ProToCoLLo 2002-2004: Track presentation on semantics

Lorenzo Tortora de Falco
Università Roma Tre

What do we do when we do “semantics” ?

For the common sense, “semantics” has something to do with “meaning”.

In Logic and Theoretical Computer Science :
1) We cannot avoid the infinity (a possible reading of Gödel’s incompleteness theorem)
2) What is infinite has something to do with “semantics” (Gödel’s completeness theorem).

Semantics = a way to tame the infinity (which we cannot avoid).

Denotational semantics (and GOI) = a way to tame the infinity hidden in the interaction process.

The methods:
- “concrete” semantics: an infinite object is the infinite union of finite objetcs (Domain theory “algebraic DCPOs”, **Coherent semantics**,...)
- “abstract” semantics: what are the abstract properties needed to capture a given phenomenon, what is its internal structure (Category theory).

Our semantics: denotational semantics and GOI

Basic idea: a proof of $A \vdash B$ is a function from $\mathcal{A}$ to $\mathcal{B}$.

Denotational semantics: associate with every formula A some kind of mathematical structure $\mathcal{A}$ and with every proof π of A an element $[\pi]$ of $\mathcal{A}$ in such a way that

- if π and π' are two proofs/terms such that $\pi \twoheadrightarrow \pi'$ then $[\pi] = [\pi']$ **(invariant)**
- if $[\pi] = [\pi']$ and λ (resp. λ') is obtained from π (resp. π') by applying the same rule, then $[\lambda] = [\lambda']$ **(congruence)**
- there exists a formula A and two proofs/terms π, π' of A s.t. $[\pi] \neq [\pi']$ **(non trivial)**

Geometry of interaction: represent the calculus itself within a mathematical structure: there exists a formula EX (the execution formula) such that if π' is the normal form of π, then $EX([\pi]) = [\pi']$.

Why do we do semantics ?

It is interesting for several reasons:

- it allows to *understand* internal structure: Cartesian Closed Categories, Scott D_∞,...

- it allows to reveal hidden structure and to produce new objects: Linear Logic, proof-nets.

- it allows to address precise technical questions, often difficult ones, and the efforts we do to answer them make us advance in the understanding of the objects (proofs/terms/computational processes) we deal with:

  - faithfullness/separation/injectivity

  - full completeness/surjectivity

  - full abstraction.

The main questions

Full completeness (surjectivity): A model is fully complete (surjective) w.r.t. a proof/term system when all the inhabitants of the interpretation of a formula are interpretations of a proof.

Faithfullness (separation, injectivity): A model is faithfull (injective) w.r.t. a proof/term system when it realizes the syntactical theory (two different normal proofs/terms are interpreted by two different objects in the model).

Full abstraction: A model is fully abstract w.r.t. an observational equivalence of a proof/term system when it realizes the observational equivalence.

Observational equivalence and full abstraction for PCF

If s and t are PCF-terms of the same type, we say that s *observationally approximates* t (and we write $s \leq t$) when for every (suitably typed) context $C[\ ]$ and for every observable o (i.e. a constant -tt, ff, n-), if $C[s] \Downarrow o$ then $C[t] \Downarrow o$.

Two terms of the same type s and t are observationally equivalent ($t \sim_{obs} s$) if $s \leq t$ and $t \leq s$.

A model $[\ ]$ is fully abstract when $t \sim_{obs} s \iff [t] = [s]$.

External/internal separation : Injectivity/Böhm theorem

(D. Mazza, M. Pagani and L. Tortora de Falco, Roma)

[Tortora de Falco03]: studies the question of injectivity of coherent and relational semantics for Linear Logic proof-nets.

[Pagani04]: Böhm theorem for the simply typed λ-calculus in a purely syntactical framework (no models), following [Joly00].
$MAALP$ (free weakening on negative formulas) enjoys internal separation.

[Mazza Pagani04]: find a counterexample to the injectivity of coherent semantics for polarized Linear Logic ($MELL_{pol}$)

	internal separation	set-based coherent semantics	multiset-based coherent semantics	relational semantics
$MELL$	NO	NO	NO	? (yes)
$MELL\backslash\{?W\}$	?	? (yes)	? (yes)	? (yes)
$MELL_{pol}$	NO	NO	NO	? (yes)
$?\wp LL$	? (no)	? (yes)	YES	YES
$MAALP$	YES	YES	YES	YES
$\lambda MELL$	YES	YES	YES	YES

External separation : computing with injectivity

(L. Tortora de Falco, Roma)

In [Laurent Tortora de Falco04], injectivity allows to "compute with the additives in a sliced way" in LL_{pol}: let
- π be a sequent calculus proof and π_0 be any normal form of π semantically correct ($[\pi] = [\pi_0]$)
- S_π (resp. S_{π_0}) be the set of ($MELL$) slices of π (resp. π_0)
- S_0 be the (unique) normal form of S_π.

One has: $[S_0] = [S_\pi] = [\pi] = [\pi_0] = [S_{\pi_0}]$. By injectivity: $S_{\pi_0} = S_0$. And S_0 can be "semantically computed": first compute $[\pi]$, then "rebuild" the unique cut-free proof having $[\pi]$ as interpretation: it is $S_0 = S_{\pi_0}$, the sliced version of *any* normal form of π (technique similar to "Normalization by evaluation").

$$\begin{tikzcd} \pi \arrow[r] \arrow[d] & \pi_0 \arrow[d] \\ S_\pi \arrow[r] & S_{\pi_0} \end{tikzcd}$$

This diagram expresses a simulation property of the cut-elimination (in sequent calculus) by proof-net reductions. The injectivity property of the semantics allows to obtain such a result by *semantical* means.

External/internal separation (3): Open questions

1) Fill the holes in the previous slide (essentially: prove the injectivity property of relational semantics for $MELL$ and of coherent semantics for $MELL \setminus \{?W\}$)

2) Uniformity: it seems that in presence of both weakening and contraction "uniform" semantics are never injective while "non uniform" ones are always injective (see [Laurent03], where injectivity for LL_{pol} is proven w.r.t -non uniform- polarized games). This seems related to the fact that contractions can "hide" pieces of proofs which become interactively unobservable, *at least by using the standard (and uniform) notion of cut-elimination*. To be able to observe these "missing" subproofs, one might try to extend the syntax and thus modify the cut-elimination procedure (maybe inspired by "new" semantical ideas ?). Another way out might be to restrict the syntax by "forbidding" these "black holes" to occur.

Full abstraction and stability: coherent semantics for PCF

(L. Paolini, Torino)

Many people have studied how to extend a language in order to obtain the full-abstraction with respect to some models.

[Plotkin75]: the Scott domain model of PCF is not fully abstract w.r.t. PCF, but it becomes fully abstract if it is extended with a parallel-if operator.

[Jim and Meyer76]: they consider the particular case of stable functions (on dI-domains) and do not succed in extending PCF in order to turn the stable model into a fully abstract one. They show some negative results.

[Paolini03]: the coherent space semantics is not fully abstract w.r.t. PCF, but it becomes fully abstract if it is extended with a Gustave-like operator and a new control operator test.

Open Questions:

1) Is it possible to refine the notion of stable function in order to rule out test-like functions?
2) What relations are there between the new language and strongly stable functions?

Full abstraction and stability: coherent semantics for PCF

(L. Paolini, Torino)

[Paolini03]: the coherent space semantics is fully abstract w.r.t. PCF extended with a Gustave-like operator and a new control operator test, whose operational behavior is the following. Let $\Downarrow$ be our evaluation relation, let M be a closed term of type $\iota \to \iota$ (namely M applied to a numeral gives back a numeral or diverges) and let Ω_ι be a diverging term. Morally $test$ can be described with the two following rules:

1) if $M0 \Downarrow$ and $M\Omega_\iota \Uparrow$ then $testM \Downarrow 0$;
2) if $M0 \Downarrow$ and $M\Omega_\iota \Downarrow$ then $testM \Downarrow 1$
3) $testM$ is undefined otherwise.

That is: the application of $test$ to M tells us whether or not, in order to converge on 0 (to a numeral), the term M really uses 0.

In the paper, a more constructive operational description (based on an abstract machine) is given in details.

Strong Normalization and Lazy Lambda-calculus

(L. Paolini and E. Pimentel and S. Ronchi Della Rocca, Torino)

[Plotkin 75]: defines the call-by-value λ-calculus ($\lambda\beta_v$): β-reduction is performed only on values ($(\lambda x M)N \rightarrow M[N/x]$ iff $N \in \mathcal{V}$, where $\mathcal{V} ::= x|\lambda x t$). Redexes are called β_v-redexes.
Plotkin also introduces the lazy λ-calculus. A lazy β-redex is a β-redex which is not under the scope of a λ-abstraction.

[Paolini-Ronchi della Rocca 1999]: adapt the notion of solvability to $\lambda\beta_v$ and characterize in a logical way the $\lambda\beta_v$ solvable terms. They use the notion of "potentially valuable (p.v.) term": a term M is p.v. iff there exists a substitution s (of free variables by closed terms) s.t. $s(M)$ reduces to a value. The class of $\lambda\beta_v$ solvable terms is strictly contained in the class of p.v. terms.

[Paolini and Pimentel and Ronchi della Rocca 2004]: characterize the class of lazy strongly β-normalizing terms in a logical way (by means of a suitable intersection type assignement system). The call-by-value p.v. terms are exactly the lazy strongly β-normalizing terms (bridge between CBN and CBV evaluation).

Open question: is there a "λ-calculus" s.t. solvable=strongly β-normalizing ?

Full abstraction and Lazy Lambda-calculus (1)

(L. Paolini and S. Ronchi Della Rocca, Torino)

[Plotkin 75]: introduces the lazy λ-calculus, where evaluation is performed by reducing the outermost redex *provided it is not under the scope of a λ-abstraction*. It can be a β-redex (lazy CBN evaluation) or a β_v-redex (lazy CBV evaluation). Each of these evaluation strategies induces a pre-order relation ($M \leq_{op} N$ iff for every context $C[\,]$ s.t. $C[M]$ and $C[N]$ are closed, $C[M]$ reduces to an abstraction implies $C[N]$ reduces to an abstraction), and consequently an equivalence relation ($M \sim_{op} N$ iff $M \leq_{op} N$ and $N \leq_{op} M$).

[Abramski Ong 1993]: gives a denotational model (in the category of topological spaces) of the lazy CBN λ-calculus. The model is correct but not complete w.r.t. $\sim_{op}$, and a complete topological model does not exist.

Full abstraction and Lazy Lambda-calculus (2)

(L. Paolini and S. Ronchi Della Rocca, Torino)

[Egidi Honsell Ronchi Della Rocca 1992]: does the same for CBV λ-calculus. It gives a denotational model (in the category of topological spaces) of the lazy CBV λ-calculus. The model is correct but not complete w.r.t. $\sim_{op}$, and a complete topological model does not exist. A fully abstract model is given, using a notion of applicative bisimulation on terms. This technique cannot be applied in the CBN case.

[Paolini Ronchi Della Rocca 2004]: starting from a logical description of the two previous topological models, it defines a preorder relation stratified by types (in a uniform way for both models). Such a relation allows to build two fully abstract models, by collapsing all filters belonging to the same equivalence class.

Full abstraction/full completeness and game models

(P. Di Gianantonio, Udine)

Game semantics has been used (introduced ?) to prove full abstraction and full completeness for various typed and untyped languages (most notably to answer the longstanding question of full abstraction for PCF [Abramsky-Jagadeesan-Malacaria94, Hyland-Ong94]).

[Di Gianantonio01]: introduces a new category of games, the monotonic games, and prove full abstraction for the pure lazy λ-calculus (i.e. the lazy λ-calculus without constants).

[Di Gianantonio-Ong03]: gives a characterization of Lévy-Longo trees in terms of "games with persistence"

Current lines of research: 1) use games with persistence to obtain a model of **sistem F** (in [Abramsky-Jagadeesan03] a game model of F has been given, but the games they used are rather different)

2) use games to define an "abstract semantics" (in the style of [Malacaria-Hankin98] and [Abramsky-Ghiga-Murausky-Ong03]), this should apply to verification.

Game semantics for the nu-calculus

(P. Di Gianantonio, Udine)

Aim: broaden the use of game semantics to model other aspects of programming. More precisely: model the mechanism of creation of names.

The nu-calculus highlights this mechanism. nu-calculus= simply typed λ-calculus + new type for names + special constructor for the definition on new names. The only operation on names is comparison.

Several game models have been proposed for the nu-calculus ([Laird 04], [Abramski et al 04]), all using Fraenkel-Mostowski set theory (a Fraenkel-Mostowski set= a standard set + an equivalence relation induced by a finite permutation on names). These model are rather complex.

[Di Giannantonio 04]: uses a different approach and construct simple models, based on two main ideas:
1) Delegate the creation of new names to the environment (the Opponent), a new name is created by calling a function that at each call returns a different value (this is used in real world computing: Unix implements `gensym`, a function generating symbols, to be used for precisely that purpose).
2) Define an equivalence relation on games making it impossible for strategies to distinguish between names.

Combinatory algebras, λ-models, Geometry of Interaction

(Honsell and Lenisa, Udine)

Particle vs Wave Geometry of Interaction

There are two variants of "GOI category" $\mathcal{C}$:

- **Particle-style/dynamic**: tensor on $\mathcal{C}$ is a **coproduct**, ! is a **countable copower**.
  Composition in the GoI category can be understood by simulating the flow of a particle around a network.
  Instances: Girard's GoI, games à la [AJM96], LCAs of [Abramsky-L.99,00,01]

- **Wave-style/static**: tensor on $\mathcal{C}$ is a **product**, ! is a **countable power**.
  Composition in the GoI category is defined statically and globally.
  Instances: Abramsky-Mellies99 concurrent games, timeless games of Baillot-Danos-Ehrard-Regnier97.

Axiomatization of GOI and λ-models

[Abramsky-Haghverdi-Scott02] answers two questions:

1. define a general axiomatization of GOI

2. Of what is a given GOI interpretation a model (in the traditional sense) ?

Remember [Barendregt 1984]:

- A λ-model is a combinatory algebra which is (also) a wealkly extensional λ-algebra

- $\lambda \vdash t = u \iff t = u$ holds in all λ-models.

[Abramsky-Haghverdi-Scott02]:

- define a general axiomatic setting for both the “wave-style” and the “particle-style” GOI called a **Geometry of Interaction situation**: a traced symmetric monoidal category + extra structure (essentially, a strong monoidal endofunctor on the category, which is given by the countable co-power (resp. power) in particle-style (resp. wave-style) models)

- exhibit a construction allowing to associate with every GOI situation, successively: a weak linear category, a linear combinatory algebra, a combinatory algebra.

Linear combinatory algebras stands in the same relation to the Hilbert-style axiomatization of the $(!, \multimap)$-fragment of linear logic as standard combinatory algebras do to minimal logic (typed λ-calculus).

We can now answer question 2: a GOI interpretation gives rise to a (linear) combinatory algebra, then to a λ-model (with its equational theory).

Basic Settings for particle/wave Semantics

Abramsky suggested

- $(Rel, +)$ together with a **trace operator** as **basic setting** for **particle-style/dynamic semantics**.

- $(Rel, \times)$ together with (another) **trace operator** as **basic setting** for **wave-style/static (traditional) semantics**.

But: while $(Rel, +)$ has been widely studied, $(Rel, \times)$ and the connections between wave GoI and traditional semantics have not. In particular, the thesis that $(Rel, \times)$ is the basic setting for static semantics is not immediate, because $(Rel, \times)$ **escapes** Abramsky's **GoI axiomatization**.

Problems

- How does $(Rel, \times)$ fit in the general axiomatic pattern?

- What kind of wave models arise from $(Rel, \times)$?

- Do wave models induce a **richer/different class** of **untyped λ-theories** than **game** (**particle**) models?

Some Answers [Honsell-Lenisa03]

- We introduce a **weaker GoI axiomatization** (**weak GoI situation**). This allows to capture the category of **pointed relations** Rel^*.

- The GoI LCAs induced by Rel^* are **pointed linear graph models**.

- We introduce a notion of **strict (weak) GoI Situation**, in order to recover the case of $(Rel, \times)$.

- The GoI LCAs induced by Rel are **strict linear graph models**, i.e. models of **restricted** λ-calculi, such as the λI-calculus and the $\lambda\beta_{KN}$-calculus of [HL99].

- We show that, differently from game models, the class of λ-theories induced by wave models is quite rich, including also **non-sensible** (strict) λ-theories, i.e. not all unsolvable of order 0 are equated.

Others (F. Barbanera-U. de' Liguoro)

The theme is "semantics" of mobile calculi by means of logic.

[Barbanera-de' Liguoro03]: A semantics of mobile calculi modeling ambients carring code from object oriented languages is obtained by means of a type assignment system. This gives a logical interpretation of the calculus, which is adequate and complete.

Bibliography

[Tortora de Falco03]: "Obsessional experiments for Linear Logic Proof-nets", *Mathematical Structures in Computer Science* 13, (Issue 6) pp. 799 - 855, december 2003

[Pagani04]: "Separazione e Logica Lineare", Tesi di dottorato in Filosofia, Università Roma Tre, in preparazione

[Mazza Pagani04]: "A counterexample to the injectivity of coherent semantics in LL_{pol}", manuscript, 2004

[Laurent Tortora de Falco04]: "Slicing polarized additive normalization", to appear edited by *Cambridge University Press* in the "London Mathematical Society Lecture Notes Series", in the book "Linear Logic in Computer Science" (Thomas Ehrhard, Jean-Yves Girard, Paul Ruet and Phil Scott editors), 2004

[Paolini03]: "Lambda-theories: some investigations". PhD thesis in Computer Science, Università di Genova and Université de la Méditerranée, 2003

[Paolini and Pimentel and Ronchi della Rocca 2004]: "Lazy strong normalization", Proceedings of Workshops on Intersection Types and Related Systems (ITRS'04), Electronic Notes in Theoretical Computer Science, 2004.

[Paolini and Ronchi della Rocca 2004]: "Lazy logical semantics", Electronic Notes in Theoretical Computer Science, 2004.

[Di Gianantonio-Ong03]: "Games characterizing Levy-Longo trees", Proceedings of the International Colloquium on Automata Language and Programming (ICALP '02) LNCS 2380, pp. 476-487

[Di Giannantonio 04]: "Games for the ν-calculus", manuscript, 2004

[Honsell-Lenisa03]: "Wave-style Geometry of Interaction Models in Rel are Graph-like Lambda Models", TYPES'2003 Conference Proceedings, M. S. Berardi, M. Coppo, F. Damiani eds., Springer LNCS , vol. 3085, 2004

[Barbanera-de' Liguoro03]: "Type Assignment for Mobile Objects", in COMETA'0
ENTCS. Elsevier, 2003

Thematic workshop

The thematic workshop on semantics, organized by Lorenzo Tortora de Falco (Università Roma Tre), took place in Torino on Friday Septemper 17^{th} and Saturday Septemper 18^{th}. The main themes have been: 1) differential proof-nets 2) denotational semantics of "light" logics 3) the question of injectivity (or "semantical separation").

Invited speakers:
Patrick Baillot (CNRS-Université Paris 13)
Pierre Boudes (IML Marseille)
Thomas Ehrhard (CNRS-IML Marseille)
Olivier Laurent (CNRS-Université Paris 7).

Participants:
Patrick Baillot (CNRS-Université Paris 13), Gianluigi Bellin (Università di Verona), Pierre Boudes (IML Marseille), Daniel De Carvalho (IML Marseille), Ugo de' Liguoro (Università di Torino), Pietro Di Gianantonio (Università di Udine), Arnaud Fleury (Università di Verona), Olivier Laurent (CNRS-Université Paris 7), Damiano Mazza (IML Marseille e Università Roma Tre), Michele Pagani (Università Roma Tre), Luca Paolini (Università di Torino), Simona Ronchi della Rocca (Università di Torino), Luca Roversi (Università di Torino), Lorenzo Tortora de Falco (Università Roma Tre).