Roma, 10–12 February, 2003

LINEAR Meeting

Semantics: perspectives and open problems

Marina Lenisa
Dipartimento di Matematica e Informatica
University of Udine (Italy)

Semantic techniques connected to Linear Logic

Linear Logic has refreshed research in **denotational semantics**, by shedding new light on old semantic techniques and by inspiring entirely new approaches:

- **Static semantics**:
  - **Coherent spaces** with **stable functions**.
- **Dynamic semantics**:
  - **Geometry of Interaction**.
  - **Game Semantics**.
  - **Linear Realizability**.

The above semantics techniques allow for a finer analysis of **dynamic aspects** of **proofs** and **operational properties** of **programs**.

Full Completeness and Full Abstraction

A **categorical** model of a **logic** (**type theory**) is **fully complete** (**surjective**) if, for all formulae (types) A, B, all morphisms $f : [\![A]\!] \to [\![B]\!]$ are **denotations** of a **proof term** of the entailment $A \vdash B$. I.e. the interpretation functor from the category of syntax to the category of denotations is **full**.

A fully complete model of a type theory is **faithful** (**injective**) if it realizes the **syntactical theory**. I.e. the interpretation functor from the category of syntax to the category of denotations is **faithful**.

A model of a **programming language** is **fully abstract** w.r.t. an **observational equivalence** if it realizes the observational equivalence.

A fully abstract model of a programming language is **universal** if all semantical elements are denotations of syntactical elements.

Semantic Themes in the Linear Project

1. **Semantics of proofs** [**Roma: M.Pedicini, L.Tortora de Falco; Verona: G.Bellin, A.Fleury**]:

   - proof nets semantics;
   - categorical semantics of LL;
   - pragmatics from semantics.

2. **Semantics of programming languages** [**Torino: L.Paolini, S.Ronchi; Udine: P.Di Gianantonio, F.Honsell, M.Lenisa**]:

   - coherent semantics for PCF;
   - game semantics for (un)typed λ-calculus;
   - linear realizability;
   - Geometry of Interaction categories.

Task 1: Semantics of Proofs

Coherent semantics of LL proof(-nets)

Problem. Is Girard's **coherent semantics** of **proof-nets injective** (faithful)? I.e., does it determine the **syntactical equivalence** on proof-nets induced by cut-elimination?

[Tortora de Falco03] provides an answer on the basis of a new notion of **experiment**.

Injective obsessional experiments

Experiments [Girard87] allow to "effectively compute" the **semantics** of a **proof-net**. $[\![R]\!] = \{\gamma \in \mid \mathcal{P}\Gamma \mid :$ there exists an experiment e with result $\gamma\}$

The experiments introduced in [Tortora de Falco03] allow to prove that:

- **coherent semantics** is **injective** for the **(?⅋)LL fragment of LL**:

$$A \ ::= \ X \ \mid ?A \wp A \ \mid \ A \wp ?A \ \mid \ A \wp A \ \mid \ A \otimes A \ \mid !A$$

- **coherent semantics** is **not** injective for **MELL**.

Injectivity for larger fragments of LL?

	coherent semantics	multiset-based coherent semantics	relational semantics
MELL	NO	NO	?(yes)
MELL$\setminus\{?W\}$	?(yes)	?(yes)	?(yes)
LL$_{pol}$	?(yes)	?(yes)	?(yes)
(?$\wp$)LL	YES	YES	YES

where
MELL$\setminus\{?W\}$ is the subsystem of MELL containing all proof-nets whose normal forms do **not** contain any **weakening link**;
LL$_{pol}$ is the system of **polarized** proof-nets, i.e. the types of the conclusions are all subformulas of a positive or a negative formula. LL$_{pol}$ allows to encode classical logic.

Some open questions

- The **uniformity** of a semantics essentially amounts to the fact that an experiment is **uniquely** determined by its result.

  Is this a "good" property of a semantics? This issue is related to the extension of ludics to the exponential fragment.

- Is there a notion of experiment capturing **light proof-nets**? (Cfr. the stratified semantics of [Baillot03]).

Categorical semantics of LL [Fleury,Bellin]

Aim: to study categorical semantics of LL in a **general** setting, starting from [Fleury].

E.g. the exchange rule will be considered in **braided form**, and a corresponding **coherence theorem** for ribbon braided $*$-autonomous categories will be investigated.

This should help in clarifying the relations between **classical** and **intuitionistic** formulations of **non-commutative braided** rules.

Pragmatics from Semantics [Bellin]

Aim: to develop a **logical system** for **pragmatics**, based on a multi-modal classical semantics and an intuitionistic pragmatics, using linear implication to model causality and process interaction.

This should be based on an appropriate extension of Gödel-McKinsey-Tarski interpretation of intuitionistic logic into $S4$.

This system should suggest a new approach to the formalizations of **constructive** fragments of **classical** logic.

Applications in the field of the axiomatization of **rights** and **legal arguments** are expected.

Task 2: Semantics of Programming Languages

Coherent space semantics for PCF

The **Scott domain model** of **PCF** is fully abstract w.r.t. **PCF** extended with a **parallel-if** operator [Plotkin76].

Question:
Is there an extension of PCF for which the **coherent space model** is fully abstract?

[Paolini03]: the coherent space semantics is fully abstract w.r.t. PCF extended with a **Gustave operator** and a new **control operator test**.

Open Question:
Is it possible to refine the notion of stable function in order to rule out **test**-like functions?

Game Semantics

Game Semantics has been used for providing **fully abstract** and **fully complete** models for various **typed** and **untyped** languages:

- PCF [Abramsky-Jagadeesan-Malacaria94, Hyland-Ong94];
- (untyped) λ-calculus [Abramsky-McCusker95,DiGianantonio-Franco-Honsell99, Kehr-Nickau-Ong99,DiGianantonio01, Ong-DiGianantonio01];
- languages with locally-scoped references [Abramsky-McCuscker97,. . .];
- languages with control operators [Laird98].

Game Semantics of untyped λ-calculi

A negative result: all game models of the untyped λ-calculus realize either the theory $\mathcal{H}^*$ or the theory of **Böhm trees** or the theory of **Levy-Longo trees** [DiGianantonio-Franco-Honsell99,DiGianantonio-Franco00]. Cfr. **abstract machines** implementing **weak head reduction** [Curien-Herbelin96,Danos-Herbelin-Regnier96].

Other theories can be recovered by suitable "a posteriori" **quotient operations**: e.g. the **lazy observational equivalence**.

- In [Abramsky-McCusker95]: full abstraction for the lazy λ-calculus plus a (sequential) convergence test.

- In [DiGianantonio01]: full abstraction for the pure lazy λ-calculus in a category of [AJM]-games enriched with an **order relation**.

- In [Ong-DiGianantonio01]: universality for the pure lazy λ-calculus in a category of innocent games.

Current lines of research

- Are there game models which
  - capture Böhm trees up-to η?
  - do not equate all unsolvables of order 0?
- Use **ordered categories** for:
  - Modeling **call by-value** λ-theories.
  - In the typed setting: modeling **System F** (cfr. the game model of genericity of [Abramsky-Jagadeesan02]).
- Is there only a limited number of **typed** λ-theories captured by game models?

Linear Realizability

The **linear realizability** technique [Abramsky-L.99] amounts to constructing a category of **Partial Equivalence Relations** (**PERs**) over a **Linear Combinatory Algebra** (**LCA**).

- Clear and relatively simple construction.
- PERs yield models with **extensionality** properties.
- No need of further (extra) quotienting operations.

Linear Combinatory Algebra, [Abramsky96]

A **Linear Combinatory Algebra** is an applicative structure $(A, \bullet)$ with a **unary** (injective) **operation !**, and **combinators B**,**C**,**I**,**K**,**W**,**D**,δ,**F** such that

B xyz = x(yz)	**Cut**
C xyz = (xz)y	**Exchange**
I x = x	**Identity**
K x !y = x	**Weakening**
W x !y = x !y !y	**Contraction**
D !x = x	**Dereliction**
δ !x = !!x	**Comultiplication**
F !x !y = !(xy)	**Closed Functoriality**

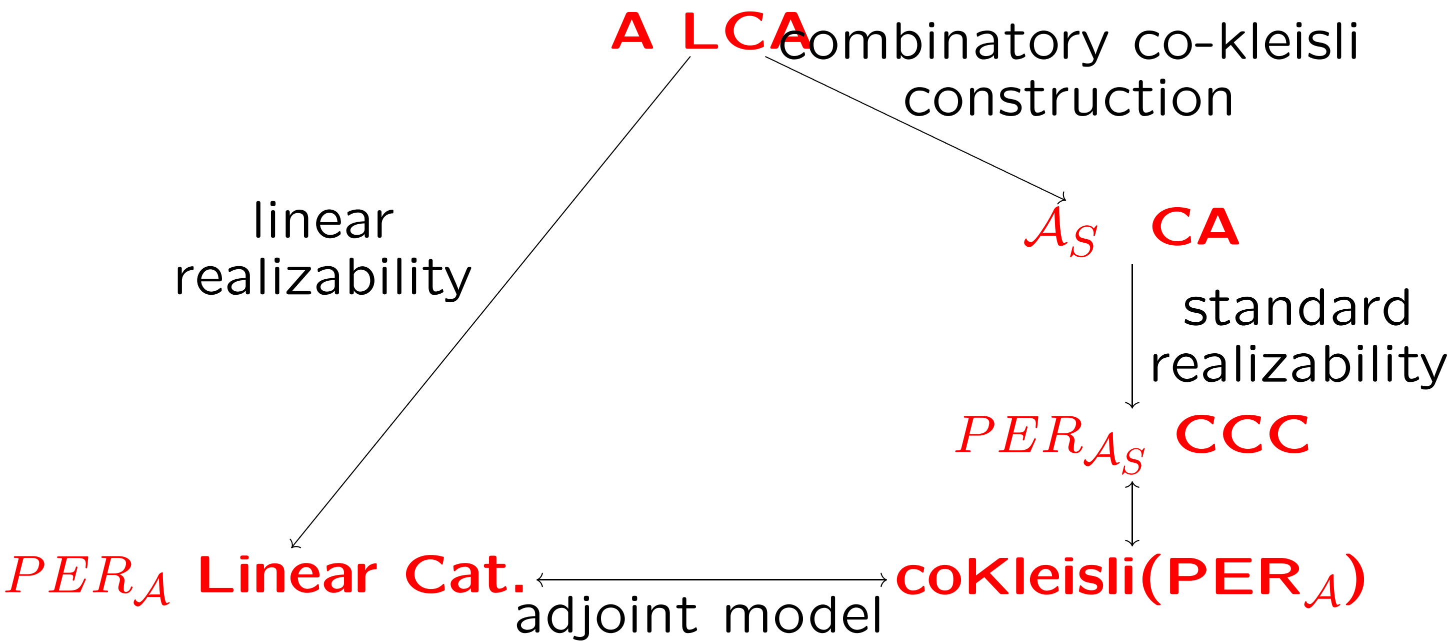
LCAs correspond to **Hilbert style** axiomatization of $\multimap$, **!** fragment of **LL**.

LCAs give rise to standard CAs, by defining:

$$\alpha \bullet_s \beta \ = \ \alpha \bullet !\beta \ .$$

Theorem [Abramsky-L.]

$\mathcal{A} = (A, \bullet, !)$ **Linear Combinatory Algebra** $\Longrightarrow$



$$R \to S \simeq\ !R \multimap S, \qquad !(R \times S) \simeq\ !R \otimes !S, \qquad !1 \simeq I.$$

Linear realizability has been used to provide **fully complete**/**fully abstract** models for various **typed λ-calculi**:

- The fragment of **System F** consisting of **ML types** [Abramsky-L.99];

- The **maximal theory** on the **simply typed λ-calculus** with finitely many ground constants [Abramsky-L.01];

- An **infinitary** version of the simply typed λ-calculus [Abramsky-L.01];

- **Unary PCF**;

- **PCF** [Abramsky-Longley00];

- ...

A "concrete" LCA [Abramsky97]

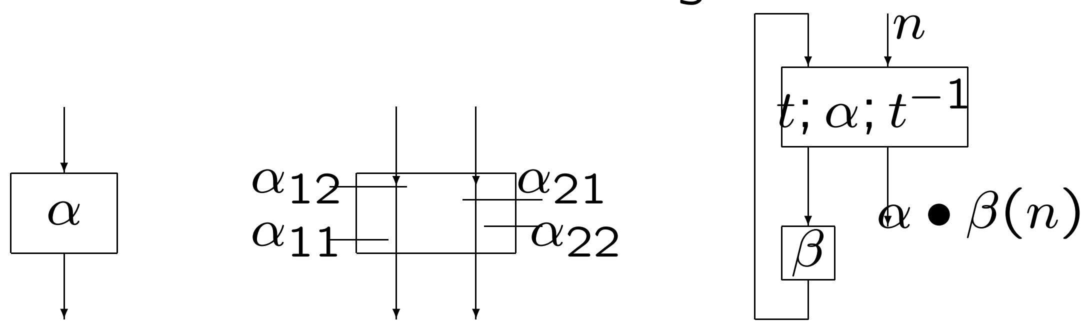
Define $\mathcal{A}_{Pfn} = ([\text{Nat} \rightharpoonup \text{Nat}], \bullet, !)$ as follows:

Fix two injective **codings**: $t : \text{Nat} + \text{Nat} \rightharpoonup \text{Nat}$, $p : \text{Nat} \times \text{Nat} \rightharpoonup \text{Nat}$.

Geometrical description of linear application:

$\alpha \bullet \beta$ is computed using a sort of **Execution Formula** [Girard].

In order to define the application $\alpha \bullet \beta$, for $\alpha, \beta \in [\text{Nat} \rightharpoonup \text{Nat}]$, we regard α as a two-input/two-output function via the coding t.



Algebraic description of linear application: $\alpha \bullet \beta = \alpha_{22} \cup \alpha_{21}; (\beta; \alpha_{11})^\star; \beta; \alpha_{12}$.

The operation !: it is intended to produce, from a single copy of α, **infinitely** many copies of α. These are obtained by tagging each of these copies with a natural number:

$$!\alpha = p^{-1}; (id_{\text{Nat}} \times \alpha); p \ .$$

An algebra of reversible computations

$\mathcal{A}_{PInv} \;=\; ([N \rightharpoonup_{Inv} N], \bullet, !)$

$\alpha : N \rightharpoonup N$ is an **involution** iff graph(f) is **symmetric**.

- $\mathcal{A}_{PInv}$ is an **highly constraint** algebra, in which all **computations** are **reversible** [Abramsky01].

- Partial involutions can be regarded as generalized **copy-cat strategies**.

- We can think of partial involutions as **abstract families** of **axiom links** as in the proof-nets of LL.

Open problems

- The theories captured by models of **partial involutions** are all **decidable**. Is there a connection between **decidability** of the theory and **reversibility** of the model? Can fully complete models help in proving decidability of syntactical theories?

- Extend the full completeness result to a larger class of System F types.

Axiomatic Full Completeness and Full Abstraction

Another line of research is that of **categorical axiomatizations** of **fully complete/fully abstract models**.

This is a useful **guide** in **factoring** proofs of full completeness and full abstraction. [Abramsky97, Abramsky-L.00, ...]

Many of such proofs follow a **common general pattern**

- in the typed setting, this is based on a **Decomposition Theorem** for strategies;
- in the untyped setting, this is based on an **analysis** of (finite parts of) strategies, corresponding to branches of trees.

Up-to now axiomatic f.c./f.a. has been explored for typed languages (PCF, simply typed λ-calculus, ...). One can probably develop appropriate axiomatizations for the untyped case.

Geometry of Interaction

Girard's original **Geometry of Interaction** gives an interpretation of linear logic, in which **cut-elimination** is modelled as a **dynamical process**, and which induces only a **weak equivalence** on proof terms.

Abramsky has developed a general **axiomatic categorical** framework for GoI. **Game categories** arise as instances of this general costruction.

What kind of models give Geometry of Interaction categories? **Linear Combinatory Algebras** [Abramsky97].

Are there **Geometry of Interaction** categories other than games which allow to capture a richer/different class of **untyped** λ-theories? Yes [Honsell-L.03].

Categorical Geometry of Interaction, [Abr96,AHP02]

$\mathcal{C}$ **traced symmetric monoidal** with T **traced strong monoidal functor**

$$\Downarrow$$

GoI category $\mathcal{G}(\mathcal{C})$ **compact closed** with **symmetric monoidal functor** !

where $\mathcal{G}(\mathcal{C})$ is defined by:

- **Objects**: (A^+, A^-), for A^+, A^- objects of $\mathcal{C}$.
- **Arrows**: $f : (A^+, A^-) \to (B^+, B^-)$ is $f : A^+ \otimes B^- \to A^- \otimes B^+$ in $\mathcal{C}$.
- **Composition**: **symmetric feedback**.
  For $f : (A^+, A^-) \to (B^+, B^-)$, $g : (B^+, B^-) \to (C+, C^-)$,
  $$g \circ f \triangleq Tr^{B^- \otimes B^+}_{A^+ \otimes C^-, A^- \otimes C^+}(\gamma' \circ (f \otimes g) \circ \gamma) \ .$$
- **Tensor**: $(A^+, A^-) \otimes (B^+, B^-) \triangleq (A^+ \otimes B^+, A^- \otimes B^-)$.
- **Bang**: $!(A^+, A^-) \triangleq (TA^+, TA^-)$.

Particle-style and Wave-style GoI

Particle-style GoI: tensor in $\mathcal{C}$ is **coproduct** and T is **countable copower**. Girard's GoI is an instance of this. Composition in the GoI category can be understood by simulating the flow of a particle around a network.

$(Rel, +, Nat \times (\ ))$ is the **basic setting** for particle-style GoI.

Trace in $(Rel, +)$: $Tr^U_{A,B}(\ ) : Rel(A+U, B+U) \rightarrow Rel(A,B)$,

$$Tr^U_{A,B}(f) \triangleq f_{AB} \cup f_{UB} f^*_{UU} f_{AU}\ .$$

Wave-style GoI: tensor in $\mathcal{C}$ is **product** and T is **countable power**. Composition in the GoI category is defined **globally** and **statically**.

$(Rel, \times, (\ )^\omega)$ is the **basic setting** for wave-style GoI.

Trace in $(Rel, \times)$: $Tr^U_{A,B}(\ ) : Rel(A \times U, B \times U) \rightarrow Rel(A,B)$,

$$Tr^U_{A,B}(f) \triangleq \{(a,b) \mid \exists u.\ (a,u,b,u) \in f\}\ .$$

GoI Situation [Abr97,AHS02]

$(\mathcal{C}, T, U)$ is a **GoI situation** if:

- $\mathcal{C}$ **traced symmetric monoidal** category;
- $T : \mathcal{C} \to \mathcal{C}$ **traced strong symmetric monoidal** functor with retractions:
  1. $e : TT \lhd T : e'$ **(Comultiplication)**
  2. $d : Id \lhd T : d'$ **(Dereliction)**
  3. $c : T \otimes T \lhd T : c'$ **(Contraction)**
  4. $w : K_I \lhd T : w'$ **(Weakening)**.
- U **GoI reflexive object** of $\mathcal{C}$, i.e.
  1. $\theta_1 : U \otimes U \lhd U : \theta_1'$
  2. $I \lhd U$
  3. $\theta_2 : TU \lhd U : \theta_2'$.

GoI Linear Combinatory Algebras, [Abr97,AHS02]

$(\mathcal{C}, T, U)$ **GoI situation** $\Longrightarrow$

- $(\mathcal{G}(\mathcal{C}), !)$ **weak linear category**;

- $(\mathcal{C}(U, U), \cdot, !)$ **linear combinatory algebra**, where:

$$f \cdot g \triangleq \mathsf{Tr}^{U}_{U,U}((\mathsf{id}_U \otimes g) \circ (\theta_1' \circ f \circ \theta_1))$$

$$!f \triangleq \theta_2 \circ Tf \circ \theta_2' \ .$$

Wave GoI algebras are graph models [Honsell-L.02]

Question: Are **wave** GoI categories richer than Game Categories?

Yes [Honsell-L.02]:

- Linear Combinatory Algebras arising from **wave GoI categories** are **graph models**.

- The λ-theories captured by wave GoI LCAs go beyond sensible and semi-sensible λ-theories.

Wave GoI Algebras on Rel

- For any GoI reflexive object U in Rel, $(Rel, \times, (\ )^\omega_f, U)$, is a **strict** (weak) **GoI situation**, where $(\ )^\omega_f$ is the functor of **streams** with **finite codomain**.

- The GoI category $\mathcal{G}(Rel, \times, (\ )^\omega_f)$ is a **strict weak linear category**.

- GoI algebras on $(Rel, \times, (\ )^\omega_f)$ are **strict linear graph models**. I.e. only a **restricted** K combinator is available, erasing **non**-bottom elements.

  Strict LGMs give models of the $\lambda\beta_{KN}$**-calculus**:
  $(\lambda x.M)N \to_{\beta_{KN}} M[N/x]$, if $(\lambda x.M)N$ is either a $I\beta$-redex or a $K\beta$-redex, where N is a **variable** or a **closed normal form**.

Wave GoI Algebras on Rel^*

- For any GoI reflexive object U in Rel^*, $(Rel^*, \times, (\ )_f^{*\omega}, U)$ is a (weak) **GoI situation**.

- The GoI category $\mathcal{G}(Rel^*, \times^*, (\ )_f^{*\omega})$ is a **weak linear category**.

- GoI algebras on $(Rel^*, \times^*, (\ )_f^{*\omega})$ are **pointed linear graph models**.

Open problems

- Is the whole class of λ-theories induced by **graph models** captured by **wave GoI algebras**?

- Given any particle GoI LCA, is it possible to recover a **λ-algebra** through a **quotient operation** similar to the one defined on games? (cfr. games without quotient)

- Are there **particle style** λ-algebras alternative to the ones based on games which yield a richer class of untyped λ-theories?