A survey on Geometry of Interactions

Carmine-Emanuele Cella

Università di Bologna

13 may 2010

References for G.o.I. (1)

- J.-Y. Girard. **Le point aveugle**, tome 1: vers la perfection, tome 2: vers limperfection. Visions des Sciences. Hermann, Paris, 2006-2007 (chapters 6, 9, 19).
- J.-Y. Girard: **Linear Logic, its syntax and semantics**. Advances in Linear Logic, eds Girard, Lafont, Regnier, London Mathematical Society Lecture Notes Series 222, Cambridge University Press 1995.
- J.-Y. Girard. **Towards a geometry of interaction**. In Categories in Computer Science and Logic, pages 69 108, Providence, 1989. American Mathematical Society. Proceedings of Symposia in Pure Mathematics n. 92.

References for G.o.I. (2)

- J.-Y. Girard. **Geometry of interaction I: interpretation of system F**. In Ferro, Bonotto, Valentini, and Zanardo, editors, Logic Colloquium 88, pages 221 260, Amsterdam, 1989. North-Holland.
- J.-Y. Girard. **Geometry of interaction II: deadlock-free algorithms**. In Martin-Löf and Mints, editors, Proceedings of COLOG 88, volume 417 of Lecture Notes in Computer Science, pages 76 93, Heidelberg, 1990. Springer-Verlag.
- J.-Y. Girard. **Geometry of interaction III: accommodating the additives**. In Girard, Lafont, and Regnier, editors, Advances in Linear Logic, pages 329 389, Cambridge, 1995. Cambridge University Press.

References for G.o.I. (3)

- J.-Y. Girard. **Geometry of interaction IV: the feedback equation**. In Stoltenberg-Hansen and Vnnen, editors, Logic Colloquium 03, pages 76 117. Association for Symbolic Logic, 2006.
- J.-Y. Girard. **Geometry of Interaction V: logic in the hyperfinite factor**. Fully revised version (October 2009).

Introduction

- Geometry of interactions is a *program for proof-theory*

Introduction

- Geometry of interactions is a *program for proof-theory*
- It develops a logic of *actions*, i.e. non-reusable facts (vs. situations)

Introduction

- Geometry of interactions is a *program for proof-theory*
- It develops a logic of *actions*, i.e. non-reusable facts (vs. situations)
- The meaning of a formula is in its proof, not in its truth

Introduction

- Geometry of interactions is a *program for proof-theory*
- It develops a logic of *actions*, i.e. non-reusable facts (vs. situations)
- The meaning of a formula is in its proof, not in its truth
- Each proof has and underlying **geometrical structure**

Introduction

- Geometry of interactions is a *program for proof-theory*
- It develops a logic of *actions*, i.e. non-reusable facts (vs. situations)
- The meaning of a formula is in its proof, not in its truth
- Each proof has and underlying **geometrical structure**
- G. of I. replaces static infinite situations by finite dynamic situations that lie in the geometrical structure of Gentzen's Hauptsatz

Basic logical commitments (1)

- Following Hilbert's program, it was possible to reduce any scientific activity to mathematics

Basic logical commitments (1)

- Following Hilbert's program, it was possible to reduce any scientific activity to mathematics
- It is only through very heavy and *ad-hoc* paraphrases that real implication may be put into mathemtics (usually by adding a parameter for time)

Basic logical commitments (1)

- Following Hilbert's program, it was possible to reduce any scientific activity to mathematics
- It is only through very heavy and *ad-hoc* paraphrases that real implication may be put into mathemtics (usually by adding a parameter for time)
- The logical laws extracted from mathematics are only adapted to eternal truth; the same principles applied in real life lead to absurdity because of the *interactive* nature of real implication

Basic logical commitments (2)

- Tertium non datur: $A \vee \neg A$

Basic logical commitments (2)

- Tertium non datur: $A \vee \neg A$
- It leaves no room for proof: if reality exists prior to anything else, proofs should be seen as a subjective process of understanding the real world

Basic logical commitments (2)

- Tertium non datur: $A \vee \neg A$
- It leaves no room for proof: if reality exists prior to anything else, proofs should be seen as a subjective process of understanding the real world
- Intuitionism proposed: *proofs as functions* (Heyting's semanthics)

Basic logical commitments (2)

- Tertium non datur: $A \vee \neg A$
- It leaves no room for proof: if reality exists prior to anything else, proofs should be seen as a subjective process of understanding the real world
- Intuitionism proposed: *proofs as functions* (Heyting's semanthics)
- Geometry of interactions proposes: *proofs as actions*

Basic logical commitments (2)

- Tertium non datur: $A \vee \neg A$
- It leaves no room for proof: if reality exists prior to anything else, proofs should be seen as a subjective process of understanding the real world
- Intuitionism proposed: *proofs as functions* (Heyting's semanthics)
- Geometry of interactions proposes: *proofs as actions*
- The logical twist from functions to actions leads to *linear logic*

Actions vs. situations

- Classical/intuitionistic logics deal with stable truth: if A and $A \Rightarrow B$ then B (A **still holds**)

Actions vs. situations

- Classical/intuitionistic logics deal with stable truth: if A and $A \Rightarrow B$ then B (A **still holds**)
- In real life implication is *causal*: cannot be iterated since conditions are modified after use (*reaction*)

Actions vs. situations

- Classical/intuitionistic logics deal with stable truth: if A and $A \Rightarrow B$ then B (A **still holds**)
- In real life implication is *causal*: cannot be iterated since conditions are modified after use (*reaction*)
- *Objection*: there are both in real life and in mathematics cases where reactions are negligible (*situations*)

Actions vs. situations

- Classical/intuitionistic logics deal with stable truth:
  if A and $A \Rightarrow B$ then B (A **still holds**)
- In real life implication is *causal*: cannot be iterated since conditions are modified after use (*reaction*)
- *Objection*: there are both in real life and in mathematics cases where reactions are negligible (*situations*)
- To cope with situations, special connectives (*exponentials*) are needed: ! and ?, where ! means infinite iterability

Actions vs. situations

- Classical/intuitionistic logics deal with stable truth: if A and $A \Rightarrow B$ then B (A **still holds**)
- In real life implication is *causal*: cannot be iterated since conditions are modified after use (*reaction*)
- *Objection*: there are both in real life and in mathematics cases where reactions are negligible (*situations*)
- To cope with situations, special connectives (*exponentials*) are needed: ! and ?, where ! means infinite iterability
- It is possible to define a new type of implication called *linear implication* ($\multimap$) such as:
  $A \Rightarrow B = (!A) \multimap B$

Operators (1)

- In linear logic, two conjunctions coexist: $\otimes$ and &

Operators (1)

- In linear logic, two conjunctions coexist: $\otimes$ and &
- They correspond to two radically different uses of the word *and*, expressing the availability of two actions: in $\otimes$ both will be done, while in & only one (that must be chosen) will be performed

Operators (1)

- In linear logic, two conjunctions coexist: $\otimes$ and &
- They correspond to two radically different uses of the word *and*, expressing the availability of two actions: in $\otimes$ both will be done, while in & only one (that must be chosen) will be performed
- Example: $A =$ to spend \$1, $B =$ to get a pack of Camels, $C =$ to get a pack of Marlboro

Operators (1)

- In linear logic, two conjunctions coexist: $\otimes$ and &
- They correspond to two radically different uses of the word *and*, expressing the availability of two actions: in $\otimes$ both will be done, while in & only one (that must be chosen) will be performed
- Example: $A =$ to spend \$1, $B =$ to get a pack of Camels, $C =$ to get a pack of Marlboro
- Given $A \rightarrow B$ and $A \rightarrow C$ it's impossible to form $A \rightarrow B \otimes C$ (getting two packs of cigarettes for \$1)

Operators (1)

- In linear logic, two conjunctions coexist: $\otimes$ and $\&$
- They correspond to two radically different uses of the word *and*, expressing the availability of two actions: in $\otimes$ both will be done, while in $\&$ only one (that must be chosen) will be performed
- Example: $A =$ to spend \$1, $B =$ to get a pack of Camels, $C =$ to get a pack of Marlboro
- Given $A \rightarrow B$ and $A \rightarrow C$ it's impossible to form $A \rightarrow B \otimes C$ (getting two packs of cigarettes for \$1)
- It's anyway possible to form $A \rightarrow B \& C$ (superposition of two actions)

Operators (1)

- In linear logic, two conjunctions coexist: $\otimes$ and $\&$
- They correspond to two radically different uses of the word *and*, expressing the availability of two actions: in $\otimes$ both will be done, while in $\&$ only one (that must be chosen) will be performed
- Example: $A =$ to spend \$1, $B =$ to get a pack of Camels, $C =$ to get a pack of Marlboro
- Given $A \rightarrow B$ and $A \rightarrow C$ it's impossible to form $A \rightarrow B \otimes C$ (getting two packs of cigarettes for \$1)
- It's anyway possible to form $A \rightarrow B \& C$ (superposition of two actions)
- This corresponds to **IF** ... **THEN** ... **ELSE**

Operators (2)

- Two disjunctions coexist: $\oplus$ (dual of &) and $⅋$ (dual of $\otimes$)

Operators (2)

- Two disjunctions coexist: $\oplus$ (dual of &) and $\parr$ (dual of $\otimes$)
- The meaning of $\oplus$ is to express two actions together

Operators (2)

- Two disjunctions coexist: $\oplus$ (dual of $\&$) and $⅋$ (dual of $\otimes$)
- The meaning of $\oplus$ is to express two actions together
- The meaning of $⅋$ is more complex and is related to *linear neagation*: $⅋$ is the symmetric form of linear implication

States and transitions (1)

- The notions of *state* has been overlooked in mathematics: eg. the current state of a Petri net, the current state of a Turing machine, the current position of a chessboard, the current list of beliefs in an expert system, . . .

States and transitions (1)

- The notions of *state* has been overlooked in mathematics: eg. the current state of a Petri net, the current state of a Turing machine, the current position of a chessboard, the current list of beliefs in an expert system, ...
- In all cases it is needed to introduce a temporal paramter to pass, say, from state S modelized as (S, t) to a new state modelized as $(S\prime, t + 1)$; this is very akward

States and transitions (1)

- The notions of *state* has been overlooked in mathematics: eg. the current state of a Petri net, the current state of a Turing machine, the current position of a chessboard, the current list of beliefs in an expert system, . . .
- In all cases it is needed to introduce a temporal paramter to pass, say, from state S modelized as (S, t) to a new state modelized as $(S\prime, t + 1)$; this is very akward
- It is better to represent states by formulas and trasitions by implications of states; in such a way S is accessible from $S\prime$ when $S \rightarrow S\prime$ is provable from the transitions, taken as axioms

States and transitions (1)

- The notions of *state* has been overlooked in mathematics: eg. the current state of a Petri net, the current state of a Turing machine, the current position of a chessboard, the current list of beliefs in an expert system, ...
- In all cases it is needed to introduce a temporal paramter to pass, say, from state S modelized as (S, t) to a new state modelized as $(S\prime, t+1)$; this is very akward
- It is better to represent states by formulas and trasitions by implications of states; in such a way S is accessible from $S\prime$ when $S \rightarrow S\prime$ is provable from the transitions, taken as axioms
- Example: **chemical equations**; in usual logic the phenomenon of *updating* cannot be represented

States and transitions (2)

- The rules of classical logic
  $A \Rightarrow A \wedge A$ and $A \wedge B \Rightarrow A$
  become wrong when $\Rightarrow$ is replaced by $\rightarrow$ and $\wedge$ is replaced by $\otimes$

States and transitions (2)

- The rules of classical logic
  $A \Rightarrow A \wedge A$ and $A \wedge B \Rightarrow A$
  become wrong when $\Rightarrow$ is replaced by $\rightarrow$ and $\wedge$ is replaced by $\otimes$
- The first says that proportions don't matter; the second would enable us to add an atom of carbon to the left that is not present on the right

States and transitions (2)

- The rules of classical logic
  $A \Rightarrow A \wedge A$ and $A \wedge B \Rightarrow A$
  become wrong when $\Rightarrow$ is replaced by $\multimap$ and $\wedge$ is replaced by $\otimes$
- The first says that proportions don't matter; the second would enable us to add an atom of carbon to the left that is not present on the right
- The first is a way of writing *contraction*, while the second is a way of writing *weakening*

States and transitions (2)

- The rules of classical logic
  $A \Rightarrow A \wedge A$ and $A \wedge B \Rightarrow A$
  become wrong when $\Rightarrow$ is replaced by $\rightarrow$ and $\wedge$ is replaced by $\otimes$
- The first says that proportions don't matter; the second would enable us to add an atom of carbon to the left that is not present on the right
- The first is a way of writing *contraction*, while the second is a way of writing *weakening*
- Example: **formal grammars**; it shows clearly the connection between linear logic and computation processes, that's why linear logic finds a natural application in computer science

States and transitions (3)

- *Theory = linear logic + axioms + current state*

States and transitions (3)

- *Theory* = *linear logic* + *axioms* + *current state*
- The axioms are forever, the current state us available for a single use

States and transitions (3)

- *Theory* = *linear logic* + *axioms* + *current state*
- The axioms are forever, the current state us available for a single use
- When a state has been used to prove another state, the **theory is updated** and the proved state becomes the *current* state

Linear negation

- Linear negation ($\perp$) is the only negative operation of linear logic

Linear negation

- Linear negation ($\perp$) is the only negative operation of linear logic
- Can be defined as: $A \rightarrow B = A\perp \invamp B$

Linear negation

- Linear negation ($\perp$) is the only negative operation of linear logic
- Can be defined as: $A \rightarrow B = A\perp \parr B$
- It behaves like transposition in linear algebra ($A \rightarrow B$ is the same as $B\perp \rightarrow A\perp$)

Linear negation

- Linear negation ($\perp$) is the only negative operation of linear logic
- Can be defined as: $A \rightarrow B = A\perp \invamp B$
- It behaves like transposition in linear algebra ($A \rightarrow B$ is the same as $B\perp \rightarrow A\perp$)
- It expresses *duality*: *action of type* A = *reaction of type* $A\perp$

Linear negation

- Linear negation ($\perp$) is the only negative operation of linear logic
- Can be defined as: $A \rightarrow B = A\perp ⅋ B$
- It behaves like transposition in linear algebra ($A \rightarrow B$ is the same as $B\perp \rightarrow A\perp$)
- It expresses *duality*: *action of type* $A =$ *reaction of type* $A\perp$
- Property: $A = A\perp\perp$ (like in classical logic)

Linear negation

- Linear negation ($\perp$) is the only negative operation of linear logic
- Can be defined as: $A \rightarrow B = A\perp \parr B$
- It behaves like transposition in linear algebra ($A \rightarrow B$ is the same as $B\perp \rightarrow A\perp$)
- It expresses *duality*: *action of type* $A =$ *reaction of type* $A\perp$
- Property: $A = A\perp\perp$ (like in classical logic)
- It is more primitive (stronger) than usual negation

Sequent calculus

- Introduced by Gentzen in 1934 as a tool for studying basic laws of logic

Sequent calculus

- Introduced by Gentzen in 1934 as a tool for studying basic laws of logic
- It is made of three major structural rules: *weakening*, *contraction*, *exchange*

Sequent calculus

- Introduced by Gentzen in 1934 as a tool for studying basic laws of logic
- It is made of three major structural rules: *weakening*, *contraction*, *exchange*
- **Weakening**: it speaks of causes without effects; it allows not to use all hypothesis in a deduction

Sequent calculus

- Introduced by Gentzen in 1934 as a tool for studying basic laws of logic
- It is made of three major structural rules: *weakening*, *contraction*, *exchange*
- **Weakening**: it speaks of causes without effects; it allows not to use all hypothesis in a deduction
- **Contraction**: it speaks about infinity; it says that what you have, you will always keep no matter how you use it

Sequent calculus

- Introduced by Gentzen in 1934 as a tool for studying basic laws of logic
- It is made of three major structural rules: *weakening*, *contraction*, *exchange*
- **Weakening**: it speaks of causes without effects; it allows not to use all hypothesis in a deduction
- **Contraction**: it speaks about infinity; it says that what you have, you will always keep no matter how you use it
- **Exchange**: it expresses the commutativity of multiplicative operators; it exists to achieve more expressive power

Linear sequent calculus (1)

- By combining structural rules we obtain different kinds of logic

Linear sequent calculus (1)

- By combining structural rules we obtain different kinds of logic
- C+W+E = classical logic

Linear sequent calculus (1)

- By combining structural rules we obtain different kinds of logic
- C+W+E = classical logic
- W+E = affine logic

Linear sequent calculus (1)

- By combining structural rules we obtain different kinds of logic
- C+W+E = classical logic
- W+E = affine logic
- E = linear logic

Linear sequent calculus (1)

- By combining structural rules we obtain different kinds of logic
- C+W+E = classical logic
- W+E = affine logic
- E = linear logic
- nothing = non-commutative linear logic

Linear sequent calculus (2)

- It is possible to define a full sequent calculus for linear logic, made of structural, identity and logical groups

Linear sequent calculus (2)

- It is possible to define a full sequent calculus for linear logic, made of structural, identity and logical groups
- Structural group will be made only of *exchange*

Linear sequent calculus (2)

- It is possible to define a full sequent calculus for linear logic, made of structural, identity and logical groups
- Structural group will be made only of *exchange*
- Idendity group will be made of identity axiom and of the cut-rule

Linear sequent calculus (2)

- It is possible to define a full sequent calculus for linear logic, made of structural, identity and logical groups
- Structural group will be made only of *exchange*
- Idendity group will be made of identity axiom and of the cut-rule
- Logical group will contain additives (disjunctions), multiplicatives (conjunctions) and exponentials (modalities)

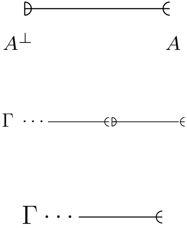
Linear sequent calculus (2)

- It is possible to define a full sequent calculus for linear logic, made of structural, identity and logical groups
- Structural group will be made only of *exchange*
- Idendity group will be made of identity axiom and of the cut-rule
- Logical group will contain additives (disjunctions), multiplicatives (conjunctions) and exponentials (modalities)
- The only dynamical feature of the system is the cut-rule: it puts together an action and a reaction of the same type

Linear sequent calculus (3)

The calculus consist of replacing hypothesis with conclusions (*axioms*) and vice-versa (*cut-rule*):

An electrical representation of which is:



Proof-structures

If we accept two additional links:

$$\frac{A \qquad B}{A \otimes B}\ (\textit{times link}) \qquad\qquad \frac{A \qquad B}{A \parr B}\ (\textit{par link})$$

then we can associate any proof in a linear sequent calculus a graph-like proof structure.

Proof-nets

- A proof-structure is nothing but a graph whose vertices are formulas and whose edges are links; moreover each formula is the conclusion of exactly one link and premise of at most one link

Proof-nets

- A proof-structure is nothing but a graph whose vertices are formulas and whose edges are links; moreover each formula is the conclusion of exactly one link and premise of at most one link
- The formulas which are not premises are *conclusions* of the structure

Proof-nets

- A proof-structure is nothing but a graph whose vertices are formulas and whose edges are links; moreover each formula is the conclusion of exactly one link and premise of at most one link
- The formulas which are not premises are *conclusions* of the structure
- Inside proof-structures let's call *proof-nets* those which can be obtained as the interpretation of sequent calculus proofs

Cut-elimination for proof-nets

- The keypoint is to find an *independent* characterization of proof-nets (to get rid of syntax)

Cut-elimination for proof-nets

- The keypoint is to find an *independent* characterization of proof-nets (to get rid of syntax)
- This can be achieved by handling syntactic manipulations at the level of proof-nets

Cut-elimination for proof-nets

- The keypoint is to find an *independent* characterization of proof-nets (to get rid of syntax)
- This can be achieved by handling syntactic manipulations at the level of proof-nets
- This is done by means of the *cut-elimination* procedure, that has good properties:
  - it enjoys the Church-Rosser property (confluence);
  - it is linear in time;
  - the treatment of multiplicative fragment is purely local; in fact all cut-links can be simultaneously eliminated

Coherent semantics

- The only existant semantics for mathematics are denotational (i.e. static), such as Scott semantics

Coherent semantics

- The only existant semantics for mathematics are denotational (i.e. static), such as Scott semantics
- These semantics interpret proofs as functions instead of actions

Coherent semantics

- The only existant semantics for mathematics are denotational (i.e. static), such as Scott semantics
- These semantics interpret proofs as functions instead of actions
- There is no name for such *operational semantics*, that's why Girard introduced the expression **GEOMETRY OF INTERACTIONS**

Coherent semantics

- The only existant semantics for mathematics are denotational (i.e. static), such as Scott semantics
- These semantics interpret proofs as functions instead of actions
- There is no name for such *operational semantics*, that's why Girard introduced the expression **GEOMETRY OF INTERACTIONS**
- The inadequation of the denotational semantics for computation becomes conspicuous if we note that such semantics will have a strong tendency to be infinite, whereas programs are finite dynamical processes

Against subjectivism

- What is closest to the idea of dynamics is syntax, which makes all the necessary distinctions of sense and is finite

Against subjectivism

- What is closest to the idea of dynamics is syntax, which makes all the necessary distinctions of sense and is finite
- This leads to the opposition: *geometry / taxonomy (categories)*

Against subjectivism

- What is closest to the idea of dynamics is syntax, which makes all the necessary distinctions of sense and is finite
- This leads to the opposition: *geometry / taxonomy (categories)*
- The problem with syntax is that it contains irrelevant informations of temporal nature

Against subjectivism

- What is closest to the idea of dynamics is syntax, which makes all the necessary distinctions of sense and is finite
- This leads to the opposition: *geometry / taxonomy (categories)*
- The problem with syntax is that it contains irrelevant informations of temporal nature
- It is therefore needed to find what's hidden behind syntax *without* going to denotation: a *non-subjectivistic approach to sense*

Against subjectivism

- What is closest to the idea of dynamics is syntax, which makes all the necessary distinctions of sense and is finite
- This leads to the opposition: *geometry / taxonomy (categories)*
- The problem with syntax is that it contains irrelevant informations of temporal nature
- It is therefore needed to find what's hidden behind syntax *without* going to denotation: a *non-subjectivistic approach to sense*
- There should be a purely *geometrical* notion of finite dynamical structure; in other words there should be a *geometrical* interpretation of Gentzen Hauptsatz

Phase space

- A *phase space* is a pair $(M, \perp)$ where M is a commutative monoid (multiplicative) and $\perp$ is a subset of M

Phase space

- A *phase space* is a pair $(M, \perp)$ where M is a commutative monoid (multiplicative) and $\perp$ is a subset of M
- Given two subsets X, Y of M it is possible to define $X \rightarrow Y := \{m \in M; \forall n \in X; mn \in Y\}$

Phase space

- A *phase space* is a pair $(M, \perp)$ where M is a commutative monoid (multiplicative) and $\perp$ is a subset of M
- Given two subsets X, Y of M it is possible to define $X \rightarrow Y := \{m \in M; \forall n \in X; mn \in Y\}$
- It's also possible to define for each subset X of M its *orthogonal*: $X\perp := X \rightarrow \perp$

Phase space

- A *phase space* is a pair $(M, \perp)$ where M is a commutative monoid (multiplicative) and $\perp$ is a subset of M
- Given two subsets X, Y of M it is possible to define $X \rightarrow Y := \{m \in M; \forall n \in X; mn \in Y\}$
- It's also possible to define for each subset X of M its *orthogonal*: $X\perp := X \rightarrow \perp$
- A *fact* is any subset of M equal to its orthogonal; it is immediate that $X \rightarrow Y$ is a fact is Y is a fact

Interpretation of the connectives

It's possible to interpret all the operations in linear logic by operations on facts:

1. **times** : $X \otimes Y := \{mn \,;\, m \in X \wedge n \in Y\}^{\perp\perp}$
2. **par** : $X ⅋ Y := (X^{\perp} \otimes Y^{\perp})^{\perp}$
3. **1** : $\mathbf{1} := \{1\}^{\perp\perp}$, where 1 is the neutral element of M
4. **plus** : $X \oplus Y := (X \cup Y)^{\perp\perp}$
5. **with** : $X \& Y := X \cap Y$
6. **zero** : $\mathbf{0} := \emptyset^{\perp\perp}$
7. **true** : $\top := M$
8. **of course** : $!X := (X \cap I)^{\perp\perp}$, where I is the set of idempotents of M which belong to **1**
9. **why not** : $?X := (X^{\perp} \cap I)^{\perp}$

It's easily seen that this semantics is sound and complete.

Coherent spaces

- A *coherent space* is a reflexive undirected graph. In other terms it consists of a set $|X|$ of atoms together with compatibility or *coherence* relation between atoms, noted $x \smile\frown y$ or $y \smile\frown x[modX]$ if there is any ambiguity as to X

Coherent spaces

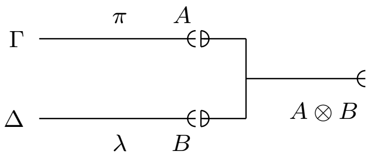
- A *coherent space* is a reflexive undirected graph. In other terms it consists of a set $|X|$ of atoms together with compatibility or *coherence* relation between atoms, noted $x \smile\frown y$ or $y \smile\frown x[modX]$ if there is any ambiguity as to X
- Given a coherent space X, its linear negation $X\perp$ is defined by:
  - $|X\perp| = |X|$
  - $x \smile\frown y[modX\perp]$ **iff** $x \smile\frown y[modX]$

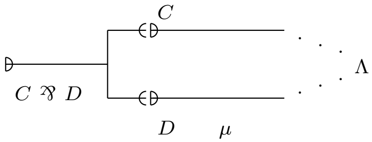
Coherent spaces

- A *coherent space* is a reflexive undirected graph. In other terms it consists of a set $|X|$ of atoms together with compatibility or *coherence* relation between atoms, noted $x \smile\frown y$ or $y \smile\frown x[modX]$ if there is any ambiguity as to X
- Given a coherent space X, its linear negation $X\perp$ is defined by:
  - $|X\perp| = |X|$
  - $x \smile\frown y[modX\perp]$ **iff** $x \smile\frown y[modX]$
- It is possible to define in this way all operators of linear logic (multiplicatives, additives, exponentials) to create a complete semantics

Geometry of interactions (1)

In a previous example we represented the links for proof-nets as electrical plugs; this is possible for all rules in linear logic, such as $\otimes - rule$ and $\invamp - rule$:





Geometry of interactions (2)

- If we interpret the electrical encoders as $\otimes-$ or $\invamp-$ link, we get a very precise modelization of cut-elimination of proof-nets

Geometry of interactions (2)

- If we interpret the electrical encoders as $\otimes-$ or $\parr-$ link, we get a very precise modelization of cut-elimination of proof-nets
- Since this coding is based on the development by means of Fourier series (which involve Hilbert space), everything that was done can be formulated in terms of operator algebras

Geometry of interactions (3)

- The general formula for cut-elimination (*execution formula*) is the following:
  $EX(u, \phi) := (1 - \phi^2)u(1 - \phi u)^{-1}(1 - \phi^2)$

Geometry of interactions (3)

- The general formula for cut-elimination (*execution formula*) is the following:
  $EX(u, \phi) := (1 - \phi^2)u(1 - \phi u)^{-1}(1 - \phi^2)$
- This gives the interpretation of the elimination of cuts (represented by ϕ) in a proof represented by u

Geometry of interactions (3)

- The general formula for cut-elimination (*execution formula*) is the following:
  $EX(u, \phi) := (1 - \phi^2)u(1 - \phi u)^{-1}(1 - \phi^2)$
- This gives the interpretation of the elimination of cuts (represented by ϕ) in a proof represented by u
- Termination of the process is interpreted as the nilpotency of ϕu and the part $u(1 - \phi u)^{-1}$ is a candidate for the execution

Interpretation of System F (1)

Given M and n it's possible to define the oriented graph $G_n(M)$ such as:

- Nodes: λ-abstraction and applications (box nodes)

Interpretation of System F (1)

Given M and n it's possible to define the oriented graph $G_n(M)$ such as:

- Nodes: λ-abstraction and applications (box nodes)
- Edges: labelled with weight

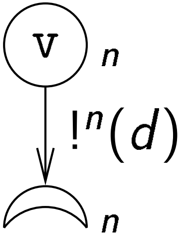
Interpretation of System F (1)

Given M and n it's possible to define the oriented graph $G_n(M)$ such as:

- Nodes: λ-abstraction and applications (box nodes)
- Edges: labelled with weight
- One exiting edge per free variable plus one entering edge for M

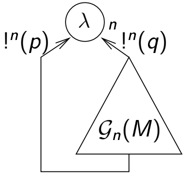
Interpretation of System F (2)

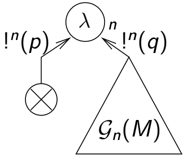
Variable case:



Interpretation of System F (3)

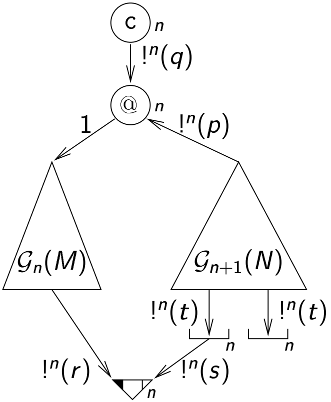
Abstraction:





Interpretation of System F (4)

Application:



Comments

- Geometry of interactions is a semantics for linear logic

Comments

- Geometry of interactions is a semantics for linear logic
- Its basic idea is to remove taxonomy (i.e. temporality) by means of something like proof-nets, in order to give the maximum degree of freedom for the execution of a program

Comments

- Geometry of interactions is a semantics for linear logic
- Its basic idea is to remove taxonomy (i.e. temporality) by means of something like proof-nets, in order to give the maximum degree of freedom for the execution of a program
- Logical rules should not be symmetric w.r.t. time

Comments

- Geometry of interactions is a semantics for linear logic
- Its basic idea is to remove taxonomy (i.e. temporality) by means of something like proof-nets, in order to give the maximum degree of freedom for the execution of a program
- Logical rules should not be symmetric w.r.t. time
- Processes communicates without understanding each other (by means of global operations such as erasing, duplicating, sending back)