]). We have already defined the object part of the tensor product $A \otimes B$, the linear negation $A^{\perp}$ and the tensor unit.

The action of tensor on morphisms is defined as follows. If $\sigma_f : A \to B$, $\tau_g : A' \to B'$, then $\sigma \otimes \tau : A \otimes A' \to B \otimes B'$ is induced by

$$\begin{aligned} h &= (M_A^+ + M_{A'}^+) + (M_B^- + M_{B'}^-) \cong (M_A^+ + M_B^-) + (M_{A'}^+ + M_{B'}^-) \\ &\stackrel{f+g}{\to} (M_A^- + M_B^+) + (M_{A'}^- + M_{B'}^+) \\ &\cong (M_A^- + M_{A'}^-) + (M_B^+ + M_{B'}^+) \end{aligned}$$

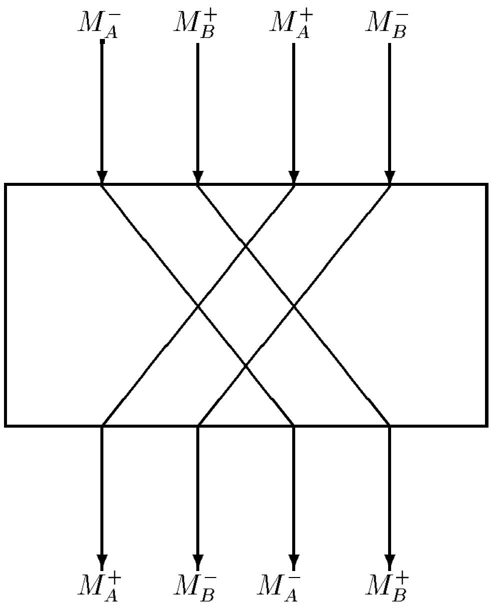
The natural isomorphisms for associativity, commutativity and unit of the tensor product are induced from those witnessing the symmetric monoidal structure of coproduct (disjoint union) in **Set**; say `assoc`, `symm`, `unit`. For example, the associativity of Tensor is given by $\sigma_h : (A \otimes B) \otimes C \cong A \otimes (B \otimes C)$, where

$$h : ((M_A^+ + M_B^+) + M_C^+) + (M_A^- + (M_B^- + M_C^-)) \cong ((M_A^- + M_B^-) + M_C^-) + (M_A^+ + (M_B^+ + M_C^+))$$

is the canonical isomorphism constructed from `assoc` and `symm`.

Similarly, the application morphism $\mathtt{apply} : (A \multimap B) \otimes A \to B$ is induced by

$$(M_A^- + M_B^+) + M_A^+) + M_B^- \cong (M_A^+ + M_B^-) + M_A^-) + M_B^+$$



This "message switching" function can be understood in algorithmic terms as follows. A demand for output from the application at M_B^- is switched to the function part of the input, $A \multimap B$; a demand by the function input for information about its input at M_A^- is forwarded to the input port A; a reply with this information about the input at M_A^+ is sent back to the function; an answer from the function to the original demand for output at M_B^+ is sent back to the output port B. Thus, this strategy does indeed correspond to a protocol for linear function application—linear in that the "state" of the inputs changes as we interact with them, and there are no other copies available allowing us to backtrack.

As for currying, given $\sigma_f : A\otimes B \to C$, where $f : (M_A^+ + M_B^+) + M_C^- \rightharpoonup (M_A^- + M_B^-) + M_C^+$, $\Lambda(\sigma) : A \to (B{\multimap}C)$ is induced by

$$M_A^+ + (M_B^+ + M_C^-) \cong (M_A^+ + M_B^+) + M_C^- \xrightarrow{f} (M_A^- + M_B^-) + M_C^+) \cong M_A^- + (M_B^- + M_C^+)$$

Finally, note that $A{\multimap}\bot \cong A^\bot$, where this isomorphism is induced by the bijection

$$(M_A^+ + \varnothing) + M_A^- \cong (M_A^- + \varnothing) + M_A^+$$

This yields $(A{\multimap}\bot){\multimap}\bot \cong A^{\bot\bot} = A$.

3.5.2 $\mathcal{G}$ as a $\star$-autonomous category

Proposition 4 *$\mathcal{G}$ is a $\star$-autonomous category; $\mathcal{G}_{\mathtt{hf}}$ is a sub-$\star$-autonomous category of $\mathcal{G}$.*

Proof: We first need to extend the definitions of $\sigma\otimes\tau$ and $\Lambda(\sigma)$ from $\mathcal{G}_{\mathtt{hf}}$ to $\mathcal{G}$. This is done as follows. Let $\sigma : A \to B$, $\tau : A' \to B'$. Then

$$\sigma\otimes\tau = \{s \in P_{A\otimes A' \multimap B\otimes B'} \mid s{\upharpoonright}A, B \in \sigma,\ s{\upharpoonright}A', B' \in \tau\}$$

We must establish that $\sigma\otimes\tau$ is well-defined and agrees with the definition in Section 3.5.1 for history-free strategies. Firstly, note that, if $s \cdot c \in \sigma\otimes\tau$ and c is an O-move:

$$\begin{array}{rcl} c \text{ in } A \text{ or } B\ \wedge (s\cdot c){\upharpoonright}A, B \cdot d \in \sigma & \Rightarrow & s\cdot c \cdot d \in \sigma\otimes\tau \\ c \text{ in } A' \text{ or } B'\ \wedge (s\cdot c){\upharpoonright}A', B' \cdot d \in \tau & \Rightarrow & s\cdot c \cdot d \in \sigma\otimes\tau \end{array}$$

Now, we show that if $s \in \sigma\otimes\tau$, and c is an O-move in A or B such that $s\cdot c \in P_{A\otimes A'\multimap B\otimes B'}$, then the unique d such that $s \cdot c\cdot d \in \sigma\otimes\tau$ is $\hat{\sigma}((s\cdot c){\upharpoonright}A, B)$; and similarly if c is in A' or B', with respect to τ. We argue by induction on $|s|$; *i.e.* we assume the required property for all proper prefixes of s. It suffices to show that, with the above notation, if c is in A or B, then d in A' or B' and $(s\cdot c\cdot d){\upharpoonright}A', B' \in \tau$ implies that d is an O-move in A' or B', and hence $s\cdot c\cdot d \notin \sigma\otimes\tau$. There are two cases: if $s{\upharpoonright}A', B' = \epsilon$, then d must be an initial move in τ and hence an O-move. Otherwise, applying the induction hypothesis to some proper prefix of s, the last O-move in A', B' in s must have had its response in A', B' in s and hence again it is Opponent to move in $s{\upharpoonright}A', B'$ according to τ.

Let $\sigma : A\otimes B \to C$. Then $\Lambda(\sigma) = \{\mathsf{assoc}^\star(s) \mid s \in \sigma\}$ where

$$\mathsf{assoc} : (M_A + M_B) + M_C \cong M_A + (M_B + M_C)$$

We omit the straightforward verification that this definition agrees with that of Section 3.5.1 on history-free strategies.

At this point, by Proposition 3 we only need to show that $\mathcal{G}$ is a $\star$-autonomous category. We do a sample calculation below to illustrate the proof.

Firstly, we prove a lemma which halves the work.

Lemma 1 *Winning strategies are incomparable under inclusion; if σ, τ are winning strategies in A, then $\sigma \subseteq \tau$ implies $\sigma = \tau$.*

Proof: Note that any winning strategy σ in A satisfies the following property: if $s \in \sigma$, O to move at s, then for all a such that $s\cdot a \in P_A$, there is a unique b such that $s\cdot a\cdot b \in \sigma$. Now, we prove by induction on $|s|$ that $s \in \tau \Rightarrow s \in \sigma$. The base case $s = \epsilon$ is clear. Now, suppose O is to move at $s \in \tau$, and consider any $s\cdot a \in P_A$. By induction hypothesis, $s \in \sigma$ and since σ, τ are winning, $s\cdot a\cdot b' \in \sigma$ and $s\cdot a\cdot b'' \in \tau$, for unique b', b''. Since $\sigma \subseteq \tau$, $s\cdot a\cdot b'' \in \tau$ and $b' = b''$. Thus, $s\cdot a\cdot b'' \in \sigma$. ∎

Let $\sigma : A \otimes B \to C$. We show that the following diagram commutes (different subscripts have been used on B, C to distinguish the different occurrences)

$$\begin{tikzcd}
(B_3 \multimap C_1) \otimes B_2 \arrow[r, "\mathtt{apply}"] & C_2 \\
A \otimes B_1 \arrow[u, "\Lambda(\sigma) \otimes \mathtt{id}_B"] \arrow[ur, "\sigma"'] &
\end{tikzcd}$$

From the definitions,

$$\Lambda(\sigma) \otimes \mathtt{id}_B ; \mathtt{apply} = \{ s \upharpoonright A, B_1, C_2 \mid s \in S \}$$

where

$$\begin{array}{rcl}
S & = & \{ s \in \mathcal{L}(A \otimes B_1, (B_3 \multimap C_1) \otimes B_2, C_2) \mid \\
& & \quad s \upharpoonright A, B_1, B_2, B_3, C_1 \in P_{A \otimes B_1 \multimap (B_3 \multimap C_1) \otimes B_2}, \\
& & \quad s \upharpoonright B_2, B_3, C_1, C_2 \in P_{(B_3 \multimap C_1) \otimes B_2 \multimap C_2} \\
& & \quad s \upharpoonright A, B_3, C_1 \in \sigma \\
& & \quad s \upharpoonright B_1, B_2 \in \mathtt{id}_B, \ s \upharpoonright B_2, B_3 \in \mathtt{id}_B \\
& & \quad s \upharpoonright C_1, C_2 \in \mathtt{id}_C \}
\end{array}$$

We shall define a map h such that, for all $s \in \sigma$, $h(s) \in S$ and $h(s) \upharpoonright A, B_1, C_2 = s$. This will show that $\sigma \subseteq \Lambda(\sigma) \otimes \mathtt{id}_B ; \mathtt{apply}$, and hence the desired equation by the above lemma.

We define h as the unique monoid homomorphism extending the following assignment:

O-moves: $a \mapsto a, \ b \mapsto b_1 \cdot b_2 \cdot b_3, \ c \mapsto c_2 \cdot c_1$

P-moves: $a \mapsto a, \ b \mapsto b_3 \cdot b_2 \cdot b_1, \ c \mapsto c_1 \cdot c_2$

It is clear that for all $s \in \sigma$, $h(s)$ has the following properties:

1. $h(s) \upharpoonright A, B_1, C_2 = s$
2. $h(s) \upharpoonright B_1 = h(s) \upharpoonright B_2 = h(s) \upharpoonright B_3$
3. $h(s) \upharpoonright C_1 = h(s) \upharpoonright C_2$
4. $|s|$ even $\Rightarrow$ last move in $h(s)$ in A, B_1 or C_2

It remains to show that $h(s) \in S$. Clearly, (2) applied to all prefixes of s implies that $h(s) \upharpoonright B_1, B_2 \in \mathtt{id}_B$ and $h(s) \upharpoonright B_2, B_3 \in \mathtt{id}_B$. Similarly, (3) implies that $h(s) \upharpoonright C_1, C_2 \in \mathtt{id}_C$. Also, (1), (2) and (3) and $s \in \sigma$ implies that $h(s) \upharpoonright A, B_3, C_1 \in \sigma$.

Now, let $t = h(s) \upharpoonright A, B_1, B_2, B_3, C_1$, $T = P_{A \otimes B_1 \multimap (B_3 \multimap C_1) \otimes B_2}$. We will show that $t \in T$, by induction on $|s|$. For the key case, suppose Opponent to move at s. Let $s \cdot d \cdot e \in \sigma$. We now consider the various subcases according to the locations of d and e. For example, suppose $d = b$ is in B, and $e = c$ is in C. Then $h(s \cdot b \cdot c) = h(s) \cdot b_1 \cdot b_2 \cdot b_3 \cdot c_1 \cdot c_2$ and $h(s \cdot b \cdot c) \upharpoonright A, B_1, B_2, B_3, C_1 = t \cdot b_1 \cdot b_2 \cdot b_3 \cdot c_1$. By induction hypothesis, $t \in T$. By (1), (4) and $s \cdot d \in \sigma$, $t \cdot b_1 \in T$. Using (2), $t \cdot b_1 \cdot b_2 \cdot b_3 \in T$. Using (1), (3) and $s \cdot d \cdot e \in \sigma$, we get the required result. A similar argument shows that $h(s) \upharpoonright B_2, B_3, C_1, C_2 \in P_{(B_3 \multimap C_1) \otimes B_2 \multimap C_2}$. Also, note that if $s \cdot d \cdot e \in \sigma$, where d is in A or B and e is in C, then e must be a P-move; similarly, if d is in C and e is in A or B. It then easily follows, by induction on $|s|$, that $h(s) \in \mathcal{L}(A \otimes B_1, (B_3 \multimap C_1) \otimes B_2, C_2)$.

We also verify the unicity equation $\Lambda(\tau \otimes \mathtt{id}_B; \mathtt{apply}) = \tau$, where $\tau : A \to (B \multimap C)$. We define $\Lambda^{-1}(\tau) = \{(\mathtt{assoc}^{-1})^\star(s) \mid s \in \tau\}$. Clearly, $\Lambda(\Lambda^{-1}(\tau)) = \tau$ and $\Lambda^{-1}(\tau) : A \otimes B \to C$. Now,

$$\begin{aligned}
\Lambda(\tau \otimes \mathtt{id}_B; \mathtt{apply}) &= \Lambda(\Lambda(\Lambda^{-1}(\tau)) \otimes \mathtt{id}_B; \mathtt{apply}) \\
&= \Lambda(\Lambda^{-1}(\tau)) \\
&= \tau.
\end{aligned}$$

∎

3.6 Variable types and uniform strategies

An *embedding* $e : A \rightarrowtail B$ is a 1–1 map $e : M_A \to M_B$ such that

(e1) $\lambda_B \circ e = \lambda_A$

(e2) $e^\star(P_A) \subseteq P_B$

(e3) $(\forall s \in P_A^\infty)\, [s \in W_A \iff e^\omega(s) \in W_B]$

where $e^\star, e^\omega$ are the canonical extensions of e to $M_A^\star, M_A^\omega$ respectively. We write $\mathcal{G}^e$ for the evident category of games and embeddings. Note that given an embedding $e : A \rightarrowtail B$, we can derive functions $e^- : M_A^- \rightarrowtail M_B^-$ and $e^+ : M_A^+ \rightarrowtail M_B^+$.

Proposition 5 *Tensor, Par and Involution can be extended to covariant functors over $\mathcal{G}^e$.*

Proof: If $e : A \rightarrowtail B$, $e' : A' \rightarrowtail B'$, then $e \otimes e' = e + e'$ and $e^\perp = e$. We just check the only non-obvious part, namely that condition (e3) is satisfied by $e^\perp$. Given $s \in P_{A^\perp}^\infty = P_A^\infty$,

$$\begin{aligned}
s \in W_{A^\perp} &\iff s \in P_A^\infty \setminus W_A \\
&\iff s \in P_A^\infty,\ e^\omega(s) \notin W_B \\
&\iff s \in P_{A^\perp}^\infty,\ e^\omega(s) \in W_{B^\perp}
\end{aligned}$$

Thus, $s \in W_{A^\perp} \iff e^\omega(s) \in W_{B^\perp}$. ∎

Now, given a multiplicative formula A with propositional atoms $\alpha_1, \ldots, \alpha_n$, this induces a functor $F_A : (\mathcal{G}^e)^n \to \mathcal{G}^e$. Similarly, a sequent $\Gamma(\alpha_1, \ldots, \alpha_n)$ induces a functor $F_\Gamma : (\mathcal{G}^e)^n \to \mathcal{G}^e$ (where Γ is interpreted as $⅋\Gamma$).

A strategy for $\Gamma(\alpha_1, \ldots, \alpha_n)$ will be a family $\{\sigma_{\vec{A}}\}$, where for each n-tuple of games $\vec{A}$, $\sigma_{\vec{A}}$ is a strategy in $F_\Gamma(\vec{A})$. We express the uniformity of this family by a naturality condition. Given $F : (\mathcal{G}^e)^n \to \mathcal{G}^e$, we define two functors $F^-, F^+ : (\mathcal{G}^e)^n \to \mathtt{Set}^p$, where $\mathtt{Set}^p$ is the category of sets and partial functions.

$$F^-(\vec{A}) = M^-_{F(\vec{A})} \qquad\qquad F^-(\vec{e}) = F(\vec{e})^-$$
$$F^+(\vec{A}) = M^+_{F(\vec{A})} \qquad\qquad F^+(\vec{e}) = F(\vec{e})^+$$

If $\sigma = \{\sigma_{\vec{A}}\}$ is a family of history free strategies, then each $\sigma_{\vec{A}}$ is of the form $\sigma_{f_{\vec{A}}}$. So we get a family of partial functions $\{f_{\vec{A}}\}$ where $f_{\vec{A}} : M^-_{F(\vec{A})} \to M^+_{F(\vec{A})}$, *i.e.* $f_{\vec{A}} : F^-(\vec{A}) \to F^+(\vec{A})$. We say that σ is *uniform* if f is a natural transformation $f : F^- \to F^+$.

Now, for each $n \in \omega$, we can define a category $\mathcal{G}_{\mathtt{hf}}(n)$, whose objects are functors $F : (\mathcal{G}^e)^n \to \mathcal{G}^e$ and whose morphisms $\sigma : F \to G$ are uniform, history-free winning strategies $\{\sigma_{\vec{A}}\}$, where $\sigma_{\vec{A}} : F(\vec{A}) \to G(\vec{A})$, *i.e.* $\sigma_{\vec{A}}$ is a strategy in $F(\vec{A}) \multimap G(\vec{A})$. Composition is pointwise: if $\sigma : F \to G$, $\tau : G \to H$, then $(\sigma; \tau)_{\vec{A}} = \sigma_{\vec{A}}; \tau_{\vec{A}}$. Note that $\mathcal{G}_{\mathtt{hf}}(0) \cong \mathcal{G}_{\mathtt{hf}}$.

Proposition 6 *For each n, $\mathcal{G}_{\mathtt{hf}}(n)$ is a $\star$-autonomous category; $\mathcal{G}_{\mathtt{hf}} : \mathbb{B} \to \star$-**Aut** is an indexed $\star$-autonomous category with base $\mathbb{B}$, the category of finite ordinals and set maps.*

Proof: The $\star$-autonomous structure on $\mathcal{G}_{\mathtt{hf}}(n)$ is defined pointwise from that on $\mathcal{G}_{\mathtt{hf}}$, *e.g.* $(F \otimes G)(\vec{A}) = F(\vec{A}) \otimes G(\vec{A})$.

We will show that composition preserves uniformity. Given functions f, g as in Section 3.4.1, we write $\mathtt{EX}(f, g)$ for the execution formula applied to f, g. Now, if $\sigma : F \to G$, $\tau : G \to H$, $\sigma = \sigma_f$ and $\tau = \tau_g$, and $\vec{e} : \vec{A} \rightarrowtail \vec{B}$, we must show that

$$\begin{array}{ccc} M^-_{F(\vec{A}) \multimap H(\vec{A})} & \xrightarrow{\mathtt{EX}(f_{\vec{A}}, g_{\vec{A}})} & M^+_{F(\vec{A}) \multimap H(\vec{A})} \\ {\scriptstyle (F \multimap H)(\vec{e})^-} \Big\downarrow & & \Big\downarrow {\scriptstyle (F \multimap H)(\vec{e})^+} \\ M^-_{F(\vec{B}) \multimap H(\vec{B})} & \xrightarrow[\mathtt{EX}(f_{\vec{B}}, g_{\vec{B}})]{} & M^+_{F(\vec{B}) \multimap H(\vec{B})} \end{array}$$

Writing $\mathtt{EX}(f_{\vec{A}}, g_{\vec{A}}) = \bigvee_{k \in \omega} m^{\vec{A}}_k$ where $m^{\vec{A}}_k = \pi^\star_{\vec{A}} \circ ((f_{\vec{A}} + g_{\vec{A}}) \circ \mu_{\vec{A}})^k \circ (f_{\vec{A}} + g_{\vec{A}}) \circ \pi_{\vec{A}}$, we must show that

$$(F(\vec{e})^- + H(\vec{e})^+) \circ \bigvee_{k \in \omega} m^{\vec{A}}_k = \bigvee_{k \in \omega} m^{\vec{B}}_k \circ (F(\vec{e})^+ + H(\vec{e})^-)$$

Since composition distributes over joins, it suffices to show that for all k,

$$(F(\vec{e})^- + H(\vec{e})^+) \circ m^{\vec{A}}_k = m^{\vec{B}}_k \circ (F(\vec{e})^+ + H(\vec{e})^-) \tag{1}$$

Note firstly that

$$\begin{aligned} (F(\vec{e})^- + H(\vec{e})^+) \circ \pi^\star_{\vec{A}} &= \pi^\star_{\vec{B}} \circ (F(\vec{e})^- + G(\vec{e})^+ + G(\vec{e})^- + H(\vec{e})^+) \\ \pi_{\vec{B}} \circ (F(\vec{e})^+ + H(\vec{e})^-) &= (F(\vec{e})^+ + G(\vec{e})^- + G(\vec{e})^+ + H(\vec{e})^-) \circ \pi_{\vec{A}} \\ (F(\vec{e})^+ + G(\vec{e})^- + G(\vec{e})^+ + H(\vec{e})^-) \circ \mu_{\vec{A}} &= \mu_{\vec{B}} \circ (F(\vec{e})^- + G(\vec{e})^+ + G(\vec{e})^- + H(\vec{e})^+) \end{aligned}$$

and by uniformity of f and g

$$(f_{\vec{B}} + g_{\vec{B}}) \circ (F(\vec{e})^+ + G(\vec{e})^- + G(\vec{e})^+ + H(\vec{e})^-) = (F(\vec{e})^- + G(\vec{e})^+ + G(\vec{e})^- + H(\vec{e})^+) \circ (f_{\vec{A}} + g_{\vec{A}})$$

A straightforward induction on k using these equations establishes (1).

The uniformity of the morphisms witnessing the $\star$-autonomous structure on $\mathcal{G}_{\mathtt{hf}}(n)$ follows directly from the naturality of the canonical isomorphisms for coproduct in $\mathtt{Set}$ from which they are defined.

Given $f : \{1, \ldots, n\} \to \{1, \ldots, m\}$ (where we take the liberty of representing the ordinal n by $\{1, \ldots, n\}$), we define

$$\begin{aligned} \mathcal{G}_{\mathtt{hf}}(f)(F)(A_1, \ldots, A_m) &= F(A_{f(1)}, \ldots, A_{f(n)}) \\ \mathcal{G}_{\mathtt{hf}}(f)\{\sigma_{A_1, \ldots, A_n}\} &= \{\sigma_{A_{f(1)}, \ldots, A_{f(n)}}\} \end{aligned}$$

The verification that $\mathcal{G}_{\mathtt{hf}}(f)$ is a $\star$-autonomous functor is straightforward from the pointwise definition of the $\star$-autonomous structure on $\mathcal{G}_{\mathtt{hf}}(n)$. The functoriality of $\mathcal{G}_{\mathtt{hf}}$ itself is a routine calculation. ∎

Using this Proposition, we can interpret proofs in MLL + MIX by uniform, history-free strategies; see [See89] for further details. This is the semantics for which Full Completeness will be proved.

4 Full Completeness

In this section, we prove full Completeness of the game semantics for MLL + MIX. The proof is structured into a number of steps.

- Firstly, we show that a uniform, history free winning strategy for Γ induces a proof structure on Γ.
- Next, we reduce the problem to that for *binary* sequents, in which each atom occurring does so once positively and once negatively.
- We then make a further reduction to *simple* binary sequents, in which every formula is either a literal, or the tensor product of two literals.
- Finally, we show that for such sequents, there can only be a winning strategy if the corresponding proof structure satisfies the correctness criterion, *i.e.* is a proof net.

4.1 Strategies induce Axiom links

We begin by establishing some notation. We are given an MLL sequent $\Gamma(\alpha_1, \ldots, \alpha_k)$ where $\alpha_1, \ldots, \alpha_k$ are the propositional atoms occurring in Γ. We enumerate the occurrences of literals in Γ as $c_1, \ldots c_n$; each c_i is an occurrence of l_i, where $l_i = \alpha_{j_i}$ or $l_i = \alpha_{j_i}^{\perp}$ for some j_i, $1 \leq i \leq n, 1 \leq j_i \leq k$. Given any sequence $\vec{A} = A_1, \ldots, A_k$ of games instantiating $\alpha_1, \ldots, \alpha_k$, we obtain a game $F(\vec{A})$, where $F = F_\Gamma$ is the interpretation of $\parr\Gamma$. Note that $M_{F(\vec{A})} = \sum_{i=1}^{n} M_{C_i}$, where $C_i = A_{j_i}$ or $A_{j_i}^{\perp}$. We represent $M_{F(\vec{A})}$ concretely as $\cup_{i=1}^{n} \{i\} \times M_{C_i}$. We refer to the C_i as the *constituents* of $M_{F(\vec{A})}$.

Proposition 7 *With notation as above, let $\sigma = \{\sigma_A\}$ be a uniform history free winning strategy for $F = F_\Gamma$. Then, for some involution ϕ such that (Γ, ϕ) is a proof structure, for all $\vec{A}$,*

$$\sigma_{\vec{A}} = \sigma_{f_{\vec{A}}}$$

where $f_{\vec{A}}((i, a)) = (\phi(i), a)$.

Proof: A game A is *full* if $P_A = M_A^{\circledast}$. Given any game A, there is an embedding $e_A^{\mathtt{full}} : A \rightarrowtail A^{\mathtt{full}}$, where $A^{\mathtt{full}} = (M_A, \lambda_A, M_A^{\circledast}, W_A)$ and $e_A^{\mathtt{full}} = \mathtt{id}_{M_A}$.

By uniformity,

$$\begin{array}{ccc} F^-(\vec{A}) & \xrightarrow{F^-(e_{\vec{A}}^{\mathtt{full}})} & F^-(\vec{A}^{\mathtt{full}}) \\ \Big\downarrow f_{\vec{A}} & & \Big\downarrow f_{\vec{A}^{\mathtt{full}}} \\ F^+(\vec{A}) & \xrightarrow[F^+(e_{\vec{A}}^{\mathtt{full}})]{} & F^+(\vec{A}^{\mathtt{full}}) \end{array}$$

But $F^-(e_A^{\mathtt{full}}) = \mathtt{id}_{M^-_{F(A)}}$, $F^+(e_A^{\mathtt{full}}) = \mathtt{id}_{M^+_{F(A)}}$. Hence $f_{\vec{A}} = f_{\vec{A}^{\mathtt{full}}}$. Thus, it suffices to prove the Proposition for full games.

Let $i \in \{i, \ldots, n\}$ and $a \in M_{C_i}^-$. Thus, (i, a) is an O-move in the i'th constituent of $F(\vec{A})$. Consider the vector $\vec{B}$, where the i'th constituent is instantiated with

$$B = (\{b\}, \{(b, O)\}, \{\epsilon, b\}, \varnothing),$$

all constituents labelled with the same literal by B, all constituents labelled with the dual literal by $B^\perp$, all other constituents with the empty game. Since $\sigma_{\vec{B}}$ is winning, we must have $f_{\vec{B}}((i,b)) = (j,b)$, for some constituent j with dual label to that of i.

Now there is an embedding from B to A_{j_i}, hence from $\vec{B}$ to $\vec{A}$, sending b to a. By uniformity, this implies that $f_{\vec{A}}((i,a)) = (j,a)$. Note that this will apply to *all* (i,a') for the given i, so all O-moves in the i'th constituent are mapped to the *same* fixed constituent j. Thus, we can define an endofunction ϕ on $\{1,\ldots,n\}$ such that, for all full $\vec{A}$, and hence for all $\vec{A}$, for all $i \in \{1,\ldots,n\}$, $a \in M^-_{A_{j_i}}$, $f_{\vec{A}}((i,a)) = (\phi(i),a)$. Moreover, $l_i = l^\perp_{\phi(i)}$, so in particular ϕ is fixpoint free.

It only remains to be shown that ϕ is an involution. Consider the game

$$C = (\{a',b'\},\{(a',O),(b',P)\},\{\epsilon,a',a'\cdot b'\},\varnothing)$$

Consider the instance $\vec{C}$ defined similarly to $\vec{B}$, with C used in place of B. We already know that $f_{\vec{C}}((i,a')) = (\phi(i),a')$. Since $\sigma_{\vec{C}}$ is winning, we must have $f_{\vec{C}}((\phi(i),b')) = (i,b')$. So $\phi^2(i) = i$, and ϕ is an involution as required. ∎

Corollary 1 *If there is a uniform history-free winning strategy for $F = F_\Gamma$, then Γ must be balanced,* i.e. *each atom must occur the same number of times positively as negatively.*

Proof: The function ϕ of Proposition 7 establishes a bijection between positive and negative occurrences of each atom. ∎

4.2 Reduction to binary sequents

Let σ be a history free strategy for a proof structure (Γ,ϕ). We define a binary sequent Γ_ϕ by relabelling each pair of literals as specified by ϕ with distinct atoms. Note that a binary sequent has a unique associated proof structure; so the involution is redundant in this case. It is clear from the definition of the correctness criterion that

$$(\Gamma,\phi) \text{ is a proof net} \iff \Gamma_\phi \text{ is a proof net}$$

Now given a proof structure (Γ,ϕ), the corresponding uniform, history-free strategy $\sigma_{(\Gamma,\phi)}$ for Γ is defined by

$$\sigma_{(\Gamma,\phi)} = \sigma_{f_{(\Gamma,\phi)}}, \text{ where } f_{(\Gamma,\phi),\vec{A}}((i,a)) = (\phi(i),a)$$

Proposition 8 *Let (Γ,ϕ) be a proof structure.*

$$\sigma_{(\Gamma,\phi)} \textit{ is winning for } \Gamma \iff \sigma_{\Gamma_\phi} \textit{ is winning for } \Gamma_\phi$$

Proof: Since every instance of Γ is an instance of Γ_ϕ, the right to left implication is clear.

For the converse, given an instance $\vec{A}$ for Γ_ϕ, consider the following instance for Γ: for each α occurring k times positively in Γ, with $A_{j_1},\ldots,A_{j_k}$ instantiating these occurrences in $\vec{A}$, instantiate α with the disjoint union $A_{j_1}+\cdots+A_{j_k}$. Since $\sigma_{(\Gamma,\phi)}$ is winning by assumption, it defeats every play by Opponent, in particular those plays in which Opponent plays only in A_{j_i} in the game instantiating the i'th occurrence of α. This shows that σ_{Γ_ϕ} is winning as required. ∎

4.3 Reduction to simple sequents

Let Γ be a binary sequent. We write $\Gamma = C[A]$, where $C[\cdot]$ is a monotone context, *i.e.* with the "hole" $[\cdot]$ appearing only under the scope of Tensors and Pars. For such a context, we have

$$A \multimap B \vdash C[A] \multimap C[B]$$

Lemma 2 *Let* $\Gamma = C[A \otimes (B \parr C)]$ *be a binary sequent. Let* $\Gamma_1 = C[(A \otimes B) \parr C]$ *and* $\Gamma_2 = C[(A \otimes C) \parr B]$ *. Then*

1. $(\forall i)\ \vdash \Gamma \multimap \Gamma_i$
2. $\vdash \Gamma \iff (\forall i)\ \vdash \Gamma_i$

Proof:

1. $A \otimes (B \parr C) \multimap (A \otimes B) \parr C$ and $A \otimes (B \parr C) \multimap (A \otimes C) \parr B$ are both theorems of MLL.
2. We use the correctness criterion. Suppose Γ is not provable, *i.e.* for some switching S, $G(C[A \otimes (B \parr C)], S)$ has a cycle. If S sets the indicated par link to L, there will be a cycle in Γ_1; if S sets the indicated par link to R, there will be a cycle in Γ_2. ∎

Lemma 3 *Let* $\Gamma = C[A \otimes (B \otimes C)]$ *be a binary sequent. Let* $\Gamma_1 = C[A \otimes (B \parr C)]$, $\Gamma_2 = C[A \parr (B \otimes C)]$. *Then,*

1. $(\forall i)\ \vdash \Gamma \multimap \Gamma_i$
2. $\vdash \Gamma \iff (\forall i)\ \vdash \Gamma_i$

Proof:

1. $\alpha \otimes \beta \multimap \alpha \parr \beta$ is a theorem of MLL + MIX.
2. We use the correctness criterion. Suppose Γ is not provable, *i.e.* for some switching S $G(\Gamma, S)$ has a cycle. In particular fix some *simple* cycle in $G(\Gamma, S)$ (*i.e.* no internal node is visited more than once). This implies that the cycle cannot visit all of the A, B, C edges. Thus, there are four possible cases:
   - The cycle does not visit $A \otimes (B \otimes C)$ at all. Then clearly both Γ_1, Γ_2 have cycles.
   - The cycle visits the A and B edges: Then $G(\Gamma_1, S')$ has a cycle, where S' sets the switch of the new Par node to L, and otherwise is defined like S.
   - The cycle visits the A and C edges: Symmetric to the previous case.
   - The cycle visits the B and C edges: Then $G(\Gamma_2, S')$ has a cycle, where S' sets the switch of the new Par node to R, and otherwise is defined like S. ∎

Proposition 9 *Let* Γ *be a binary sequent. Then there is a set of* simple *binary sequents* $\Gamma_1, \ldots, \Gamma_n$ *such that:*

1. $(\forall i)\ \vdash \Gamma \multimap \Gamma_i$
2. $\vdash \Gamma_\phi \iff (\forall i)\ \vdash \Gamma_i$

Proof: Firstly, use Lemma 2 repeatedly to push all Pars to the top and then replace them by commas. Then, given a nested occurrence of Tensor, we can use Lemma 3 to replace it with a Par, and use Lemma 2 again to eliminate this Par. In this way, we eventually reach a set of *simple* binary sequents. ∎

4.4 Winning strategies are acyclic

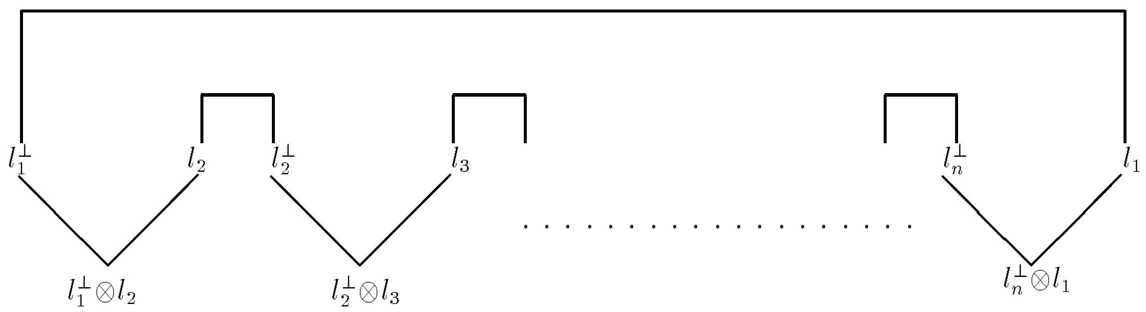
We now establish the crucial connection between winning strategies and the correctness criterion for proof nets.

Proposition 10 *Let* Γ *be a* simple *binary sequent. Let* σ_Γ *be the associated uniform history free strategy as in Proposition 8. If* σ_Γ *is winning, then* Γ *is acyclic.*

Proof: Suppose Γ has a cycle. Since Γ is simple, this is necessarily of the form

$$l_1^\perp, \otimes, l_2, l_2^\perp, \otimes, \ldots, l_n, l_n^\perp, \otimes, l_1$$

For example:



(This picture is not completely general; non-planar arrangements are possible. However, this will not play any role in the argument).

We will assign games $\vec{A}$ to atoms in Γ in such a way that Opponent has a winning strategy in $F_\Gamma(\vec{A})$, thus showing that there can be no uniform winning strategy for Γ.

We label the literals $l_1^\perp, l_2, l_2^\perp, \ldots, l_n, l_n^\perp, l_1$ alternately *tt* and *ff*. We define $\vec{A}$ such that each literal labelled *tt* is assigned

$$(\{a, b\}, \{(a, P), (b, O)\}, \{(a \cdot b)^\star \cdot (\epsilon + a)\}, \{(a \cdot b)^\omega\})$$

and each literal labelled *ff* is assigned

$$(\{a, b\}, \{(a, O), (b, P)\}, \{(a \cdot b)^\star \cdot (\epsilon + a)\}, \varnothing)$$

and all unlabelled literals are assigned the empty game.

We now describe the strategy for Opponent. Note that by assumption, Player is following the strategy σ_Γ, so his response to Opponent's moves is determined a priori.

Consider the following play:

O plays a in l_1
P plays a in $l_1^\perp$
O plays a in l_2
P plays a in $l_2^\perp$
$\vdots$
O plays a in l_n
P plays a in $l_n^\perp$
O plays b in $l_n^\perp$
P plays b in l_n
O plays b in $l_{n-1}^\perp$
P plays b in l_{n-1}
$\vdots$
O plays b in $l_1^\perp$
P plays b in l_1

Now Opponent cycles repeatedly through this whole sequence of moves, forcing the infinite play with projection $(a \cdot b)^{\omega}$ on all labelled literals, and empty projection on all unlabelled literals.

Note that Opponent is able to make the indicated moves, since it always responds in the same component as the last move of Player, or switches components in a tensor sub-game.

Clearly, Player loses in each Tensor sub-game of the cycle, since it loses in each component labelled *ff*. But then Player loses in Γ, since to win an infinite play in a Par, it must win an infinite play in some component, and all components of $⅋\Gamma$ either have empty plays, or infinite plays with Player losing. ∎

Note that infinite plays, *i.e.* the use of non-well-founded games, are essential to this argument.

4.5 Main result

Theorem 1 *(Full Completeness)*
If σ is a uniform history-free winning strategy for Γ, then it is the denotation of a unique proof net (Γ, ϕ).

Proof: By Proposition 7, we know that there is a unique proof structure (Γ, ϕ) with $\sigma = \sigma_{(\Gamma,\phi)}$. It remains to show that (Γ, ϕ) is a proof net. By Proposition 8, $\sigma_{(\Gamma,\phi)}$ winning implies σ_{Γ_ϕ} winning. Applying Proposition 9 to Γ_ϕ, there is a set of simple binary sequents $\Gamma_1, \ldots, \Gamma_n$ such that

1. $(\forall i) \ \vdash \Gamma_\phi \multimap \Gamma_i$
2. $\vdash \Gamma_\phi \iff (\forall i) \ \vdash \Gamma_i$

Since the game semantics is sound, (1) and the validity of Γ_ϕ in the game semantics implies that there is a uniform, history-free winning strategy for each Γ_i. By Proposition 7, this strategy is necessarily of the form σ_{Γ_i}. By Proposition 10, this implies that each Γ_i is acyclic. By (2), this implies that Γ_ϕ is a proof net. By the remark before Proposition 8, this implies that (Γ, ϕ) is a proof net. ∎

5 Beyond the multiplicatives

Up to this point, we have only considered the multiplicative fragment of Linear Logic. However, our game semantics in fact yields a categorical model of full second-order (or even ω-order) Classical Linear Logic. In this section, we will outline the interpretation of the additives and exponentials. A detailed treatment of this material, and of the game semantics for the second-order quantifiers, will be given in a sequel to the present paper.

5.1 Polarities

To proceed, we focus on the fact that our games may admit some positions in which Player starts, some in which Opponent starts.

Definition 4 *A game A is* positive *(has polarity +1) if every valid initial move in A is by Player;* negative *(has polarity -1) if every valid initial move in A is by Opponent; and* neutral *(polarity 0) otherwise.*

Although we use the same notation for polarities as Girard [Gir91b], they have a somewhat different interpretation. Our polarities have a very direct computational reading. If we interpret moves by Opponent as demands for data and moves by Player as generating data, then *positive* games model purely data-driven computation; *negative* games model purely demand-driven computation; while *neutral* games allow both modes of computation. These notions give rise to the following situation. We have full subcategories

$$I^- : \mathcal{G}^- \hookrightarrow \mathcal{G} \hookleftarrow \mathcal{G}^+ : I^+$$

of positive and negative games. There are evident constructions A^+ (A^-) taking a game A in $\mathcal{G}$ to $\mathcal{G}^+$ ($\mathcal{G}^-$) simply by deleting all positions of P_A starting with a move by Opponent (Player) and correspondingly pruning W_A.

Proposition 11

- *$\mathcal{G}^+$ is reflective and $\mathcal{G}^-$ is co-reflective in $\mathcal{G}$, with $I^- \dashv (\cdot)^-$, $(\cdot)^+ \dashv I^+$.*
- *Linear negation $(\cdot)^\perp$ cuts down to a duality $\mathcal{G}^- \simeq \mathcal{G}^{+\mathrm{op}}$; in fact $(A^-)^\perp = (A^\perp)^+$, $(A^+)^\perp = (A^\perp)^-$.*

5.2 Exponentials

Jacobs has recently investigated the decomposition of the exponentials !, ? into weakening parts $!_w$, $?_w$ and contraction parts $!_c$, $?_c$ [Jac92]. He develops a general theory for this decomposition. We will use a little of this theory to structure our presentation of the exponentials.

5.2.1 Weakening

The reflection and co-reflection of Proposition 11 give rise to a monad and a comonad on $\mathcal{G}$ respectively, which we denote by $?_w$ and $!_w$. Our reason for this notation is explained by the following proposition.

Proposition 12 *There are natural transformations*

$$!_w A \otimes B \rightarrow B, \quad B \rightarrow ?_w A ⅋ B$$

As a consequence of this proposition, the following weakening rule is valid in the game semantics.

$$\frac{\vdash \Gamma}{\vdash \Gamma, ?_w A}$$

5.2.2 Exponentials

We want to define $!A$ as the type of objects which are copyable versions of objects of type A. We achieve copyability by *backtracking*; *cf.* [AV93]. That is, at any stage in a play in $!A$, the Opponent may return to a previous stage to make his move. In this way, a single play in $!A$ will correspond to a tree of plays in A.

Definition 5 *$!A$ is defined as follows:*

- $M_{!A} = M_A^+ \cup (\omega \times M_A^-)$
- $\lambda_{!A}(a) = P, \quad \lambda_{!A}((i,a)) = O$
- *Define*
  - $s(i) = s_1 \cdots s_i$, $s \backslash i = s_1 \cdots s_{|s|-i}$

– $\overline{(\ )} : M^{\star}_{!A} \to M^{\star}_{!A}$ *by* $\overline{\epsilon} = \epsilon$, $\overline{s \cdot a} = \overline{s} \cdot a$, $\overline{s \cdot (i, a)} = \overline{(s \backslash i)} \cdot a$

– $\hat{s} = \{\overline{s(i)} \mid 0 \leqslant i \leqslant |s|\}$

Also, a partial strategy *is defined like a strategy except that it need not satisfy* **(s3)**. *Then,*

$$P_{!A} = \{s \in M^{\circledast}_{!A} \mid (\forall j : 1 \leqslant j \leqslant |s|)\ s_j = (i, a) \Rightarrow i < j,\ \hat{s} \textit{ is a partial strategy in } A\}.$$

- *Given* $s \in P^{\infty}_{!A}$, *let* $\check{s}$ *be the set of all* $t \in P^{\infty}_{A}$ *such that every finite prefix of* t *is* $\overline{s(i)}$ *for some* $i \in \omega$. *Then,*

$$W_{!A} = \{s \in P^{\infty}_{!A} \mid \check{s} \subseteq W_A\}$$

Proposition 13 ! *is a comonad on* $\mathcal{G}$, *satisfying* $! = ! \circ !_w = !_w \circ !$. *Moreover,* ! *has a natural commutative comonoid structure on its free algebras,* i.e. *maps*

$$\delta_A : !A \to !A \otimes !A$$

such that !- *algebra morphisms between its free algebras are automatically comonoid homomorphisms.*

As a consequence of this proposition, the contraction rule is valid in the game semantics:

$$\frac{\vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A}$$

where ? is the monad defined by duality from !: $?A = (!A^{\perp})^{\perp}$.

5.3 Additives

The additives of Linear Logic are problematic. This is seen in various ways: by the difficulties of getting a "reasonable" implementation (for example, in terms of interaction nets) of the commutative conversions for the additives [GAL92]; and, most conspicuously, by the problems they engender with the Geometry of Interaction [Gir89b, Gir89a, Gir88].

Our notion of polarities throws some light on these matters and suggests a refinement of Linear Logic which may allow these problems to be addressed.

Proposition 14 $\mathcal{G}^{+}$ *has coproducts, and* $\mathcal{G}^{-}$ *has products, both defined by disjoint union of games.*

These definitions can be extended to get weak products and coproducts on $\mathcal{G}$, defined as follows.

$$\begin{array}{rcl}
M_{A\&B} & = & M_A + M_B + \{*, l, r\} \\
\lambda_{A\&B} & = & [\lambda_A, \lambda_B, \{(*, P), (l, O), (r, O)\}] \\
P_{A\&B} & = & \text{prefix closure of } (P_{A^-} + P_{B^-}) \cup (* \cdot l \cdot P_{A^+} + * \cdot r \cdot P_{B^+}) \\
W_{A\&B} & = & (W_{A^-} + W_{B^-}) \cup (* \cdot l \cdot W_{A^+} + * \cdot r \cdot W_{B^+})
\end{array}$$

Note that $(A\&B)^{-} = A^{-} + B^{-}$ (disjoint union of games), *i.e.* the weak product in $\mathcal{G}$ is carried to the product in $\mathcal{G}^{-}$ by the co-reflection.

It is important to note that the above proposition is stated only for $\mathcal{G}$, not for $\mathcal{G}_{\mathtt{hf}}$. History free strategies do *not* suffice for the additives. This seems to the key reason underlying the problems encountered with additives in the Geometry of Interaction.

We also note that the surjective pairing axiom for product (and hence the commutative conversion for With) will only be valid in $\mathcal{G}^{-}$. This suggest a syntactic restriction on the With rule, based on the polarities.

Firstly we give a table of how the connectives act on polarities. Read +1 (−1) as "*must be* positive (negative)" and 0 as "*may be* neutral".

A	B	$A\otimes B$	$A⅋B$	$A\multimap B$	$A\&B$	$A\oplus B$
$+1$	$+1$	$+1$	$+1$	0	$+1$	$+1$
$+1$	0	0	0	0	0	0
$+1$	-1	0	0	-1	0	0
0	$+1$	0	0	0	0	0
0	0	0	0	0	0	0
0	-1	0	0	0	0	0
-1	$+1$	0	0	$+1$	0	0
-1	0	0	0	0	0	0
-1	-1	-1	-1	0	-1	-1

A	$!A$	$?A$	$A^{\perp}$
$+1$	-1	$+1$	-1
0	-1	$+1$	0
-1	-1	$+1$	$+1$

Using these tables as a *definition*, we now have a syntactic notion of polarity, and can use it for the following refined With Rule:

$$\frac{\vdash \Gamma^+, A \qquad \vdash \Gamma^+, B}{\vdash \Gamma^+, A\&B} \qquad (\text{ With}^p)$$

The Γ^+ is meant to indicate the constraint that all formulas in Γ must be positive. Let $\mathtt{LL}^p$ be the modification of Classical Linear Logic obtained by replacing the usual With Rule with With^p. Then the commutative conversion for With will be valid in our game semantics for $\mathtt{LL}^p$. We also expect that $\mathtt{LL}^p$ can be used to extend the Geometry of Interaction interpretation to the additives.

Proposition 15 *There are isomorphisms* $!(A\&B) \cong !A\otimes !B$, $!\top \cong \mathbf{1}$ *and hence (*cf. *[See89]), the co-Kleisli category* $K_!(\mathcal{G})$ *is cartesian closed.*

6 Related Work

Since a number of researchers have recently examined categories of games, or at least categories with some game-theoretic flavour, it seems worthwhile to make some explicit comparisons.

6.1 Conway games

As far as we know, the first person to make a category of games and winning strategies was Joyal [Joy77]. His category was based on Conway games [Con76] with Conway's addition of games as the tensor product. Conway's formalization of games differs from ours in that he presents the tree of positions directly, rather than via an underlying set of moves. This means that strategies must be formalized as functions on positions, and hence are necessarily history-sensitive; the possibility of introducing history-free strategies in our sense does not even arise.

More precisely, a Conway game can be taken to be one of our games with the following property: for all $a \in M_A$ there is a unique $s \in P_A$ such that $s \cdot a \in P_A$. Call such a game *positional.*

Proposition 16 *Given any game A in $\mathcal{G}$, there is a positional game A^{pos} such that $A \cong A^{\mathrm{pos}}$ in $\mathcal{G}$. Moreover,* every *strategy in A^{pos} is history-free. However, A is* not *isomorphic to A^{pos} in $\mathcal{G}_{\mathtt{hf}}$.*

Thus working with positional games as Conway does would obliterate the distinction between history-free and history-sensitive which is crucial to our Full Completeness Theorem. In this respect, our games are more general than Conway's.

In another respect, however, Conway games are more general than ours, at least superficially. Think of the set of positions of the game as a tree, with arcs $s \rightarrow s \cdot a$ labelled P or O, according to the label of a. Say that a node is *pure* if all outgoing arcs have the same label, and *mixed* otherwise. In Blass' games, all nodes are pure. In Conway's games, all nodes are allowed to be mixed. Our games are intermediate in generality; the root is allowed to be mixed, but all other nodes are pure. Conway games—or their generalization to the non-positional case—can be represented in our framework by dropping the stipulation that positions be *strictly alternating* sequences of moves. His notion of "sum of games", which is used by Joyal as the basis for his construction of a category of games, then arises by dropping the stipulation from our definition of tensor product that only Opponent is allowed to switch components. This immediately obliterates the distinction between Tensor and Par; Hyland [Hyl90] has shown that Joyal's category does not admit satisfactory interpretations of the additives and exponentials.

Our games are *apparently* less general than Conway's; however, as soon as our definition of tensor product is adopted (with the consequent notion of morphism; note that Joyal's definition of winning strategy agrees with ours), this difference disappears. The key observation is the following. Let A, B be Conway games. Apply our definition of tensor product to form $A \otimes B$. Now, because of the stipulation that only Opponent can switch components, a strictly alternating sequence of moves in $A \otimes B$ must project onto strictly alternating sequences in A and B. (Of course, this property fails with Conway's sum of games). As a consequence of this, we have the following Proposition.

Proposition 17 *Let $\mathcal{C}$ be the category of Conway games, with our definition of tensor product, and the consequent notion of morphism from A to B as a winning strategy in $A \multimap B = (A \otimes B^{\perp})^{\perp}$. (So, in particular, this is not the category studied by Joyal [Joy77].) $\mathcal{G}$ is a full subcategory of $\mathcal{C}$. If A is a Conway game, let $A^{\mathtt{alt}}$ be the game in $\mathcal{G}$ obtained by deleting all non-strictly-alternating sequences in P_A (and correspondingly pruning W_A). Then $A \cong A^{\mathtt{alt}}$ in $\mathcal{C}$; so $\mathcal{G} \simeq \mathcal{C}$. Moreover, $(A \otimes B)^{\mathtt{alt}} \cong A^{\mathtt{alt}} \otimes B^{\mathtt{alt}}$.*

The upshot of this Proposition is that, once our definition of tensor product—which has been justified both conceptually and by our results in this paper—is adopted, then one may as well work in $\mathcal{G}$ as in $\mathcal{C}$.

6.2 Abstract Games

De Paiva has studied the Dialectica Categories DC, and Linear categories GC [dP89]. These are abstract constructions, but reflect some game-theoretic intuitions. Indeed, Blass applies his game semantics to DC [Bla92b]. Again, Lafont and Streicher [LS91] have developed a "Game Semantics for Linear Logic". An object in the category $\mathtt{Game}_K$ is a structure $(A^{\star}, A_{\star}, e)$, where $e : A^{\star} \times A_{\star} \rightarrow K$, for some fixed set K. If we think of $A^{\star}$ as strategies for Player, $A_{\star}$ as counter-strategies and e as the payoff function, we see some connection with game-theoretic ideas. However, this model is very abstract; in fact it forms a particular case of Chu's very general construction of $\star$-autonomous categories from symmetric monoidal closed categories [Bar79].

In summary, these models have only rudimentary game-theoretic content and hence only a very weak relation with our work.

6.3 Blass' game semantics

Blass' game semantics for Linear Logic is by far the nearest precursor of the present work. While we happily acknowledge its inspiration, we must also say that, in our opinion, our semantics is a decisive improvement over that of Blass, as our results show.

It is worth setting out the key points in some detail, since our identification of the problems in Blass' semantics was a crucial step in our own work and differs sharply from Blass' analysis of the discrepancy between his semantics and Linear Logic.

The games Blass considers correspond to those in $\mathcal{G}^+ \cup \mathcal{G}^-$ in our framework; that is, to either positive games (all opening moves by Player) or negative games (all opening moves by Opponent). This means, among other things, that all connectives must be defined by cases on the polarity of their arguments; and, more importantly, the resulting game must itself have a definite positive or negative polarity. The plays in Blass' games are then started by Player for a positive game and by Opponent for a negative game.

The key difference between Blass' approach and ours concerns the definition of tensor product. Blass' rule for who moves next in the tensor product is that Player moves if he is to move in either game. This makes sense if we think of "Opponent to move" as a kind of approximation to the proposition represented by the tensor product being true—since the onus is on the Opponent to move in order to avoid defeat—and the tensor as a kind of conjunction. Surprisingly enough, this definition turns out to *almost* coincide with ours. Suppose that we are in a position where Opponent is to move in both subgames; then he has the choice of moving in either component, leading to a position where Player is to move in just one component. In this latter situation, Player is forced to move in the component where Opponent last moved. Such a move will return us to a situation where Opponent is to move in both components. This leaves just one anomalous situation, where Player is to start in both components. This is the only case where the situation can arise that Player must move next in both games. Note that in our framework, this situation can never arise at all. Also, note that this situation contradicts our previous analysis of tensor; for example, in terms of the trip conditions, it corresponds to the forbidden sequence $A_\vee B_\vee$. Blass treats this anomalous situation as a special case; Player makes his opening move simultaneously in both components. This special case is at the heart of the pathologies in his semantics.

6.3.1 Composition

Composition is not associative in Blass' semantics [Bla92a]; so he does not get a category of games at all. The following counter-example is due to the authors.

Define games A, B, C, D as follows:

$$\begin{array}{rcl}
A & = & (\{a_1, a_2\}, \{(a_1, P), (a_2, O)\}, \{(a_1 \cdot a_2)^\star \cdot (\epsilon + a_1)\}, \varnothing) \\
B & = & (\{b_1, b_2\}, \{(b_1, O), (b_2, P)\}, \{(b_1 \cdot b_2)^\star \cdot (\epsilon + b_1)\}, \varnothing) \\
C & = & (\{c_1, c_2\}, \{(c_1, P), (c_2, O)\}, \{(c_1 \cdot c_2)^\star \cdot (\epsilon + c_1)\}, \{(c_1 \cdot c_2)^\omega\}) \\
D & = & (\{d_1, d_2\}, \{(d_1, O), (d_2, P)\}, \{(d_1 \cdot d_2)^\star \cdot (\epsilon + d_1)\}, \{(d_1 \cdot d_2)^\omega\})
\end{array}$$

There are winning strategies $\sigma : A \to B$, $\tau : B \to C$, $\upsilon : C \to D$. σ is the strategy that forces the entire play to stay in constituent $A^\perp$ after the first move. Similarly, υ is the strategy that forces the entire play to stay in constituent D after the first move. τ is the strategy that forces the entire play to stay in the constituent chosen by the Opponent in response to the first move of Player. More precisely,

$$\begin{array}{rcl}
\sigma & = & \{(\langle a_1, b_1\rangle, a_2)\} \cup \{(s, a_2) \mid s \in \langle a_1, b_1\rangle \cdot (a_2 \cdot a_1)^\star\} \\
\tau & = & \{(\epsilon, \langle b_1, c_1\rangle)\} \cup \{(s, b_1) \mid s \in \langle b_1, c_1\rangle \cdot b_2 \cdot (b_1 \cdot b_2)^\star\} \cup \{(s, c_1) \mid s \in \langle b_1, c_1\rangle \cdot c_2 \cdot (c_1 \cdot c_2)^\star\} \\
\upsilon & = & \{(\langle c_1, d_1\rangle, d_2)\} \cup \{(s, d_2) \mid s \in \langle c_1, d_1\rangle \cdot (d_2 \cdot d_1)^\star\}
\end{array}$$

Here a move $\langle a, b\rangle$ is an opening move in the special case described above.

Consider the winning strategy $\sigma;\tau : A \to C$. Opponent makes first move in C. Player responds using strategy τ in C and the whole play stays in the constituent C. Note that strategy σ is never used.

$$\sigma;\tau = \{(s, c_1) \mid s \in a_1 \cdot (c_1 \cdot c_2)^\star\}$$

Consider the winning strategy $\tau;\upsilon : B \to D$. Opponent makes first move in D. Note that the strategy υ cannot be used at this stage . So, Player responds with a move in B, and the rest of the play stays in the constituent B. Thus, the strategy υ is not used.

$$\tau;\upsilon = \{(s, b_1) \mid s \in d_1 \cdot (b_1 \cdot b_2)^\star\}$$

Reasoning similarly, the compositions $(\sigma;\tau);\upsilon, \sigma;(\tau;\upsilon) : A \to D$ are

$$\begin{aligned}(\sigma;\tau);\upsilon &= \{(\langle a_1, d_1\rangle, d_2)\} \cup \{(s, d_2) \mid s \in (\langle a_1, d_1\rangle \cdot (d_2 \cdot d_1)^\star\} \\ \sigma;(\tau;\upsilon) &= \{(\langle a_1, d_1\rangle, a_2)\} \cup \{(s, a_2) \mid s \in (\langle a_1, d_1\rangle \cdot (a_2 \cdot a_1)^\star\}\end{aligned}$$

and hence unequal.

6.3.2 Weakening

Weakening is valid in the Blass semantics. To see why, suppose that Player has a winning strategy for Γ. Consider the game Γ, A. If A is positive, Opponent cannot move in A and since only Player can switch components in a Par, we need never play in A at all. (Of course, this is exactly the argument for the validity of weakening with respect to $?_w A$ in our semantics). If A is negative, there are two cases.

- Some game in Γ is positive: so Player is to start in Γ and Γ, A. Thus, Player can simply play his strategy for Γ without ever entering A.
- All games in Γ, A are negative: The special case takes effect and Opponent must make his opening move in every component of Γ, A. Then, Player can simply ignore the opening move in A and play as he would have done in response to the opening moves in Γ.

By contrast, in our interpretation, unless A is positive, Opponent can move in A, and Player may have no way to respond; so Weakening is not valid.

6.3.3 An Example

Consider the example discussed in Blass' paper ([Bla92b], pp.210-213). The sequent considered there is:

$$(A^\perp ⅋ B^\perp) \otimes (C^\perp ⅋ D^\perp), (A ⅋ C) \otimes (B ⅋ D)$$

We describe a strategy for Opponent, which with suitable choice of games for A, B, C, D will defeat Player in our semantics.

1. Opponent moves in A.
2. Player moves in $A^\perp$.
3. Opponent moves in $C^\perp$.
4. Player moves in C.
5. Opponent moves in B.

At this point, Player needs to move in $B^\perp$; however, he cannot, because it is Opponent's move in the sub-game $A^\perp ⅋ B^\perp$. What saves the Player in Blass' semantics is again the special case, which would force Opponent to move in both B and D simultaneously, thus allowing Player to respond in $D^\perp$.

6.4 Sequential Algorithms

Lamarche [Lam92] and more recently, but independently, Curien[2] [Cur92] have found linear decompositions of the Berry-Curien category of sequential algorithms on (filiform) concrete data structures [BC85]. That is, they have described models of Linear Logic (Intuitionistic Linear Logic only, in Curien's case) such that the co-Kleisli category is equivalent to the Berry-Curien category. Moreover, these Linear categories have a game-theoretic flavour. In fact, we have the correspondence:

Game	Concrete Data Structure
O-Moves	Cells
P-Moves	Values
Positions	Enabling Relation
Strategy	State

We have not seen the full details of Lamarche's work; Curien's construction can be related to our work as follows. The objects in his category are exactly our negative games, minus the information about infinite plays. The morphisms correspond to strategies—which need be neither history-free nor winning. His interpretations of the Intuitionistic linear connectives, with these provisos, appear to correspond to ours. We take this link with sequential algorithms as an encouraging confirmation of the potential of game semantics. We note, finally, that the connection between sequential algorithms and negative games confirms our identification of negative games with demand-driven computation. This also ties up with the first author's association of & and ! (more precisely of $!_w$) with lazy evaluation [Abr93].

References

[Abr91] S. Abramsky. Proofs as processes. Unpublished Lecture, 1991.

[Abr93] S. Abramsky. Computational interpretations of linear logic. *Theoretical Computer Science*, 1993. To appear. Revised version of Imperial College Technical Report DoC 90/20, October 1990.

[AJ92a] S. Abramsky and R. Jagadeesan. New foundations for the geometry of interaction. In *Proceedings of the Seventh Symposium on Logic In Computer Science*, pages 211–222. Computer Society Press of the IEEE, June 1992.

[AJ92b] S. Abramsky and R. Jagadeesan. A strong completeness theorem for multiplicative linear logic: Preliminary announcement. Email communication on types mailing list, 1992.

[AV93] S. Abramsky and S. Vickers. Quantales, observational logic and process semantics. *Mathematical Structures in Computer Science*, 1993. To appear. Revised version of Imperial College Technical Report DoC 90/1, January 1990.

[Bar79] M. Barr. *$\star$-autonomous categories*, volume 752 of *Lecture Notes in Mathematics*. Springer-Verlag, 1979.

[Bar91] M. Barr. $\star$-autonomous categories and linear logic. *Mathematical Structures in Computer Science*, 1(2):159–178, July 1991.

[2] Curien's email announcement of his results appeared following ours [AJ92b] announcing the results of this paper.

[BC85] G. Berry and P.-L. Curien. Theory and practice of sequential algorithms: the kernel of the applicative language CDS. In J. C. Reynolds and M. Nivat, editors, *Algebraic Semantics*, pages 35–84. Cambridge University Press, 1985.

[BFSS90] S. Bainbridge, P. J. Freyd, A. Scedrov, and P. Scott. Functorial polymorphism. *Theoretical Computer Science*, 70:35–64, 1990.

[Bla92a] A. Blass, 1992. Personal communication.

[Bla92b] A. Blass. A game semantics for linear logic. *Annals of Pure and Applied Logic*, 56:183–220, 1992.

[Blu92] R. Blute. Linear logic, coherence and dinaturality. Technical report, McGill University, 1992.

[CGW87] T. Coquand, C. Gunter, and G. Winskel. dI-domains as a model of polymorphism. In *Third Workshop on the Mathematical Foundations of Programming Language Semantics*, pages 344–363. Springer-Verlag, 1987.

[Con76] J. H. Conway. *On Numbers and Games*, volume 6 of *London Mathematical Society Monographs*. Academic Press, 1976.

[Cur92] P. L. Curien. Concrete data structures, sequential algorithms and linear logic. Email communication on types mailing list, 1992.

[dP89] V. C. V. de Paiva. The Dialectica categories. In J. W. Gray and A. Scedrov, editors, *Categories in Computer Science and Logic*, pages 47–62, 1989.

[DP90] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 1990.

[DR89] V. Danos and L. Regnier. The structure of multiplicatives. *Archive for Mathematical Logic*, 28:181–203, 1989.

[FR90] A. Fleury and C. Retoré. The MIX rule. Unpublished note, 1990.

[FS91] P. J. Freyd and A. Scedrov. *Categories, Allegories*, volume 39 of *North-Holland Mathematical Library*. Elsevier Science Publishers, 1991.

[GAL92] G. Gonthier, M. Abadi, and J. J. Levy. Linear logic without boxes. In *Proceedings of the Seventh Symposium on Logic in Computer Science*, pages 223–234. Computer Society Press of the IEEE, 1992.

[Gir86] J.-Y. Girard. The system F of variable types, fifteen years later. *Theoretical Computer Science*, 45:159–192, 1986.

[Gir87] J.-Y. Girard. Linear Logic. *Theoretical Computer Science*, 50(1):1–102, 1987.

[Gir88] J.-Y. Girard. Geometry of interaction 2: Deadlock-free algorithms. In P. Martin-Löf and G. Mints, editors, *International Conference on Computer Logic, COLOG 88*, pages 76–93. Springer-Verlag, 1988. Lecture Notes in Computer Science 417.

[Gir89a] J.-Y. Girard. Geometry of interaction 1: Interpretation of System F. In R. Ferro et al., editor, *Logic Colloquium 88*. North Holland, 1989.

[Gir89b] J.-Y. Girard. Towards a geometry of interaction. In J. W. Gray and A. Scedrov, editors, *Categories in Computer Science and Logic*, volume 92 of *Contemporary Mathematics*, pages 69–108. American Mathematical Society, 1989.

[Gir91a] J.-Y. Girard. A new constructive logic: classical logic. *Mathematical Structures in Computer Science*, 1:255–296, 1991.

[Gir91b] J.-Y. Girard. On the unity of logic. Submitted to Annals of Pure and Applied Logic, 1991.

[Hoa85] C. A. R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.

[HRR89] J. M. E. Hyland, E. P. Robinson, and G. Rosolini. Algebraic types in per models. In *Fifth Conference on Mathematical Foundations in Programming Semantics*, pages 333–350. Springer-Verlag, 1989. Lecture Notes in Computer Science 442.

[Hyl90] J. M. E. Hyland. Conway games and linear logic. Unpublished lecture, 1990.

[Jac92] B. Jacobs. Semantics of weakening and contraction. Preprint, 1992.

[Joy77] A. Joyal. Remarques sur la theorie des jeux a deux personnes. *Gazette des sciences mathematiques du Quebec*, 1(4), 1977.

[Laf90] Y. Lafont. Interaction nets. In *Proceedings of the Seventeenth ACM Symposium on Principles of Programming Languages*, pages 95–108, 1990.

[Lam92] F. Lamarche. Sequential algorithms, games and linear logic. Unpublished Lecture, 1992.

[LS91] Y. Lafont and T. Streicher. Games semantics for linear logic. In *Proc. Sixth Annual Symposium on Logic in Computer Science*, pages 43–51. Computer Society Press, 1991.

[Mil75] R. Milner. Processes, a mathematical model of computing agents. In *Logic Colloquium, Bristol 1973*, pages 157–174. North Holland, Amsterdam, 1975.

[Plo77] G. D. Plotkin. LCF considered as a programming language. *Theoretical Computer Science*, 5:223–255, 1977.

[See89] R. Seeley. $\star$-autonomous categories, cofree coalgebras and linear logic. In J. W. Gray and A. Scedrov, editors, *Categories in Computer Science and Logic*, volume 92 of *Contemporary Mathematics*, pages 371–382. American Mathematical Society, 1989.