A Categorical Model for the Geometry of Interaction

Esfandiar Haghverdi[1] and Philip Scott[2]

[1] School of Informatics & Department of Mathematics, University of Indiana Bloomington, Indiana, USA ehaghver@indiana.edu
[2] Department of Mathematics, University of Ottawa, Ottawa, Ontario, K1N 6N5, CANADA phil@mathstat.uottawa.ca

Abstract. We consider the multiplicative and exponential fragment of linear logic (MELL) and give a Geometry of Interaction (GoI) semantics for it based on unique decomposition categories. We prove a Soundness and Finiteness Theorem for this interpretation. We show that Girard's original approach to GoI 1 via operator algebras is exactly captured in this categorical framework.

1 Introduction and Motivation

Girard introduced his Geometry of Interaction (GoI) program in the late 80's, through a penetrating series of papers [10, 9, 11].

The Geometry of Interaction was the first attempt to model, in a mathematically sophisticated way, the dynamics of cut-elimination. Traditional denotational semantics models normalization of proofs (or lambda terms) by static equalities: if Π, Π' are proofs of a sequent $\Gamma \vdash A$ and if we have a reduction $\Pi \succ \Pi'$ by cut-elimination, then their interpretations $[\![-]\!]$ in any model denote equal morphisms, i.e. $[\![\Pi]\!] = [\![\Pi']\!] : [\![\Gamma]\!] \to [\![A]\!]$. On the other hand *syntax* contains too much irrelevant information and does not yield an independent mathematical modelling of the dynamics of cut-elimination. Thus the goal of GoI is to provide precisely such a mathematical model.

The first implementation of this programme was given by Girard [10], based on the C^*-algebra of bounded linear operators on the space ℓ^2 of square summable sequences. For a much more elaborate account of the ideas above see [10, 9, 11].

The GoI interpretation was extended to untyped λ-calculus by Danos in [7]. Danos and Regnier further extended the GoI interpretation to define a *path-semantics* for proofs (=programs) and gave a detailed comparison with the λ-calculus notions of path. The idea is that a proof net is represented by a set of paths and the execution formula is an *invariant* of reduction (see [8]).

Abramsky and Jagadeesan gave the first categorical approach to GoI in [4]. Their formalisation is based on domain theory and arises from the construction of a categorical model of linear logic. The ideas and techniques used in [4] together with the development of traced monoidal categories, introduced by Joyal, Street and Verity [17], led to a more abstract formalisation of GoI via the notion of *GoI Situation* introduced by Abramsky in [2]. GoI Situations give a categorical embodiment of the essential ingredients of GoI, at least for the multiplicative and exponential fragment. Furthermore, in his Siena lecture [2] Abramsky introduced a general GoI construction. Abramsky's programme was sketched in [2] and completed in [12] and [3]. However, what was still missing was a tighter connection between the abstract GoI frameworks above and the original works of Girard et al. That is, we want our categorical models for GoI to be not only part of well-established categorical logic, but also we want our framework to explicitly connect with the details of the operator algebraic approach, e.g. the execution formula, orthogonality and the notion of type, all found in the original works but which could not be given in the generality of [3].

In this paper, we analyze how the first Girard paper GoI1 [10] fits into the general theory of GoI situations. The idea pursued here is to restrict the abstract traced monoidal categories in a GoI situation to a useful subclass: unique decomposition categories [12, 13]. These are monoidal categories whose homsets are enriched with certain infinitary sums, thus allowing us to consider morphisms as matrices, the execution formula as an infinite sum, etc. Such categories are inspired from early categorical analyses of programming languages by Elgot, Arbib and Manes, et. al. (e.g. [18]).

The main contributions of this paper are the following:

1. We present a categorical model (implementation) for GoI and show that it captures the original Hilbert space model proposed by Girard in [10], including the notions of orthogonality and type.
2. We show that the execution formula at the heart of modeling computation as cut-elimination is perfectly captured by the categorical notion of trace.
3. We prove finiteness and soundness results for our model using the categorical properties of trace and GoI Situation.

We believe that our categorical interpretation views the original Girard GoI model in a new light. Not only do the original constructions appear less ad hoc, but this paper also opens the door towards accommodating other interesting models based on different categories and GoI Situations.

The rest of the paper is organized as follows: In Section 2 we recall the definitions of traced monoidal categories and GoI Situations, following [12, 3]. In Section 3 we recall the definition of a unique decomposition category and give some examples. Sections 4 and 5 are the main sections of the paper where we discuss our categorical model for the GoI program and give the main theorems respectively. Section 6 discusses the original model introduced by Girard in [10]. Finally in section 7 we conclude by discussing related and future work.

2 Traced Monoidal Categories & GoI Situations

We recall the definitions of symmetric traced monoidal categories and GoI Situations. For more detailed expositions, see [12, 3]. The categories introduced below admit a highly geometric presentation, but for lack of space, we omit drawing the pictures, and refer the reader to the above references.

Joyal, Street and Verity [17] introduced the notion of abstract trace on a balanced monoidal category (a monoidal category with braidings and twists.) This trace can be interpreted in various contexts where it could be called contraction, feedback, parametrized fixed-point, Markov trace or braid closure. The notion of trace can be used to analyse the cyclic structures encountered in mathematics and physics, most notably in knot theory. Since their introduction, traced monoidal categories have found applications in many different areas of computer science, for example the model theory of cyclic lambda calculi [14], categorical frameworks for the semantics of asynchronous communication networks [19], full completeness theorems for multiplicative linear logic via GoI models [12], analysis of finite state machines [16], relational dataflow [15], and independently arose in Stefanescu's work in network algebra [20].

Definition 1. A *traced symmetric monoidal category* is a symmetric monoidal category $(\mathbb{C}, \otimes, I, s)$ with a family of functions $Tr^U_{X,Y} : \mathbb{C}(X \otimes U, Y \otimes U) \to \mathbb{C}(X, Y)$ called a *trace*, subject to the following axioms:

– **Natural** in X, $Tr^U_{X,Y}(f)g = Tr^U_{X',Y}(f(g \otimes 1_U))$ where $f : X \otimes U \to Y \otimes U$, $g : X' \to X$,
– **Natural** in Y, $gTr^U_{X,Y}(f) = Tr^U_{X,Y'}((g \otimes 1_U)f)$ where $f : X \otimes U \to Y \otimes U$, $g : Y \to Y'$,
– **Dinatural** in U, $Tr^U_{X,Y}((1_Y \otimes g)f) = Tr^{U'}_{X,Y}(f(1_X \otimes g))$ where $f : X \otimes U \to Y \otimes U'$, $g : U' \to U$,
– **Vanishing (I,II)**, $Tr^I_{X,Y}(f) = f$ and $Tr^{U \otimes V}_{X,Y}(g) = Tr^U_{X,Y}(Tr^V_{X \otimes U, Y \otimes U}(g))$ for $f : X \otimes I \to Y \otimes I$ and $g : X \otimes U \otimes V \to Y \otimes U \otimes V$,
– **Superposing**, $Tr^U_{X,Y}(f) \otimes g = Tr^U_{X \otimes W, Y \otimes Z}((1_Y \otimes s_{U,Z})(f \otimes g)(1_X \otimes s_{W,U}))$ for $f : X \otimes U \to Y \otimes U$ and $g : W \to Z$,
– **Yanking**, $Tr^U_{U,U}(s_{U,U}) = 1_U$.

Joyal, Street, and Verity[17] also introduced the *Int* construction on traced symmetric monoidal categories $\mathbb{C}$; $Int(\mathbb{C})$ is a kind of "free compact closure" of the category $\mathbb{C}$. $Int(\mathbb{C})$ isolates the key properties of Girard's GoI for the multiplicative connectives, in that composition in $Int(\mathbb{C})$, which is defined via the trace, uses an abstract version of Girard's Execution Formula. Of course, one of our goals in this paper is to show that in our restricted models, this is exactly the original Girard formula.

The next problem was how to extend this to the exponential connectives. In the Abramsky program (see [3]) this is achieved by adding certain additional structure to a traced symmetric monoidal category. This structure involves a monoidal endofunctor, a reflexive object, and appropriate retractions, as introduced below. It was shown in [3] that this additional structure is sufficient to generate certain *linear combinatory algebras* which capture the appropriate computational meaning of the exponentials.

Definition 2. A *GoI Situation* is a triple $(\mathbb{C}, T, U)$ where:

1. $\mathbb{C}$ is a traced symmetric monoidal category
2. $T : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor with the following retractions (note that the retraction pairs are monoidal natural transformations):
   (a) $TT \lhd T$ (e, e') (Comultiplication)
   (b) $Id \lhd T$ (d, d') (Dereliction)
   (c) $T \otimes T \lhd T$ (c, c') (Contraction)
   (d) $\mathcal{K}_I \lhd T$ (w, w') (Weakening). Here $\mathcal{K}_I$ is the constant I functor.
3. U is an object of $\mathbb{C}$, called *a reflexive object*, with retractions: (a) $U \otimes U \lhd U$ (j, k), (b) $I \lhd U$, and (c) $TU \lhd U$ (u, v).

For examples of GoI Situations see Section 6.

3 Unique Decomposition Categories

We consider monoidal categories whose homsets allow the formation of certain infinite sums. Technically, these are monoidal categories enriched in Σ-monoids (see below). In the case where the tensor is coproduct and Σ-monoids satisfy an additional condition, such categories were studied in computer science in the early categorical analyses of flow charts and programming languages by Bainbridge, Elgot, Arbib and Manes, et. al. (e.g. [18]). The general case, known as unique decomposition categories (UDC's), are particularly relevant for this paper, since they admit arbitray tensor product (not necessarily product or coproduct) and traced UDCs have a standard trace given as an infinite sum. For more facts and examples on UDCs see [12].

Definition 3. A *Σ-monoid* consists of a pair (M, Σ) where M is a nonempty set and Σ is a partial operation on the countable families in M (we say that $\{x_i\}_{i\in I}$ is *summable* if $\sum_{i\in I} x_i$ is defined), subject to the following axioms:

1. *Partition-Associativity Axiom.* If $\{x_i\}_{i\in I}$ is a countable family and if $\{I_j\}_{j\in J}$ is a (countable) partition of I , then $\{x_i\}_{i\in I}$ is summable if and only if $\{x_i\}_{i\in I_j}$ is summable for every $j \in J$ and $\sum_{i\in I_j} x_i$ is summable for $j \in J$. In that case, $\sum_{i\in I} x_i = \sum_{j\in J}(\sum_{i\in I_j} x_i)$
2. *Unary Sum Axiom.* Any family $\{x_i\}_{i\in I}$ in which I is a singleton is summable and $\sum_{i\in I} x_i = x_j$ if $I = \{j\}$.

Σ-monoids form a symmetric monoidal category (with product as tensor), called Σ**Mon**. A Σ**Mon**-category $\mathbb{C}$ is a category enriched in Σ**Mon**; i.e. the homsets are enriched with an additive structure such that composition distributes over addition from left and right. Note that such categories have nonempty homsets and automatically have zero morphisms, namely $0_{XY} : X \to Y = \sum_{i\in\emptyset} f_i$ for $f_i \in \mathbb{C}(X, Y)$. This does not imply the existence of a zero object.

Definition 4. A *unique decomposition category* (UDC) $\mathbb{C}$ is a symmetric monoidal Σ**Mon**-category which satisfies the following axiom:

(A) For all $j \in I$ there are morphisms called *quasi injections*: $\iota_j : X_j \to \otimes_I X_i$, and *quasi projections*: $\rho_j : \otimes_I X_i \to X_j$, such that 1. $\rho_k \iota_j = 1_{X_j}$ if $j = k$ and $0_{X_j X_k}$ otherwise. 2. $\sum_{i\in I} \iota_i \rho_i = 1_{\otimes_I X_i}$.

Proposition 1 (Matricial Representation). *Given $f : \otimes_J X_j \to \otimes_I Y_i$ in a UDC with $|I| = m$ and $|J| = n$, there exists a unique family $\{f_{ij}\}_{i\in I, j\in J} : X_j \to Y_i$ with $f = \sum_{i\in I, j\in J} \iota_i f_{ij} \rho_j$, namely, $f_{ij} = \rho_i f \iota_j$.*

Thus every $f : \otimes_J X_j \to \otimes_I Y_i$ in a UDC can be represented by its components. We will use the corresponding matrices to represent morphisms; for example f above (with $|I| = m$ and $|J| = n$) is represented by an $m \times n$ matrix $[f_{ij}]$. Composition of morphisms in a UDC then corresponds to matrix multiplication.

Remark. Although any $f : \otimes_J X_j \to \otimes_I Y_i$ can be represented by the unique family $\{f_{ij}\}$ of its components, the converse is not necessarily true; that is, given a family $\{f_{ij}\}$ with I, J finite there may not be a morphism $f : \otimes_J X_j \to \otimes_I Y_i$ satisfying $f = \sum_{ij} \iota_i f_{ij} \rho_j$. However, in case such an f exists it will be unique.

Proposition 2 (Execution/Trace Formula). *Let $\mathbb{C}$ be a unique decomposition category such that for every X, Y, U and $f : X \otimes U \to Y \otimes U$, the sum $f_{11} + \sum_{n=0}^{\infty} f_{12} f_{22}^n f_{21}$ exists, where f_{ij} are the components of f. Then, $\mathbb{C}$ is traced and $Tr_{X,Y}^U(f) = f_{11} + \sum_{n=0}^{\infty} f_{12} f_{22}^n f_{21}$.*

Example 1.

1. Consider the category **PInj** of sets and partial injective functions. Define $X \otimes Y = X \uplus Y$ (disjoint union); note that this does not give a coproduct, indeed **PInj** does not have coproducts. The UDC structure is given as follows: define $\iota_j : X_j \to \biguplus_{i\in I} X_i$ by $\iota_j(x) = (x, j)$, and define $\rho_j : \biguplus_{i\in I} X_i \to X_j$ by $\rho_j(x, j) = x$, and $\rho_j(x, i)$ is undefined for $i \neq j$.
2. This example will provide the connection to operator algebraic models. Given a set X let $\ell_2(X)$ be the set of all complex valued functions a on X for which the (unordered) sum $\sum_{x\in X} |a(x)|^2$ is finite. $\ell_2(X)$ is a Hilbert space and its norm is given by $||a|| = (\sum_{x\in X} |a(x)|^2)^{1/2}$ and its inner product is given by $< a, b >= \sum_{x\in X} a(x)\overline{b(x)}$ for $a, b \in \ell_2(X)$.

   Barr [6] observed that there is a contravariant faithful functor $\ell_2 : \mathbf{PInj}^{op} \to \mathbf{Hilb}$ where **Hilb** is the category of Hilbert spaces with morphisms the linear contractions (norm ≤ 1). For a set X, $\ell_2(X)$ is defined as above and given $f : X \to Y$ in **PInj**, $\ell_2(f) : \ell_2(Y) \to \ell_2(X)$ is defined by: $\ell_2(f)(b)(x) = b(f(x))$, if $x \in Dom(f)$ and 0, otherwise. This gives a correspondence between partial injective functions and partial isometries on Hilbert spaces ([11, 1]).

   Let $\mathbf{Hilb}_2 = \ell_2[\mathbf{Pinj}]$; i.e. its objects are of the form $\ell_2(X)$ for a set X and its morphisms $u : \ell_2(X) \to \ell_2(Y)$ are of the form $\ell_2(f)$ for some partial injective function $f : Y \to X$. $\mathbf{Hilb_2}$ is a (nonfull) subcategory of **Hilb**.
   For $\ell_2(X)$ and $\ell_2(Y)$ in $\mathbf{Hilb}_2$, the Hilbert space tensor product $\ell_2(X) \otimes \ell_2(Y)$ and the direct sum $\ell_2(X) \oplus \ell_2(Y)$ yield tensor products in $\mathbf{Hilb}_2$. $\mathbf{Hilb_2}$ is a traced UDC with respect to $\oplus$, where the UDC structure is induced from that of **PInj**; for more details see [12, 3].
3. All partially additive categories [18, 12] are examples of traced UDCs.

4 Interpretation of Proofs

In this section we define the GoI interpretation for proofs of MELL without the neutral elements. Let $\mathbb{C}$ be a traced UDC, T an additive endofunctor and U an object of $\mathbb{C}$, such that $(\mathbb{C}, T, U)$ is a GoI Situation. We interpret proofs in the homset $\mathbb{C}(U, U)$ of endomorphisms of U. Formulas (= types) will be interpreted in the next Section 5 as certain subsets of $\mathbb{C}(U, U)$; however, this introduces some novel ideas and is not needed to read the present section.

Convention: All identity morphisms are on tensor copies of U however we adopt the convention of writing 1_Γ instead of $1_{U^{\otimes n}}$ with $|\Gamma| = n$. U^n denotes the n-fold tensor product of U by itself. The retraction pairs are fixed once and for all.

Every MELL sequent will be of the form $\vdash [\Delta], \Gamma$ where Γ is a sequence of formulas and Δ is a sequence of cut formulas that have already been made in the proof of $\vdash \Gamma$ (e.g. $A, A^\perp, B, B^\perp$). This is used to keep track of the cuts that are already made in the proof of $\vdash \Gamma$. Suppose that Γ consists of n and Δ consists of $2m$ formulas. Then a proof Π of $\vdash [\Delta], \Gamma$ is represented by a morphism $[\![\Pi]\!] \in \mathbb{C}(U^{n+2m}, U^{n+2m})$. Recall that this corresponds to a morphism from U to itself, using the retraction morphisms $U \otimes U \lhd U\ (j, k)$. However, it is much more convenient to work in $\mathbb{C}(U^{n+2m}, U^{n+2m})$ (matrices on $\mathbb{C}(U, U)$). Define the morphism $\sigma : U^{2m} \to U^{2m}$, as $\sigma = s \otimes \cdots \otimes s$ (m-copies) where s is the symmetry morphism, the 2×2 antidiagonal matrix $[a_{ij}]$, where $a_{12} = a_{21} = 1; a_{11} = a_{22} = 0$. Here σ represents the cuts in the proof of $\vdash \Gamma$, i.e. it models Δ. If Δ is empty (that is for a cut-free proof), we define $\sigma : I \to I$ to be the zero morphism 0_{II}. Note that $U^0 = I$ where I is the unit of the tensor in the category $\mathbb{C}$.

Let Π be a proof of $\vdash [\Delta], \Gamma$. We define the GoI interpretation of Π, denoted by $[\![\Pi]\!]$, by induction on the length of the proof as follows.

1. Π is an *axiom* $\vdash A, A^\perp$, then $m = 0$, $n = 2$ and $[\![\Pi]\!] = s$.
2. Π is obtained using the *cut* rule on Π' and Π'' that is

$$\dfrac{\begin{array}{cc}\Pi' & \Pi'' \\ \vdots & \vdots \\ \vdash [\Delta'], \Gamma', A & \vdash [\Delta''], A^\perp, \Gamma''\end{array}}{\vdash [\Delta', \Delta'', A, A^\perp], \Gamma', \Gamma''}\ (cut)$$

   Define $[\![\Pi]\!]$ as follows: $[\![\Pi]\!] = \tau^{-1}([\![\Pi']\!] \otimes [\![\Pi'']\!])\tau$, where τ is a permutation.

3. Π is obtained using the *exchange* rule on the formulas A_i and A_{i+1} in Γ'. That is Π is of the form

$$\dfrac{\begin{array}{c}\Pi' \\ \vdots \\ \vdash [\Delta], \Gamma'\end{array}}{\vdash [\Delta], \Gamma}\ (exchange)$$

where in Γ' we have A_i, A_{i+1}. Then, $[\![\Pi]\!]$ is obtained from $[\![\Pi']\!]$ by interchanging the rows i and $i+1$. So suppose that $\Gamma' = \Gamma'_1, A_i, A_{i+1}, \Gamma'_2$, then $\Gamma = \Gamma'_1, A_{i+1}, A_i, \Gamma'_2$ and $[\![\Pi]\!] = \tau^{-1}[\![\Pi']\!]\tau$, where $\tau = 1_{\Gamma'_1} \otimes s \otimes 1_{\Gamma'_2 \otimes \Delta}$.

4. Π is obtained using an application of the *par* rule, that is Π is of the form:

$$\dfrac{\begin{array}{c}\Pi' \\ \vdots \\ \vdash [\Delta], \Gamma', A, B\end{array}}{\vdash [\Delta], \Gamma', A ⅋ B}\ (⅋)$$

Then $[\![\Pi]\!] = g[\![\Pi']\!]f$, where $f = 1_{\Gamma'} \otimes k \otimes 1_\Delta$ and $g = 1_{\Gamma'} \otimes j \otimes 1_\Delta$, recall that $U \otimes U \lhd U\,(j,k)$.

5. Π is obtained using an application of the *times* rule, that is Π has the form

$$\dfrac{\begin{array}{c}\Pi' \\ \vdots \\ \vdash [\Delta'], \Gamma', A\end{array} \quad \begin{array}{c}\Pi'' \\ \vdots \\ \vdash [\Delta''], \Gamma'', B\end{array}}{\vdash [\Delta', \Delta''], \Gamma', \Gamma'', A \otimes B}\ (times)$$

Then $[\![\Pi]\!] = g\tau^{-1}([\![\Pi']\!] \otimes [\![\Pi'']\!])\tau f$, where τ is a permutation and $f = 1_{\Gamma' \otimes \Gamma''} \otimes k \otimes 1_{\Delta' \otimes \Delta''}$ and $g = 1_{\Gamma' \otimes \Gamma''} \otimes j \otimes 1_{\Delta' \otimes \Delta''}$.

6. Π is obtained from Π' by an *of course* rule, that is Π has the form :

$$\dfrac{\begin{array}{c}\Pi' \\ \vdots \\ \vdash [\Delta], ?\Gamma', A\end{array}}{\vdash [\Delta], ?\Gamma', !A}\ (ofcourse)$$

Then $[\![\Pi]\!] = ((ue_U)^{\otimes n} \otimes u \otimes u^{\otimes 2m})\varphi^{-1}T((v^{\otimes n} \otimes 1_A \otimes 1_\Delta)[\![\Pi']\!](u^{\otimes n} \otimes 1_A \otimes 1_\Delta))\varphi((e'_U v)^{\otimes n} \otimes v \otimes v^{\otimes 2m})$, where $TT \lhd T\,(e, e')$, $|\Gamma'| = n, |\Delta| = 2m$, and $\varphi : (T^2U)^{\otimes n} \otimes TU \otimes (TU)^{\otimes 2m} \to T((TU)^{\otimes n} \otimes U \otimes U^{\otimes 2m})$ is the canonical isomorphism.

7. Π is obtained from Π' by the *dereliction* rule, that is Π is of the form :

$$\dfrac{\begin{array}{c}\Pi' \\ \vdots \\ \vdash [\Delta], \Gamma', A\end{array}}{\vdash [\Delta], \Gamma', ?A}\ (dereliction)$$

Then $[\![\Pi]\!] = (1_{\Gamma'} \otimes ud_U \otimes 1_\Delta)[\![\Pi']\!](1_{\Gamma'} \otimes d'_U v \otimes 1_\Delta)$ where $Id \lhd T\,(d, d')$.

8. Π is obtained from Π' by the *weakening* rule, that is Π is of the form:

$$\dfrac{\begin{array}{c}\Pi' \\ \vdots \\ \vdash [\Delta], \Gamma'\end{array}}{\vdash [\Delta], \Gamma', ?A}\ (weakening)$$

Then $[\![\Pi]\!] = (1_{\Gamma'} \otimes u w_U \otimes 1_{\Delta}) [\![\Pi']\!] (1_{\Gamma'} \otimes w'_U v \otimes 1_{\Delta})$, where $\mathcal{K}_I \lhd T\,(w, w')$.

9. Π is obtained from Π' by the *contraction* rule, that is Π is of the form :

$$\dfrac{\begin{array}{c}\Pi' \\ \vdots \\ \vdash [\Delta], \Gamma', ?A, ?A\end{array}}{\vdash [\Delta], \Gamma', ?A}\;(contraction)$$

Then $[\![\Pi]\!] = (1_{\Gamma'} \otimes u c_U (v \otimes v) \otimes 1_{\Delta}) [\![\Pi']\!] (1_{\Gamma'} \otimes (u \otimes u) c'_U v \otimes 1_{\Delta})$, where $T \otimes T \lhd T\,(c, c')$.

Example 2. Let Π be the following proof:

$$\dfrac{\vdash A^{\perp}, A \qquad \vdash A^{\perp}, A}{\vdash [A, A^{\perp}], A^{\perp}, A}\;(cut)$$

Then the GoI semantics of this proof is given by

$$[\![\Pi]\!] = \begin{bmatrix} 1\,0\,0\,0 \\ 0\,0\,0\,1 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \end{bmatrix} \begin{bmatrix} 0\,1\,0\,0 \\ 1\,0\,0\,0 \\ 0\,0\,0\,1 \\ 0\,0\,1\,0 \end{bmatrix} \begin{bmatrix} 1\,0\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \\ 0\,1\,0\,0 \end{bmatrix} = \begin{bmatrix} \mathbf{0} & Id_2 \\ Id_2 & \mathbf{0} \end{bmatrix}$$

where Id_2 is the 2×2 identity matrix and $\mathbf{0}$ is the 2×2 zero matrix.

4.1 Dynamics

Dynamics is at the heart of the GoI interpretation as compared to denotational semantics and it is hidden in the cut-elimination process. The mathematical model of cut-elimination is given by the *execution formula* defined as follows:

$$EX([\![\Pi]\!], \sigma) = Tr^{U^{2m}}_{U^n, U^n}((1_{U^n} \otimes \sigma)[\![\Pi]\!]) \tag{1}$$

where Π is a proof of the sequent $\vdash [\Delta], \Gamma$. Pictorially this can be represented as follows:

Note that $EX([\![\Pi]\!], \sigma)$ is a morphism from $U^n \to U^n$ and it always makes sense since the trace of any morphism in $\mathbb{C}(U^{2m+n}, U^{2m+n})$ is defined. Since we are working with a traced UDC with the standard trace, by Proposition 2 we can rewrite the execution formula (1) in a more familiar form:

$$EX([\![\Pi]\!], \sigma) = \pi_{11} + \sum_{n \geq 0} \pi_{12}(\sigma \pi_{22})^n (\sigma \pi_{21})$$

where $[\![\Pi]\!] = \begin{bmatrix} \pi_{11} & \pi_{12} \\ \pi_{21} & \pi_{22} \end{bmatrix}$. Note that the execution formula defined in this categorical framework *always* makes sense, that is we do not need a convergence criterion (e.g. nilpotency or weak nilpotency). This is in contrast to Girard's case where the infinite sum must be made to make sense and this is achieved via proving a nilpotency result.

We later show that formula (1) is the same as Girard's execution formula. The intention here is to prove that the result of this formula is what corresponds to the cut-free proof obtained from Π using Gentzen's cut-elimination procedure. We will also show that for any proof Π of MELL the execution formula is a finite sum, which corresponds to termination of computation as opposed to divergence.

Example 3. Consider the proof Π in Example 2 above. Recall also that $\sigma = s$ in this case ($m = 1$). Then

$$EX([\![\Pi]\!], \sigma) = Tr\left(\begin{bmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \end{bmatrix} \begin{bmatrix} 0&0&1&0 \\ 0&0&0&1 \\ 1&0&0&0 \\ 0&1&0&0 \end{bmatrix} \right)$$
$$= \begin{bmatrix} 0&0 \\ 0&0 \end{bmatrix} + \sum_{n \geq 0} \begin{bmatrix} 1&0 \\ 0&1 \end{bmatrix} \begin{bmatrix} 0&0 \\ 0&0 \end{bmatrix}^n \begin{bmatrix} 0&1 \\ 1&0 \end{bmatrix} = \begin{bmatrix} 0&1 \\ 1&0 \end{bmatrix}.$$

Note that in this case we have obtained the GoI interpretation of the cut-free proof obtained by applying Gentzen's Hauptsatz to the proof Π.

5 Soundness of the Interpretation

In this section we shall prove the main result of this paper: the soundness of the GoI interpretation. In other words we have to show that if a proof Π is reduced (via cut-elimination) to its cut-free form Π', then $EX([\![\Pi]\!], \sigma)$ is a finite sum and $EX([\![\Pi]\!], \sigma) = [\![\Pi']\!]$. Intuitively this says that if one thinks of cut-elimination as computation then $[\![\Pi]\!]$ can be thought of as an algorithm. The computation takes place as follows: if we run $EX([\![\Pi]\!], \sigma)$, it terminates after finitely many steps (cf. finite sum) and yields a datum (cf. cut-free proof). This intuition will be made precise in this section through the definition of type and the main theorems (see Theorems 1,2).

Lemma 1 (Associativity of cut). *Let Π be a proof of $\vdash [\Gamma, \Delta], \Lambda$ and σ and τ be the morphisms representing the cut-formulas in Γ and Δ respectively. Then*

$$EX([\![\Pi]\!], \sigma \otimes \tau) = EX(EX([\![\Pi]\!], \tau), \sigma)$$

Proof. Follows from naturality and vanishing II properties of trace.

We proceed to defining types. This and similar definitions are directly inspired by the corresponding ones in [10], generalising them to our categorical framework.

Definition 5. Let f, g be morphisms in $\mathbb{C}(U,U)$. We say that f is *nilpotent* if $f^k = 0$ for some $k \geq 1$. We say that f is *orthogonal* to g, denoted $f \perp g$ if gf is nilpotent. Orthogonality is a symmetric relation and it makes sense because 0_{UU} exists. Also, $0 \perp f$ for all $f \in \mathbb{C}(U,U)$.

Given a subset X of $\mathbb{C}(U,U)$, we define

$$X^{\perp} = \{f \in \mathbb{C}(U,U) | \forall g (g \in X \Rightarrow f \perp g)\}$$

A *type* is any subset X of $\mathbb{C}(U,U)$ such that $X = X^{\perp\perp}$. Note that types are inhabited, since 0_{UU} belongs to every type.

Definition 6. Consider a GoI situation $(\mathbb{C}, T, U)$ as above with j_1, j_2, k_1, k_2 components of j and k respectively. Let A be an MELL formula. We define *the GoI interpretation of* A, denoted θA, inductively as follows:

1. If $A \equiv \alpha$ that is A is an atom, then $\theta A = X$ an arbitrary type.
2. If $A \equiv \alpha^{\perp}$, $\theta A = X^{\perp}$, where $\theta\alpha = X$ is given by assumption.
3. If $A \equiv B \otimes C$, $\theta A = Y^{\perp\perp}$, where $Y = \{j_1 a k_1 + j_2 b k_2 | a \in \theta B, b \in \theta C\}$.
4. If $A \equiv B \parr C$, $\theta A = Y^{\perp}$, where $Y = \{j_1 a k_1 + j_2 b k_2 | a \in (\theta B)^{\perp}, b \in (\theta C)^{\perp}\}$.
5. If $A \equiv !B$, $\theta A = Y^{\perp\perp}$, where $Y = \{uT(a)v | a \in \theta B\}$.
6. If $A \equiv ?B$, $\theta A = Y^{\perp}$, where $Y = \{uT(a)v | a \in (\theta B)^{\perp}\}$.

It is an easy consequence of the definition that $(\theta A)^{\perp} = \theta A^{\perp}$ for any formula A.

Definition 7. Let $\Gamma = A_1, \cdots, A_n$. A *datum* of type $\theta\Gamma$ is a morphism $M : U^n \to U^n$ such that for any $\beta_1 \in \theta(A_1^{\perp}), \cdots, \beta_n \in \theta(A_n^{\perp})$, $(\beta_1 \otimes \cdots \otimes \beta_n)M$ is nilpotent. An *algorithm* of type $\theta\Gamma$ is a morphism $M : U^{n+2m} \to U^{n+2m}$ for some integer m such that for $\sigma : U^{2m} \to U^{2m}$ defined in the usual way, $EX(M, \sigma) = Tr((1 \otimes \sigma)M)$ is a finite sum and a datum of type $\theta\Gamma$.

Lemma 2. *Let $M : U^n \to U^n$ and $a : U \to U$. Define $CUT(a, M) = (a \otimes 1_{U^{n-1}})M : U^n \to U^n$. Note that the matrix representation of $CUT(a, M)$ is the matrix obtained from M by multiplying its first row by a. Then $M = [m_{ij}]$ is a datum of type $\theta(A, \Gamma)$ iff for any $a \in \theta A^{\perp}$, am_{11} is nilpotent and the morphism denoted $ex(CUT(a, M))$ and defined by $ex(CUT(a, M)) = Tr^A(s_{\Gamma,A}^{-1} CUT(a, M) s_{\Gamma,A})$ is in $\theta(\Gamma)$. Here $s_{\Gamma,A}$ is the symmetry morphism from $\Gamma \otimes A$ to $A \otimes \Gamma$.*

Theorem 1. *Let Γ be a sequent, and Π be a proof of Γ. Then $[\![\Pi]\!]$ is an algorithm of type $\theta\Gamma$.*

Theorem 2. *Let Π be a proof of a sequent $\vdash [\Delta], \Gamma$ in MELL. Then*

(i) *$EX([\![\Pi]\!], \sigma)$ is a finite sum.*
(ii) *If Π reduces to Π' by any sequence of cut-eliminations and "?" does not occur in Γ, then $EX([\![\Pi]\!], \sigma) = EX([\![\Pi']\!], \tau)$. So $EX([\![\Pi]\!], \sigma)$ is an invariant of reduction. In particular, if Π' is any cut-free proof obtained from Π by cut-elimination, then $EX([\![\Pi]\!], \sigma) = [\![\Pi']\!]$.*

6 Girard's Operator Algebraic model

In this section we observe that Girard's original C^*-algebra model (implementation) in GoI1 is captured in our categorical framework using the category $\mathbf{Hilb}_2$. First, recall [3] that $(\mathbf{PInj}, \mathbb{N}\times -, \mathbb{N})$ is a GoI situation.

Proposition 3. $(\mathbf{Hilb}_2, \ell^2\otimes -, \ell^2)$ *is a GoI Situation which agrees with Girard's C^*-algebraic model, where $\ell^2 = \ell_2(\mathbb{N})$. Its structure is induced via ℓ_2 from* $\mathbf{PInj}$.

We next show that Girard's execution formula agrees with ours. Note that in Girard's execution formula $[\![\Pi]\!]$ and σ are both $n+2m$ by $n+2m$ matrices. Also below $\tilde{\sigma} = s\otimes\cdots\otimes s$ (m-times.)

Proposition 4. *Let Π be a proof of $\vdash [\Delta], \Gamma$. Then in Girard's model above,*

$$(1-\sigma^2)\sum_{n=0}^{\infty}[\![\Pi]\!](\sigma[\![\Pi]\!])^n(1-\sigma^2) = Tr((1\otimes\tilde{\sigma})[\![\Pi]\!])$$

7 Conclusions and Further Work

In this paper we have given a categorical model for the GoI semantics of MELL and have proven the necessary theorems. We also showed that Girard's original operator algebra model fits this framework. We did not discuss the work by Abramsky and Jagadeesan [4] for the simple reason that it does not fit the unique decomposition category framework; that is, the category of domains does not form a UDC. This already suggests the necessity for a suitable generalization of the ideas presented in this paper. More precisely, we observe that the necessary ingredients for a categorical interpretation (model) are provided in the definition of a GoI Situation. However one still needs to give general meaning to the notions of *orthogonality* and *type* as well as provide a notion of "nilpotency", "finite sum" or "convergence". Observe that these notions found natural meanings in UDCs but a general traced category does not always have corresponding notions.

We should note that there are many concrete GoI situations based on partially additive categories; thus there are many models of this paper ([13]). However, to obtain *exactly* Girard's GoI 1, we also used Barr's ℓ_2 representation of $\mathbf{PInj}$ in $\mathbf{Hilb}$. We do not yet know of any operator-algebra representations for other models. That is an interesting open problem.

In [9], Girard addresses the issue of non-terminating algorithms and proves a convergence theorem for the execution formula (note that in this case nilpotency is out of the question). It would be interesting to see how this can be captured in our categorical framework where all existing infinite sums make sense. The challenge would be to have a means of distinguishing good and bad infinite sums, that is the ones corresponding to non-termination and to divergence.

Moreover in [11], Girard extended GoI to the full case, including the additives and constants. He also proved a nilpotency theorem for this semantics and its soundness (for a slightly modified sequent calculus) in the case of exponential-free conclusions. This too constitutes one of the main parts of our future work.

Last but certainly not least, we believe that GoI could be further used in its capacity as a new kind of semantics to analyze PCF and other fragments of functional and imperative languages and be compared to usual denotational and operational semantics through full abstraction theorems. The work on full completeness theorems for MLL via GoI in [12] is just a first step. Further related results, including those of Abramsky and Lenisa (e.g. [5]), should be examined.

References

1. Abramsky, S. (1996), Retracing Some Paths in Process Algebra. In *CONCUR 96, Springer LNCS* **1119**, 1-17.
2. Abramsky, S. (1997), *Interaction, Combinators and Complexity.* Lecture Notes, Siena, Italy.
3. Abramsky, S., Haghverdi, E. and Scott, P.J. (2002), Geometry of Interaction and Linear Combinatory Algebras. *MSCS*, vol. 12(5), 2002, 625-665, CUP.
4. Abramsky, S. and Jagadeesan, R. (1994), New Foundations for the Geometry of Interaction. *Information and Computation* **111** (1), 53-119.
5. Abramsky, S. and Lenisa, M. (2000), A Fully-complete PER Model for ML Polymorphic Types, *CSL'2000 Springer LNCS* **1862**, 140-155.
6. Barr, M. (1992), Algebraically Compact Functors. *JPAA*, vol. **82**, 211-231.
7. Danos, V. (1990), *La logique linéaire appliquée à l'étude de divers processus de normalisation et principalement du λ-calcul.* PhD thesis, Université Paris VII.
8. Danos, V. and Regnier, L. (1995), Proof-nets and the Hilbert Space. In: *Advances in Linear Logic*, London Math. Soc. Notes, **222**, CUP, 307–328.
9. Girard, J.-Y. (1988), Geometry of Interaction II: Deadlock-free Algorithms. In *Proc. of COLOG'88, LNCS* **417**, Springer, 76–93.
10. Girard, J.-Y. (1989a) Geometry of Interaction I: Interpretation of System F. In *Proc. Logic Colloquium 88,* North Holland, 221–260.
11. Girard, J.-Y. (1995), Geometry of Interaction III: Accommodating the Additives. In: *Advances in Linear Logic,* LNS **222**,CUP, 329–389,
12. Haghverdi, E. *A Categorical Approach to Linear Logic, Geometry of Proofs and Full Completeness*, PhD Thesis, University of Ottawa, Canada 2000.
13. Haghverdi, E. Unique Decomposition Categories, Geometry of Interaction and combinatory logic, *Math. Struct. in Comp. Science*, vol. **10**, 2000, 205-231.
14. Hasegawa, M. (1997), Recursion from Cyclic Sharing : Traced Monoidal Categories and Models of Cyclic Lambda Calculus, *Springer LNCS* **1210**, 196-213.
15. Hildebrandt, T. , Panangaden, P., Winskel, G, A Relational Model of Nondeterministic Dataflow, to appear in *Math. Struct. in Comp. Science*, 2004.
16. Hines, P. A categorical framework for finite state machines *Math. Struct. in Comp. Science*, vol. **13**, 2003, 451-480.
17. Joyal, A., Street, R. and Verity, D. (1996), Traced Monoidal Categories. *Math. Proc. Camb. Phil. Soc.* **119**, 447-468.
18. Manes, E.G. and Arbib, M.A. (1986), *Algebraic Approaches to Program Semantics.* Springer-Verlag.
19. Selinger, P. (1999), Categorical Structure of Asynchrony. *Electronic Notes in Theoretical Computer Science*, **20**. Elsevier Science B.V.
20. Stefanescu, G. *Network Algebra*, Springer-Verlag, 2000.