Injectivity of relational semantics for (connected) MELL proof-structures via the Taylor expansion

Giulio Guerrieri[1] Luc Pellissier[2] Lorenzo Tortora de Falco[3]

[1]PPS, Université Paris Diderot

[2]LIPN, Université Paris 13

[3]Dipartimento di Matematica e Fisica, Università Roma Tre

Topology, Algebra and Categories in Logic 2015 (TACL 2015)

Ischia, June 21, 2015

Outline

1. Introduction: linear logic in a nutshell

2. From linear logic to differential linear logic and Taylor expansion

3. The question of injectivity

Outline

1. Introduction: linear logic in a nutshell
2. From linear logic to differential linear logic and Taylor expansion
3. The question of injectivity

Proofs representations: from numbers to graphs (1 of 3)

- Hilbert/Gödel ($\sim$1920-30): a proof is a finite sequence of formulas (through Gödelization a finite sequence of integers and thus an integer)
  $\rightsquigarrow$ no structure, no dynamics

- Gentzen (1936): a proof (in sequent calculus, natural deduction) is a finite *tree* with an internal *dynamics*, cut-elimination $\to_{\text{cut}}$. This led to:
  - Curry-Howard correspondence ($\sim$1960): e.g. propositional intuitionistic minimal logic/simply typed λ-calculus, 2^{nd} order intuitionistic logic/system $\mathcal{F}$, propositional classical logic/$\lambda\mu$-calculus, etc.
    - ★ formula $A \rightsquigarrow$ type A
    - ★ proof π of $A \Rightarrow B \rightsquigarrow$ program π with input of type A and output of type B
    - ★ cut-elimination $\rightsquigarrow$ execution
  - denotational semantics ($\sim$1970): it is concerned with the mathematical meaning of proofs/programs. Goals:
    - ★ to provide mathematical tools for proving properties of proofs/programs
    - ★ to suggest new features to add to the syntax of logic/programming languages.

    The general pattern is (in categorical terms)
    - ★ formula/type $A \rightsquigarrow$ an object $\mathcal{A}$ is some category $\mathbf{C}$
    - ★ proof/program π of $A \Rightarrow B \rightsquigarrow$ a morphism $[\![\pi]\!] : \mathcal{A} \to \mathcal{B}$ in $\mathbf{C}$
    - ★ invariance under cut-elimination/execution: if $\pi \to_{\text{cut}} \pi'$ then $[\![\pi]\!] = [\![\pi']\!]$

Proofs representations: from numbers to graphs (1 of 3)

- Hilbert/Gödel ($\sim$1920-30): a proof is a finite sequence of formulas (through Gödelization a finite sequence of integers and thus an integer)
  $\rightsquigarrow$ no structure, no dynamics

- Gentzen (1936): a proof (in sequent calculus, natural deduction) is a finite *tree* with an internal *dynamics*, cut-elimination $\rightarrow_{\text{cut}}$. This led to:
  - Curry-Howard correspondence ($\sim$1960): e.g. propositional intuitionistic minimal logic/simply typed λ-calculus, 2nd order intuitionistic logic/system $\mathcal{F}$, propositional classical logic/$\lambda\mu$-calculus, etc.
    - formula $A \rightsquigarrow$ type A
    - proof π of $A \Rightarrow B \rightsquigarrow$ program π with input of type A and output of type B
    - cut-elimination $\rightsquigarrow$ execution
  - denotational semantics ($\sim$1970): it is concerned with the mathematical meaning of proofs/programs. Goals:
    - to provide mathematical tools for proving properties of proofs/programs
    - to suggest new features to add to the syntax of logic/programming languages.

    The general pattern is (in categorical terms)
    - formula/type $A \rightsquigarrow$ an object $\mathcal{A}$ is some category $\mathbf{C}$
    - proof/program π of $A \Rightarrow B \rightsquigarrow$ a morphism $[\![\pi]\!] : \mathcal{A} \to \mathcal{B}$ in $\mathbf{C}$
    - invariance under cut-elimination/execution: if $\pi \rightarrow_{\text{cut}} \pi'$ then $[\![\pi]\!] = [\![\pi']\!]$

Proofs representations: from numbers to graphs (1 of 3)

- Hilbert/Gödel (∼1920-30): a proof is a finite sequence of formulas (through Gödelization a finite sequence of integers and thus an integer)
  $\rightsquigarrow$ no structure, no dynamics

- Gentzen (1936): a proof (in sequent calculus, natural deduction) is a finite *tree* with an internal *dynamics*, cut-elimination $\to_{\mathsf{cut}}$. This led to:
  - Curry-Howard correspondence (∼1960): e.g. propositional intuitionistic minimal logic/simply typed λ-calculus, 2^{nd} order intuitionistic logic/system $\mathcal{F}$, propositional classical logic/$\lambda\mu$-calculus, etc.
    - ⋆ formula $A \rightsquigarrow$ type A
    - ⋆ proof π of $A \Rightarrow B \rightsquigarrow$ program π with input of type A and output of type B
    - ⋆ cut-elimination $\rightsquigarrow$ execution
  - denotational semantics (∼1970): it is concerned with the mathematical meaning of proofs/programs. Goals:
    - ⋆ to provide mathematical tools for proving properties of proofs/programs
    - ⋆ to suggest new features to add to the syntax of logic/programming languages.

    The general pattern is (in categorical terms)
    - ⋆ formula/type $A \rightsquigarrow$ an object $\mathcal{A}$ is some category $\mathbf{C}$
    - ⋆ proof/program π of $A \Rightarrow B \rightsquigarrow$ a morphism $[\![\pi]\!] : \mathcal{A} \to \mathcal{B}$ in $\mathbf{C}$
    - ⋆ invariance under cut-elimination/execution: if $\pi \to_{\mathsf{cut}} \pi'$ then $[\![\pi]\!] = [\![\pi']\!]$

Proofs representations: from numbers to graphs (1 of 3)

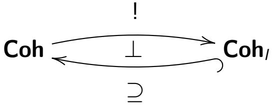
- Hilbert/Gödel ($\sim$1920-30): a proof is a finite sequence of formulas (through Gödelization a finite sequence of integers and thus an integer)
  $\leadsto$ no structure, no dynamics

- Gentzen (1936): a proof (in sequent calculus, natural deduction) is a finite *tree* with an internal *dynamics*, cut-elimination $\to_{\mathsf{cut}}$. This led to:
  - Curry-Howard correspondence ($\sim$1960): e.g. propositional intuitionistic minimal logic/simply typed λ-calculus, 2^{nd} order intuitionistic logic/system $\mathcal{F}$, propositional classical logic/$\lambda\mu$-calculus, etc.
    - ⋆ formula $A \leadsto$ type A
    - ⋆ proof π of $A \Rightarrow B \leadsto$ program π with input of type A and output of type B
    - ⋆ cut-elimination $\leadsto$ execution
  - denotational semantics ($\sim$1970): it is concerned with the mathematical meaning of proofs/programs. Goals:
    - ⋆ to provide mathematical tools for proving properties of proofs/programs
    - ⋆ to suggest new features to add to the syntax of logic/programming languages.

    The general pattern is (in categorical terms)
    - ⋆ formula/type $A \leadsto$ an object $\mathcal{A}$ is some category $\mathbf{C}$
    - ⋆ proof/program π of $A \Rightarrow B \leadsto$ a morphism $[\![\pi]\!] : \mathcal{A} \to \mathcal{B}$ in $\mathbf{C}$
    - ⋆ invariance under cut-elimination/execution: if $\pi \to_{\mathsf{cut}} \pi'$ then $[\![\pi]\!] = [\![\pi']\!]$

Proofs representations: from numbers to graphs

- Girard (1987): a proof is a graph, cut-elimination is graph rewriting
  - ▸ denotational semantics of intuitionistic logic/λ-calculus: let **Coh** (resp. $\mathbf{Coh}_l$) be the category of coherence spaces and stable $\to_{\mathsf{st}}$ (resp. linear $\multimap$) functions.
    - ⋆ formula/type $A \rightsquigarrow$ a coherence space $\mathcal{A}$
    - ⋆ proof/program π of $A \Rightarrow B \rightsquigarrow$ a stable function $[\![\pi]\!]^{\mathbf{Coh}} : \mathcal{A} \to_{\mathsf{st}} \mathcal{B}$
    - ⋆ invariance under cut-elimination/execution: if $\pi \to_{\mathsf{cut}} \pi'$ then $[\![\pi]\!]^{\mathbf{Coh}} = [\![\pi']\!]^{\mathbf{Coh}}$



$\rightsquigarrow$ $\mathbf{Coh}[A, B] = \mathbf{Coh}_l[!A, B]$, more precisely $f : \mathcal{A} \to_{\mathsf{st}} \mathcal{B} = f : !\mathcal{A} \multimap \mathcal{B}$

  - ▸ logical formulas: the operations ! and $\multimap$ can be internalized in the syntax $\rightsquigarrow$ *Linear Logic* (LL) where:
    - ⋆ structural rules (contraction, weakening) acquire a logical status thanks to a pair of duals modalities, the *exponentials* ! and ?
    - ⋆ usual connectives split into *multiplicative* and *additive*
    - ⋆ classical and intuitionistic logic can be embedded into LL, for example

$$A \Rightarrow B = !A \multimap B$$

Proofs representations: from numbers to graphs (2 of 3)

- Girard (1987): a proof is a graph, cut-elimination is graph rewriting
  - denotational semantics of intuitionistic logic/λ-calculus: let **Coh** (resp. $\mathbf{Coh}_l$) be the category of coherence spaces and stable $\to_{\mathsf{st}}$ (resp. linear $\multimap$) functions.
    - ⋆ formula/type $A \leadsto$ a coherence space $\mathcal{A}$
    - ⋆ proof/program π of $A \Rightarrow B \leadsto$ a stable function $[\![\pi]\!]^{\mathbf{Coh}} : \mathcal{A} \to_{\mathsf{st}} \mathcal{B}$
    - ⋆ invariance under cut-elimination/execution: if $\pi \to_{\mathsf{cut}} \pi'$ then $[\![\pi]\!]^{\mathbf{Coh}} = [\![\pi']\!]^{\mathbf{Coh}}$

$$\mathbf{Coh} \underset{\supseteq}{\overset{!}{\underset{\longleftarrow}{\overset{\longrightarrow}{\perp}}}} \mathbf{Coh}_l$$

$\leadsto$ $\mathbf{Coh}[A, B] = \mathbf{Coh}_l[!A, B]$, more precisely $f : \mathcal{A} \to_{\mathsf{st}} \mathcal{B} = f : !\mathcal{A} \multimap \mathcal{B}$

  - logical formulas: the operations ! and $\multimap$ can be internalized in the syntax $\leadsto$ *Linear Logic* (LL) where:
    - ⋆ structural rules (contraction, weakening) acquire a logical status thanks to a pair of duals modalities, the *exponentials* ! and ?
    - ⋆ usual connectives split into *multiplicative* and *additive*
    - ⋆ classical and intuitionistic logic can be embedded into LL, for example

$$A \Rightarrow B = !A \multimap B$$

Proofs representations: from numbers to graphs (3 of 3)

- Girard (1987): a proof is a graph, cut-elimination is graph rewriting (continued)
  - semantics of proofs in LL:
    - ⋆ proof/program π of $!A \multimap B \rightsquigarrow$ a stable function $[\![\pi]\!]^{\mathsf{Coh}} \colon \mathcal{A} \to_{\mathsf{st}} \mathcal{B}$
      $= $ a linear function $[\![\pi]\!]^{\mathsf{Coh}_l} \colon !\mathcal{A} \multimap \mathcal{B}$

      Idea: the proof π may use the hypothesis A an arbitrary number of times
      i.e. the program π may call its arguments of type A at will
    - ⋆ proof/program π of $A \multimap B \rightsquigarrow$ a linear function $[\![\pi]\!]^{\mathsf{Coh}_l} \colon \mathcal{A} \multimap \mathcal{B}$

      Idea: the proof π uses the hypothesis A linearly (exactly once)
      i.e. the program π calls its arguments of type A linearly (exactly once)
  - representations of proofs in LL: proofs in LL become particular graphs (*proof-nets*) among more general graphs (*proof-structures*)
  - cut-elimination in LL: cut-elimination can be defined on proof-structures; the modality ! marks a resource erasable and duplicable *at will*, on proof structures a ! corresponds to a *box*

- Girard (1987): a proof is a graph, cut-elimination is graph rewriting (continued)
  - semantics of proofs in LL:
    - ⋆ proof/program π of $!A \multimap B \rightsquigarrow$ a stable function $[\![\pi]\!]^{\mathsf{Coh}} : \mathcal{A} \to_{\mathsf{st}} \mathcal{B}$
      $=$ a linear function $[\![\pi]\!]^{\mathsf{Coh}_l} : !\mathcal{A} \multimap \mathcal{B}$

      Idea: the proof π may use the hypothesis A an arbitrary number of times
      i.e. the program π may call its arguments of type A at will
    - ⋆ proof/program π of $A \multimap B \rightsquigarrow$ a linear function $[\![\pi]\!]^{\mathsf{Coh}_l} : \mathcal{A} \multimap \mathcal{B}$

      Idea: the proof π uses the hypothesis A linearly (exactly once)
      i.e. the program π calls its arguments of type A linearly (exactly once)
  - representations of proofs in LL: proofs in LL become particular graphs (*proof-nets*) among more general graphs (*proof-structures*)
  - cut-elimination in LL: cut-elimination can be defined on proof-structures; the modality ! marks a resource erasable and duplicable *at will*, on proof structures a ! corresponds to a *box*

Proofs representations: from numbers to graphs (3 of 3)

- Girard (1987): a proof is a graph, cut-elimination is graph rewriting (continued)
  - semantics of proofs in LL:
    - ⋆ proof/program π of $!A \multimap B \rightsquigarrow$ a stable function $[\![\pi]\!]^{\mathsf{Coh}} : \mathcal{A} \rightarrow_{\mathsf{st}} \mathcal{B}$
      $= $ a linear function $[\![\pi]\!]^{\mathsf{Coh}_l} : !\mathcal{A} \multimap \mathcal{B}$

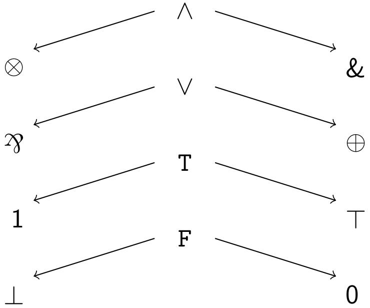
      Idea: the proof π may use the hypothesis A an arbitrary number of times
      i.e. the program π may call its arguments of type A at will
    - ⋆ proof/program π of $A \multimap B \rightsquigarrow$ a linear function $[\![\pi]\!]^{\mathsf{Coh}_l} : \mathcal{A} \multimap \mathcal{B}$

      Idea: the proof π uses the hypothesis A linearly (exactly once)
      i.e. the program π calls its arguments of type A linearly (exactly once)
  - representations of proofs in LL: proofs in LL become particular graphs (*proof-nets*) among more general graphs (*proof-structures*)
  - cut-elimination in LL: cut-elimination can be defined on proof-structures; the modality ! marks a resource erasable and duplicable *at will*, on proof structures a ! corresponds to a *box*

The syntax of Linear Logic (LL, Girard [1987])

Contraction and weakening are allowed only on formulas of the form $?A$

$\rightsquigarrow$ For the lack of unrestricted structural rules, connectives and units are split



multiplicative — monoidal structure

additive — cartesian structure

MELL = multiplicative and exponential (?, !) fragment of LL (no additives)

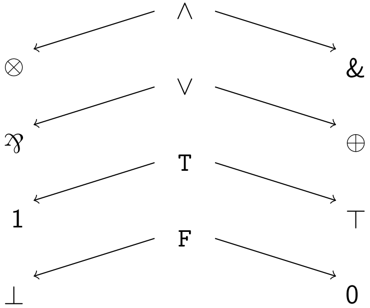
$$A, B ::= X \mid X^{\perp} \mid 1 \mid \perp \mid A \otimes B \mid A \parr B \mid !A \mid ?A$$

Negation is involutive. $A^{\perp}$ is defined by induction using the usual De Morgan identities

The syntax of Linear Logic (LL, Girard [1987])

Contraction and weakening are allowed only on formulas of the form $?A$

$\rightsquigarrow$ For the lack of unrestricted structural rules, connectives and units are split



multiplicative	additive
monoidal structure	cartesian structure

MELL = multiplicative and exponential (?, !) fragment of LL (no additives)

$$A, B ::= X \mid X^{\perp} \mid 1 \mid \perp \mid A \otimes B \mid A \parr B \mid !A \mid ?A$$

Negation is involutive. $A^{\perp}$ is defined by induction using the usual De Morgan identities

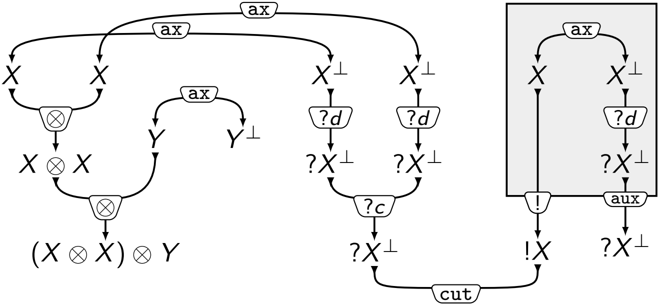
From MELL sequent calculus to MELL-proof structures

Example: the following proof of MELL sequent calculus . . .

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{}{\vdash X, X^{\perp}}\,ax
\qquad
\dfrac{}{\vdash X, X^{\perp}}\,ax
}{\vdash X \otimes X, X^{\perp}, X^{\perp}}\,\otimes
\qquad
\dfrac{}{\vdash Y, Y^{\perp}}\,ax
}{\vdash (X \otimes X) \otimes Y, X^{\perp}, X^{\perp}, Y^{\perp}}\,\otimes
}{\vdash (X \otimes X) \otimes Y, ?X^{\perp}, X^{\perp}, Y^{\perp}}\,?d
}{\vdash (X \otimes X) \otimes Y, ?X^{\perp}, ?X^{\perp}, Y^{\perp}}\,?d
}{\vdash (X \otimes X) \otimes Y, ?X^{\perp}, Y^{\perp}}\,?c
\qquad
\dfrac{
\dfrac{
\dfrac{}{\vdash X, X^{\perp}}\,ax
}{\vdash X, ?X^{\perp}}\,?d
}{\vdash !X, ?X^{\perp}}\,!
}{\vdash (X \otimes X) \otimes Y, ?X^{\perp}, Y^{\perp}}\,cut
$$

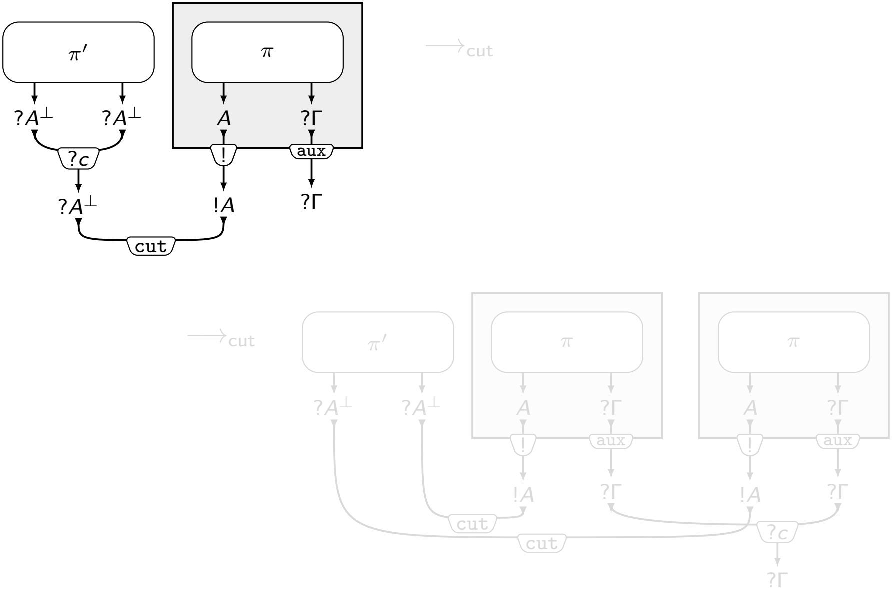
From MELL sequent calculus to MELL-proof structures

Example: ...corresponds to the following MELL proof-net



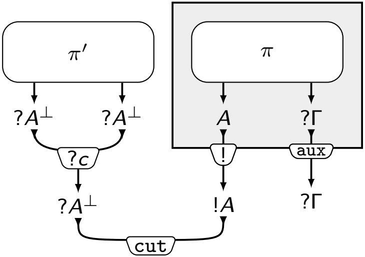
!/? cut-elimination steps in MELL-proof structures (1 of 2)

contraction/promotion: duplication of a resource



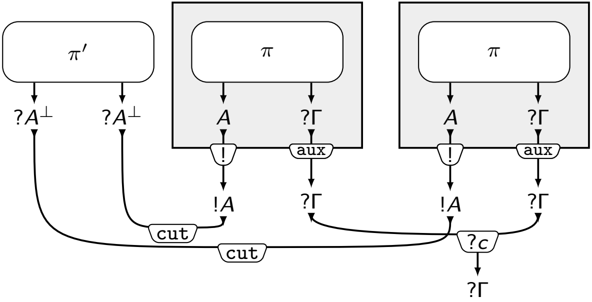
!/? cut-elimination steps in MELL-proof structures

contraction/promotion: duplication of a resource



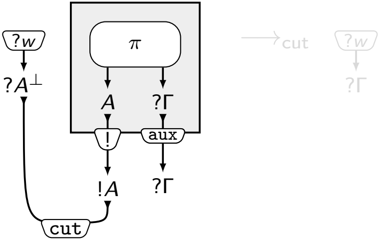
$\longrightarrow_{\text{cut}}$

$\longrightarrow_{\text{cut}}$



!/? cut-elimination steps in MELL-proof structures (2 of 2)

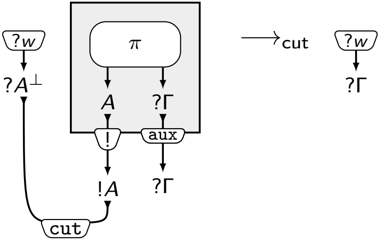
weakening/promotion: erasure of a resource



dereliction/promotion: access to a resource

!/? cut-elimination steps in MELL-proof structures (2 of 2)

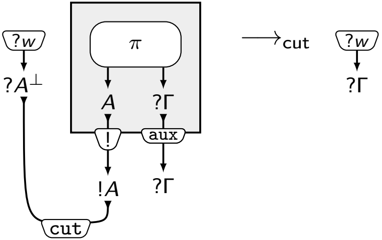
weakening/promotion: erasure of a resource



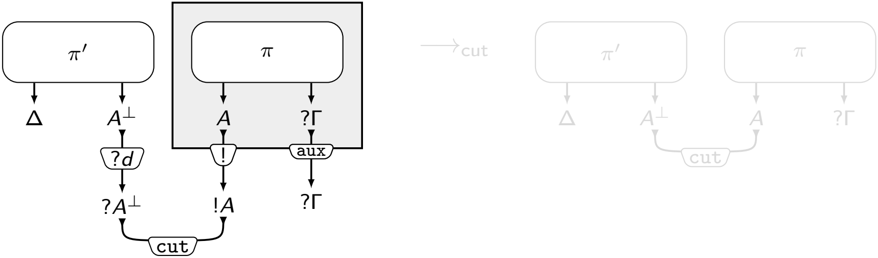
dereliction/promotion: access to a resource

!/? cut-elimination steps in MELL-proof structures

weakening/promotion: erasure of a resource

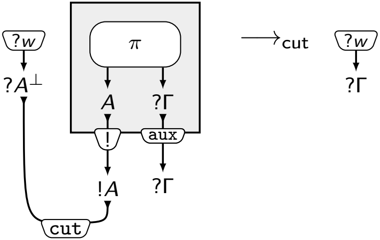


dereliction/promotion: access to a resource

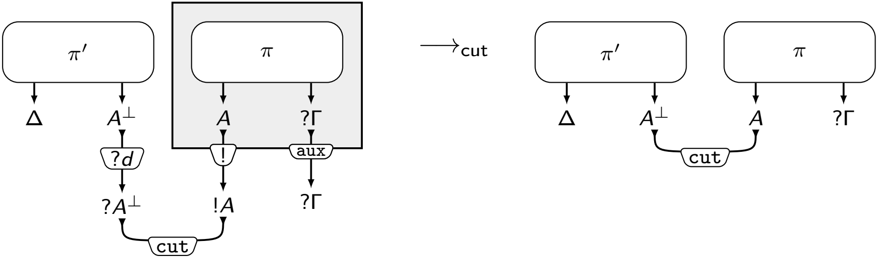


!/? cut-elimination steps in MELL-proof structures (2 of 2)

weakening/promotion: erasure of a resource



dereliction/promotion: access to a resource



Outline

Concrete denotational semantics of LL: the relational model

The simplest denotational semantics of LL is the *relational model*
$\rightsquigarrow$ LL is interpreted in **Rel**, the category of sets and relations

- formula/type $A \rightsquigarrow$ a set $\mathcal{A}$
- proof/program π of $A \multimap B \rightsquigarrow$ a relation $[\![\pi]\!]^{\mathbf{Rel}} \subseteq \mathcal{A} \times \mathcal{B}$
- invariance under cut-elimination/execution: if $\pi \to_{\mathsf{cut}} \pi'$ then $[\![\pi]\!]^{\mathbf{Rel}} = [\![\pi']\!]^{\mathbf{Rel}}$

Remark: A coherence space $\mathcal{A}$ (interpreting the formula A) can be seen as a set of elements (*points*) endowed with a reflexive and symmetric relation (*coherence*). The set $\mathcal{A}$ without the coherence relation is the interpretation of A in **Rel**.

$\rightsquigarrow$ relational model can be seen as coherence semantics "without coherence"
$\rightsquigarrow$ the price to pay is that negation is invisible in **Rel**: $\mathcal{A} = \mathcal{A}^{\perp}$

Apparently, $[\![\pi]\!]^{\mathbf{Rel}}$ is just a set of points without any structure...

Concrete denotational semantics of LL: the relational model

The simplest denotational semantics of LL is the *relational model*
$\rightsquigarrow$ LL is interpreted in **Rel**, the category of sets and relations

- formula/type $A \rightsquigarrow$ a set $\mathcal{A}$
- proof/program π of $A \multimap B \rightsquigarrow$ a relation $[\![\pi]\!]^{\mathbf{Rel}} \subseteq \mathcal{A} \times \mathcal{B}$
- invariance under cut-elimination/execution: if $\pi \to_{\mathsf{cut}} \pi'$ then $[\![\pi]\!]^{\mathbf{Rel}} = [\![\pi']\!]^{\mathbf{Rel}}$

Remark: A coherence space $\mathcal{A}$ (interpreting the formula A) can be seen as a set of elements (*points*) endowed with a reflexive and symmetric relation (*coherence*). The set $\mathcal{A}$ without the coherence relation is the interpretation of A in **Rel**.

$\rightsquigarrow$ relational model can be seen as coherence semantics "without coherence"
$\rightsquigarrow$ the price to pay is that negation is invisible in **Rel**: $\mathcal{A} = \mathcal{A}^{\perp}$

Apparently, $[\![\pi]\!]^{\mathbf{Rel}}$ is just a set of points without any structure...

Concrete denotational semantics of LL: the relational model

The simplest denotational semantics of LL is the *relational model*
$\rightsquigarrow$ LL is interpreted in **Rel**, the category of sets and relations

- formula/type $A \rightsquigarrow$ a set $\mathcal{A}$
- proof/program π of $A \multimap B \rightsquigarrow$ a relation $[\![\pi]\!]^{\textbf{Rel}} \subseteq \mathcal{A} \times \mathcal{B}$
- invariance under cut-elimination/execution: if $\pi \to_{\textsf{cut}} \pi'$ then $[\![\pi]\!]^{\textbf{Rel}} = [\![\pi']\!]^{\textbf{Rel}}$

Remark: A coherence space $\mathcal{A}$ (interpreting the formula A) can be seen as a set of elements (*points*) endowed with a reflexive and symmetric relation (*coherence*). The set $\mathcal{A}$ without the coherence relation is the interpretation of A in **Rel**.

$\rightsquigarrow$ relational model can be seen as coherence semantics "without coherence"
$\rightsquigarrow$ the price to pay is that negation is invisible in **Rel**: $\mathcal{A} = \mathcal{A}^{\perp}$

Apparently, $[\![\pi]\!]^{\textbf{Rel}}$ is just a set of points without any structure . . .

Concrete denotational semantics of LL: the finiteness spaces

Ehrhard (2005) introduced *finiteness spaces*, a new denotational semantics of LL
Fin = the category of finiteness spaces and continuous linear functions $\to_{\mathsf{lin}}$

- formula/type $A \rightsquigarrow$ a finiteness space $\mathcal{A}$ (topological vector space over a field)
- proof/program π of $A \multimap B \rightsquigarrow$ a continuous linear map $[\![\pi]\!]^{\mathbf{Fin}} \colon \mathcal{A} \to_{\mathsf{lin}} \mathcal{B}$
- invariance under cut-elimination/execution: if $\pi \to_{\mathsf{cut}} \pi'$ then $[\![\pi]\!]^{\mathbf{Fin}} = [\![\pi']\!]^{\mathbf{Fin}}$

In **Fin**, ! is an endofunctor such that:

- proof/program π of $!A \multimap B \rightsquigarrow$ a continuous linear map $[\![\pi]\!]^{\mathbf{Fin}} \colon !\mathcal{A} \to_{\mathsf{lin}} \mathcal{B}$
  $=$ a continuous analytical map $[\![\pi]\!]^{\mathbf{Fin}} \colon \mathcal{A} \to_{\mathsf{an}} \mathcal{B}$
- the categorical structure of ! corresponds to the *differential operations* on these continuous analytical maps

$\rightsquigarrow$ Any analytical map is equal to its *Taylor expansion* at any point of its domain

These differential operations and the notion of Taylor expansion can be internalizied in the syntax
$\rightsquigarrow$ *Differential Linear Logic* (DiLL_0), introduced by Ehrhard and Regnier (2006)

Concrete denotational semantics of LL: the finiteness spaces

Ehrhard (2005) introduced *finiteness spaces*, a new denotational semantics of LL
Fin = the category of finiteness spaces and continuous linear functions $\to_{\mathsf{lin}}$

- formula/type $A \rightsquigarrow$ a finiteness space $\mathcal{A}$ (topological vector space over a field)
- proof/program π of $A \multimap B \rightsquigarrow$ a continuous linear map $[\![\pi]\!]^{\mathbf{Fin}} \colon \mathcal{A} \to_{\mathsf{lin}} \mathcal{B}$
- invariance under cut-elimination/execution: if $\pi \to_{\mathsf{cut}} \pi'$ then $[\![\pi]\!]^{\mathbf{Fin}} = [\![\pi']\!]^{\mathbf{Fin}}$

In **Fin**, ! is an endofunctor such that:

- proof/program π of $!A \multimap B \rightsquigarrow$ a continuous linear map $[\![\pi]\!]^{\mathbf{Fin}} \colon !\mathcal{A} \to_{\mathsf{lin}} \mathcal{B}$
  $=$ a continuous analytical map $[\![\pi]\!]^{\mathbf{Fin}} \colon \mathcal{A} \to_{\mathsf{an}} \mathcal{B}$
- the categorical structure of ! corresponds to the *differential operations* on these continuous analytical maps

$\rightsquigarrow$ Any analytical map is equal to its *Taylor expansion* at any point of its domain

These differential operations and the notion of Taylor expansion can be internalizied in the syntax
$\rightsquigarrow$ *Differential Linear Logic* (DiLL_0), introduced by Ehrhard and Regnier (2006)

Concrete denotational semantics of LL: the finiteness spaces

Ehrhard (2005) introduced *finiteness spaces*, a new denotational semantics of LL
Fin = the category of finiteness spaces and continuous linear functions $\to_{\mathsf{lin}}$

- formula/type $A \rightsquigarrow$ a finiteness space $\mathcal{A}$ (topological vector space over a field)
- proof/program π of $A \multimap B \rightsquigarrow$ a continuous linear map $[\![\pi]\!]^{\mathbf{Fin}} \colon \mathcal{A} \to_{\mathsf{lin}} \mathcal{B}$
- invariance under cut-elimination/execution: if $\pi \to_{\mathsf{cut}} \pi'$ then $[\![\pi]\!]^{\mathbf{Fin}} = [\![\pi']\!]^{\mathbf{Fin}}$

In **Fin**, ! is an endofunctor such that:

- proof/program π of $!A \multimap B \rightsquigarrow$ a continuous linear map $[\![\pi]\!]^{\mathbf{Fin}} \colon !\mathcal{A} \to_{\mathsf{lin}} \mathcal{B}$
  = a continuous analytical map $[\![\pi]\!]^{\mathbf{Fin}} \colon \mathcal{A} \to_{\mathsf{an}} \mathcal{B}$
- the categorical structure of ! corresponds to the *differential operations* on these continuous analytical maps

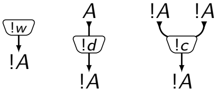
$\rightsquigarrow$ Any analytical map is equal to its *Taylor expansion* at any point of its domain

These differential operations and the notion of Taylor expansion can be internalizied in the syntax
$\rightsquigarrow$ *Differential Linear Logic* (DiLL_0), introduced by Ehrhard and Regnier (2006)

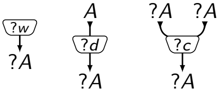
Differential Linear Logic (Ehrhard & Regnier [2006])

DiLL_0 formulas = MELL formulas

DiLL_0 proof-structures = the same as MELL except for the rules introducing !
$\rightsquigarrow$ (infinitary) boxes are replaced by three new kind of (finitary) nodes:



(perfectly symmetric to structural nodes



)

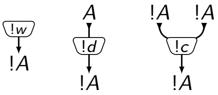
- Co-dereliction (!d) releases inputs of type !A that can be called *exactly once* (i.e. *linearly*) during the cut elimination process
- Thanks to co-contraction (!c) and co-weakening (!w), in DiLL_0 every resource is available only *finitely many times*.

Idea: a proof is an *infinite* (formal) *sum* of graphs, cut elimination is a *local* graph rewriting

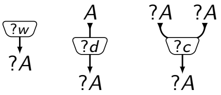
Differential Linear Logic (Ehrhard & Regnier [2006])

DiLL_0 formulas = MELL formulas

DiLL_0 proof-structures = the same as MELL except for the rules introducing !
$\rightsquigarrow$ (infinitary) boxes are replaced by three new kind of (finitary) nodes:



(perfectly symmetric to structural nodes



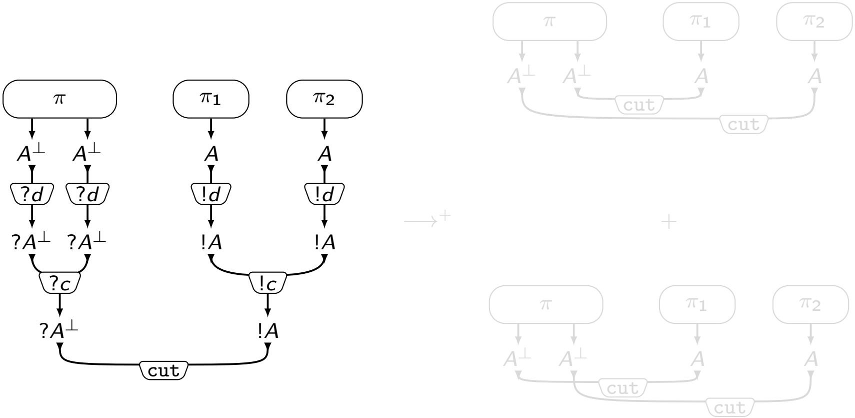
)

- Co-dereliction ($!d$) releases inputs of type $!A$ that can be called *exactly once* (i.e. *linearly*) during the cut elimination process
- Thanks to co-contraction ($!c$) and co-weakening ($!w$), in DiLL_0 every resource is available only *finitely many times*.

Idea: a proof is an *infinite* (formal) *sum* of graphs, cut elimination is a *local* graph rewriting

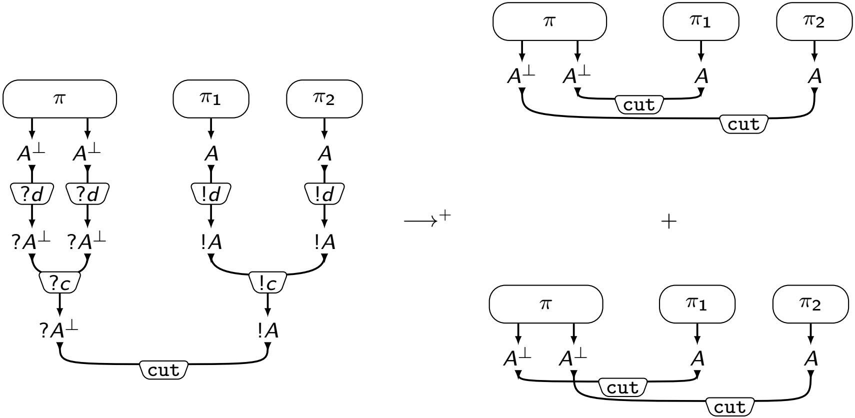
Examples of co-structural cut-elimination steps (1 of 2)

A DiLL_0 proof-structure reduces to a finite set of DiLL_0 proof-structures

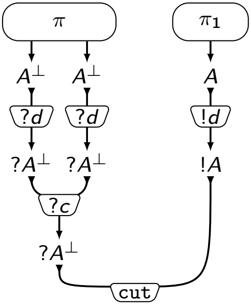


Examples of co-structural cut-elimination steps (1 of 2)

A DiLL_0 proof-structure reduces to a finite set of DiLL_0 proof-structures



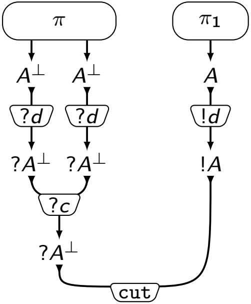
Examples of co-structural cut-elimination steps (2 of 2)



$\longrightarrow^{+} \quad 0 \quad (= \emptyset)$

Why? Because there is a mismatch: π ask for 2 copies of a resource, but only 1 copy (π_1) is available and it cannot be duplicated.

Examples of co-structural cut-elimination steps (2 of 2)



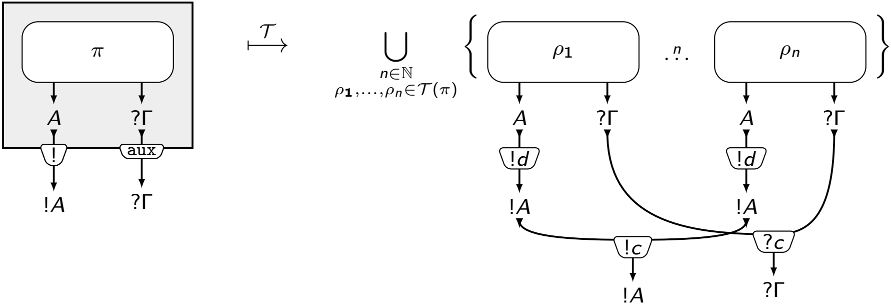
$\longrightarrow^+ \quad 0 \quad (= \emptyset)$

Why? Because there is a mismatch: π ask for 2 copies of a resource, but only 1 copy (π_1) is available and it cannot be duplicated.

Taylor expansion of a MELL proof-structure

$$\text{Taylor expansion } \mathcal{T}: \quad \begin{array}{ccc} \text{MELL} & \to & \mathcal{P}(\text{DiLL}_0) \\ \pi & \mapsto & \mathcal{T}(\pi) \end{array}$$

Idea: each box is replaced by n copies of its content, recursively (for every box and every $n \in \mathbb{N}$)



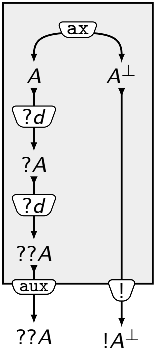
(definition by induction on the depth of π)

Taylor expansion of a MELL proof-structure

$$\text{Taylor expansion } \mathcal{T}: \quad \begin{array}{ccc} \text{MELL} & \to & \mathcal{P}(\text{DiLL}_0) \\ \pi & \mapsto & \mathcal{T}(\pi) \end{array}$$

Idea: each box is replaced by n copies of its content, recursively (for every box and every $n \in \mathbb{N}$)

Example: $\pi =$



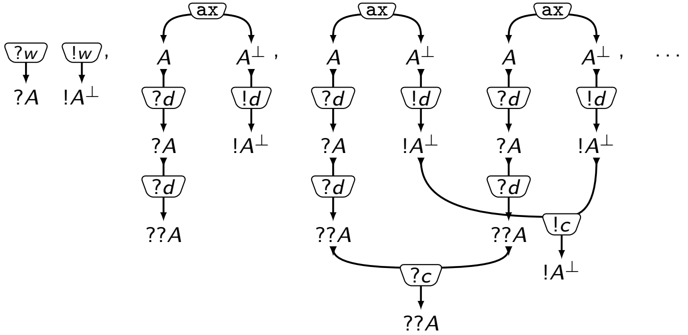
Taylor expansion of a MELL proof-structure

Taylor expansion $\mathcal{T}$: $\begin{array}{ccc} \text{MELL} & \to & \mathcal{P}(\text{DiLL}_0) \\ \pi & \mapsto & \mathcal{T}(\pi) \end{array}$

Idea: each box is replaced by n copies of its content, recursively (for every box and every $n \in \mathbb{N}$)

Example:

$\mathcal{T}(\pi) = \{$



$\}$

Taylor expansion: bridge between syntax and semantics (1 of 2)

Remark: A finiteness space (interpreting the formula A in **Fin**) can be seen as a set $\mathcal{A}$ equipped with a notion of "finitary" subsets of $\mathcal{A}$;
a MELL-proof-structure is interpreted as a finitary set.

Remark: It turns out that $\mathcal{A}$ is the same set as the interpretation of A in **Rel**, and

$$[\![\pi]\!]^{\mathbf{Fin}} = [\![\pi]\!]^{\mathbf{Rel}} (\neq [\![\pi]\!]^{\mathbf{Coh}})$$

Summing up:

- $[\![\pi]\!]^{\mathbf{Fin}} : \mathcal{A} \to_{\mathsf{an}} \mathcal{B}$ is its Taylor expansion
- the Taylor expansion can be internalized in the syntax
- the interpretation of a proof in **Fin** and in **Rel** is the same

$\rightsquigarrow$ It is natural to expect a relationship between the Taylor expansion of a MELL-proof-structure π and the interpretation of π in **Rel**

Taylor expansion: bridge between syntax and semantics (1 of 2)

Remark: A finiteness space (interpreting the formula A in **Fin**) can be seen as a set $\mathcal{A}$ equipped with a notion of "finitary" subsets of $\mathcal{A}$;
a MELL-proof-structure is interpreted as a finitary set.

Remark: It turns out that $\mathcal{A}$ is the same set as the interpretation of A in **Rel**, and

$$[\![\pi]\!]^{\mathbf{Fin}} = [\![\pi]\!]^{\mathbf{Rel}} (\neq [\![\pi]\!]^{\mathbf{Coh}})$$

Summing up:

- $[\![\pi]\!]^{\mathbf{Fin}} \colon \mathcal{A} \to_{\mathsf{an}} \mathcal{B}$ is its Taylor expansion
- the Taylor expansion can be internalized in the syntax
- the interpretation of a proof in **Fin** and in **Rel** is the same

$\rightsquigarrow$ It is natural to expect a relationship between the Taylor expansion of a MELL-proof-structure π and the interpretation of π in **Rel**

Taylor expansion: bridge between syntax and semantics (2 of 2)

Our first "contribution":

Proposition (G., P., TdF., but also "folklore")

For every normal (= *cut-free, with atomic axioms*) MELL-proof-structure π:

$$[\![\pi]\!]^{\mathbf{Rel}}_{\mathrm{inj}/\sim} \simeq \mathcal{T}(\pi)\,.$$

where $[\![\pi]\!]^{\mathbf{Rel}}_{\mathrm{inj}/\sim} \subseteq [\![\pi]\!]^{\mathbf{Rel}}$, more precisely:

- $[\![\pi]\!]^{\mathbf{Rel}}_{\mathrm{inj}}$ = the set containing the "most informative" points of $[\![\pi]\!]^{\mathbf{Rel}}$
- $\sim$ is an equivalence relation on the points based on renaming of atoms

Given a normal MELL proof-structure π, the proposition above allows us to deal with the elements of $\mathcal{T}(\pi)$ instead the points of $[\![\pi]\!]^{\mathbf{Rel}}$

$\rightsquigarrow$ a geometrical representation of the points of the relational semantics of π.

$$[\![\pi]\!] = \begin{cases} \text{a morphism in some category } (\mathbf{Rel}, \mathbf{Fin}, \mathbf{Coh}, \dots) \\ \text{a (infinite) set of points (of a set, finiteness space, coherence space, \dots)} \\ \text{a (infinite) set of graphs (the DiLL}_0\text{-proof-structures of } \mathcal{T}(\pi)) \end{cases}$$

Taylor expansion: bridge between syntax and semantics (2 of 2)

Our first "contribution":

Proposition (G., P., TdF., but also "folklore")

For every normal (= *cut-free, with atomic axioms*) MELL-proof-structure π:

$$[\![\pi]\!]^{\mathbf{Rel}}_{\mathrm{inj}/\sim} \simeq \mathcal{T}(\pi)\,.$$

where $[\![\pi]\!]^{\mathbf{Rel}}_{\mathrm{inj}/\sim} \subseteq [\![\pi]\!]^{\mathbf{Rel}}$, more precisely:

- $[\![\pi]\!]^{\mathbf{Rel}}_{\mathrm{inj}}$ = the set containing the "most informative" points of $[\![\pi]\!]^{\mathbf{Rel}}$
- $\sim$ is an equivalence relation on the points based on renaming of atoms

Given a normal MELL proof-structure π, the proposition above allows us to deal with the elements of $\mathcal{T}(\pi)$ instead the points of $[\![\pi]\!]^{\mathbf{Rel}}$

$\rightsquigarrow$ a geometrical representation of the points of the relational semantics of π.

$$[\![\pi]\!] = \begin{cases} \text{a morphism in some category } (\mathbf{Rel}, \mathbf{Fin}, \mathbf{Coh}, \dots) \\ \text{a (infinite) set of points (of a set, finiteness space, coherence space, \dots)} \\ \text{a (infinite) set of graphs (the DiLL}_0\text{-proof-structures of } \mathcal{T}(\pi)) \end{cases}$$

Taylor expansion: bridge between syntax and semantics (2 of 2)

Our first "contribution":

Proposition (G., P., TdF., but also "folklore")

For every normal (= *cut-free, with atomic axioms*) MELL-proof-structure π:

$$[\![\pi]\!]^{\mathbf{Rel}}_{\mathrm{inj}/\sim} \simeq \mathcal{T}(\pi).$$

where $[\![\pi]\!]^{\mathbf{Rel}}_{\mathrm{inj}/\sim} \subseteq [\![\pi]\!]^{\mathbf{Rel}}$, more precisely:

- $[\![\pi]\!]^{\mathbf{Rel}}_{\mathrm{inj}}$ = the set containing the "most informative" points of $[\![\pi]\!]^{\mathbf{Rel}}$
- $\sim$ is an equivalence relation on the points based on renaming of atoms

Given a normal MELL proof-structure π, the proposition above allows us to deal with the elements of $\mathcal{T}(\pi)$ instead the points of $[\![\pi]\!]^{\mathbf{Rel}}$

$\rightsquigarrow$ a geometrical representation of the points of the relational semantics of π.

$$[\![\pi]\!] = \begin{cases} \text{a morphism in some category } (\mathbf{Rel}, \mathbf{Fin}, \mathbf{Coh}, \ldots) \\ \text{a (infinite) set of points (of a set, finiteness space, coherence space, ...)} \\ \text{a (infinite) set of graphs (the DiLL}_0\text{-proof-structures of } \mathcal{T}(\pi)) \end{cases}$$

Outline

The question of injectivity and its motivations

The question of injectivity

Given two normal terms π and π' in a given syntax (with rewrite rules) and their interpretations $[\![\pi]\!]$ and $[\![\pi']\!]$ in some denotational semantics:

$$[\![\pi]\!] = [\![\pi']\!] \quad \overset{?}{\Rightarrow} \quad \pi = \pi' .$$

If the implication holds, then that denotational semantics is injective.

- In categorical terms, injectivity corresponds faithfulness of the interpretation.
- Injectivity is a natural and well studied question for denotational semantics of λ-calculi and term rewriting systems (Friedman '75, Statman '82).

In '90s Tortora de Falco addressed the question of injectivity for the following case:

- syntax $\rightsquigarrow$ Linear Logic (LL) proof-structures
- normal $\rightsquigarrow$ cut-free and η-expanded (= with atomic axioms) proof-structures
- semantics $\rightsquigarrow$ set-based model (coherence spaces, relational semantics, . . .)

Among the motivations: To prove the uniqueness of the normal form (Danos):

where π_1, π_2 are normal $\overset{\text{invar.}}{\Longrightarrow}$ $[\![\pi_1]\!] = [\![\pi]\!] = [\![\pi_2]\!]$ $\overset{\text{inj.}}{\Longrightarrow}$ $\pi_1 = \pi_2$

The question of injectivity and its motivations

The question of injectivity

Given two normal terms π and π' in a given syntax (with rewrite rules) and their interpretations $[\![\pi]\!]$ and $[\![\pi']\!]$ in some denotational semantics:

$$[\![\pi]\!] = [\![\pi']\!] \quad \overset{?}{\Rightarrow} \quad \pi = \pi' .$$

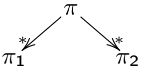
If the implication holds, then that denotational semantics is injective.

- In categorical terms, injectivity corresponds faithfulness of the interpretation.
- Injectivity is a natural and well studied question for denotational semantics of λ-calculi and term rewriting systems (Friedman '75, Statman '82).

In '90s Tortora de Falco addressed the question of injectivity for the following case:

- syntax $\rightsquigarrow$ Linear Logic (LL) proof-structures
- normal $\rightsquigarrow$ cut-free and η-expanded (= with atomic axioms) proof-structures
- semantics $\rightsquigarrow$ set-based model (coherence spaces, relational semantics, . . .)

Among the motivations: To prove the uniqueness of the normal form (Danos):



where π_1, π_2 are normal $\overset{\text{invar.}}{\Longrightarrow}$ $[\![\pi_1]\!] = [\![\pi]\!] = [\![\pi_2]\!]$ $\overset{\text{inj.}}{\Longrightarrow}$ $\pi_1 = \pi_2$

Some results about injectivity (w.r.t. MELL proof-structures)

- The question of injectivity has been deeply studied for the first time in Tortora de Falco's thesis (2000).
- Other contributions: TdF., Laurent, Boudes, Pagani, de Carvalho, . . .

Theorem: about injectivity of coherence semantics (TdF. [2003])

- Coherence semantics **is not** injective: there exist two normal MELL proof-structures π_1 and π_2 such that $[\![\pi_1]\!]^{\mathbf{Coh}} = [\![\pi_2]\!]^{\mathbf{Coh}}$ and $\pi_1 \neq \pi_2$.
- Coherence semantics is injective w.r.t. some fragments of MELL

Theorem: injectivity of relational semantics (de Carvalho, TdF. [2012])

- Given two normal MELL proof-structures π_1 and π_2 *without* $?w$ *nodes*, if $[\![\pi_1]\!]^{\mathbf{Rel}} = [\![\pi_2]\!]^{\mathbf{Rel}}$ then $\pi_1 = \pi_2$.
- Conjecture (TdF. [2003]): relational semantics is injective w.r.t. all MELL proof-structures (proof by de Carvalho).

Some results about injectivity (w.r.t. MELL proof-structures)

- The question of injectivity has been deeply studied for the first time in Tortora de Falco's thesis (2000).
- Other contributions: TdF., Laurent, Boudes, Pagani, de Carvalho, ...

Theorem: about injectivity of coherence semantics (TdF. [2003])

- Coherence semantics **is not** injective: there exist two normal MELL proof-structures π_1 and π_2 such that $[\![\pi_1]\!]^{\textbf{Coh}} = [\![\pi_2]\!]^{\textbf{Coh}}$ and $\pi_1 \neq \pi_2$.
- Coherence semantics is injective w.r.t. some fragments of MELL

Theorem: injectivity of relational semantics (de Carvalho, TdF. [2012])

- Given two normal MELL proof-structures π_1 and π_2 *without ?w nodes*, if $[\![\pi_1]\!]^{\textbf{Rel}} = [\![\pi_2]\!]^{\textbf{Rel}}$ then $\pi_1 = \pi_2$.
- Conjecture (TdF. [2003]): relational semantics is injective w.r.t. all MELL proof-structures (proof by de Carvalho).

How to prove injectivity of a set-based model?

1. Hypothesis: Let π_1 and π_2 be two normal MELL proof-structures such that $[\![\pi_1]\!] = [\![\pi_2]\!]$ (where $[\![\cdot]\!]$ is a *set-based* semantics: $[\![\cdot]\!]^{\mathbf{Rel}}, [\![\cdot]\!]^{\mathbf{Coh}}, [\![\cdot]\!]^{\mathbf{Fin}}, \dots$).

   Set-based model $\leftrightsquigarrow$ For any MELL proof-structure π, $[\![\pi]\!]$ is a *set*.

2. Key-Lemma:

   For any normal MELL proof-structures π, there is at least *one* "discriminating element" $\rho \in [\![\pi]\!]$, i.e., for any normal MELL-proof-structure π'

   $$\text{if } \rho \in [\![\pi]\!] \cap [\![\pi']\!], \text{ then } \pi = \pi'.$$

3. Conclusion (injectivity): Since $[\![\pi_1]\!] = [\![\pi_2]\!]$, then the "discriminating element" ρ of $\{\pi_1, \pi_2\}$ satisfies $\rho \in [\![\pi_1]\!] \cap [\![\pi_2]\!]$, hence $\pi_1 = \pi_2$ by the Key-Lemma.

The crucial points in this kind of proofs are:

- To define the "discriminating element" ρ of any pair $\{\pi, \pi'\}$ of normal MELL proof-structures.

  Remark: the "structure" of ρ *depends* on π, π' (the Key-Lemma claims: "$\forall \{\pi, \pi'\} \, \exists \rho$ such that if $P(\rho, \pi, \pi')$ then ρ is discriminating for π, π'")
- To build univocally $\pi = \pi'$ from ρ satisfying $P(\rho, \pi, \pi')$, i.e. to prove the "discriminating power" of such a ρ.

How to prove injectivity of a set-based model?

1. Hypothesis: Let π_1 and π_2 be two normal MELL proof-structures such that $[\![\pi_1]\!] = [\![\pi_2]\!]$ (where $[\![\cdot]\!]$ is a *set-based* semantics: $[\![\cdot]\!]^{\mathbf{Rel}}, [\![\cdot]\!]^{\mathbf{Coh}}, [\![\cdot]\!]^{\mathbf{Fin}}, \dots$).

   Set-based model $\leftrightsquigarrow$ For any MELL proof-structure π, $[\![\pi]\!]$ is a *set*.

2. Key-Lemma:

   For any normal MELL proof-structures π, there is at least *one* "discriminating element" $\rho \in [\![\pi]\!]$, i.e., for any normal MELL-proof-structure π'

   $$\text{if } \rho \in [\![\pi]\!] \cap [\![\pi']\!], \text{ then } \pi = \pi'.$$

3. Conclusion (injectivity): Since $[\![\pi_1]\!] = [\![\pi_2]\!]$, then the "discriminating element" ρ of $\{\pi_1, \pi_2\}$ satisfies $\rho \in [\![\pi_1]\!] \cap [\![\pi_2]\!]$, hence $\pi_1 = \pi_2$ by the Key-Lemma.

The crucial points in this kind of proofs are:

- To define the "discriminating element" ρ of any pair $\{\pi, \pi'\}$ of normal MELL proof-structures.

  Remark: the "structure" of ρ *depends* on π, π' (the Key-Lemma claims: "$\forall \{\pi, \pi'\} \, \exists \rho$ such that if $P(\rho, \pi, \pi')$ then ρ is discriminating for π, π'")
- To build univocally $\pi = \pi'$ from ρ satisfying $P(\rho, \pi, \pi')$, i.e. to prove the "discriminating power" of such a ρ.

How to prove injectivity of a set-based model?

1. Hypothesis: Let π_1 and π_2 be two normal MELL proof-structures such that $[\![\pi_1]\!] = [\![\pi_2]\!]$ (where $[\![\cdot]\!]$ is a *set-based* semantics: $[\![\cdot]\!]^{\mathbf{Rel}}, [\![\cdot]\!]^{\mathbf{Coh}}, [\![\cdot]\!]^{\mathbf{Fin}}, \dots$).

   Set-based model $\leftrightsquigarrow$ For any MELL proof-structure π, $[\![\pi]\!]$ is a *set*.
2. Key-Lemma:

   For any normal MELL proof-structures π, there is at least *one* "discriminating element" $\rho \in [\![\pi]\!]$, i.e., for any normal MELL-proof-structure π'

   $$\text{if } \rho \in [\![\pi]\!] \cap [\![\pi']\!], \text{ then } \pi = \pi'.$$
3. Conclusion (injectivity): Since $[\![\pi_1]\!] = [\![\pi_2]\!]$, then the "discriminating element" ρ of $\{\pi_1, \pi_2\}$ satisfies $\rho \in [\![\pi_1]\!] \cap [\![\pi_2]\!]$, hence $\pi_1 = \pi_2$ by the Key-Lemma.

The crucial points in this kind of proofs are:

- To define the "discriminating element" ρ of any pair $\{\pi, \pi'\}$ of normal MELL proof-structures.

  Remark: the "structure" of ρ *depends* on π, π' (the Key-Lemma claims: "$\forall \{\pi, \pi'\} \, \exists \rho$ such that if $P(\rho, \pi, \pi')$ then ρ is discriminating for π, π'")
- To build univocally $\pi = \pi'$ from ρ satisfying $P(\rho, \pi, \pi')$, i.e. to prove the "discriminating power" of such a ρ.

How to prove injectivity of a set-based model?

1. Hypothesis: Let π_1 and π_2 be two normal MELL proof-structures such that $[\![\pi_1]\!] = [\![\pi_2]\!]$ (where $[\![\cdot]\!]$ is a *set-based* semantics: $[\![\cdot]\!]^{\mathbf{Rel}}, [\![\cdot]\!]^{\mathbf{Coh}}, [\![\cdot]\!]^{\mathbf{Fin}}, \dots$).

   Set-based model $\leftrightsquigarrow$ For any MELL proof-structure π, $[\![\pi]\!]$ is a *set*.
2. Key-Lemma:

   For any normal MELL proof-structures π, there is at least *one* "discriminating element" $\rho \in [\![\pi]\!]$, i.e., for any normal MELL-proof-structure π'

   $$\text{if } \rho \in [\![\pi]\!] \cap [\![\pi']\!], \text{ then } \pi = \pi'.$$
3. Conclusion (injectivity): Since $[\![\pi_1]\!] = [\![\pi_2]\!]$, then the "discriminating element" ρ of $\{\pi_1, \pi_2\}$ satisfies $\rho \in [\![\pi_1]\!] \cap [\![\pi_2]\!]$, hence $\pi_1 = \pi_2$ by the Key-Lemma.

The crucial points in this kind of proofs are:

- To define the "discriminating element" ρ of any pair $\{\pi, \pi'\}$ of normal MELL proof-structures.

  Remark: the "structure" of ρ *depends* on π, π' (the Key-Lemma claims: "$\forall \{\pi, \pi'\} \, \exists \rho$ such that if $P(\rho, \pi, \pi')$ then ρ is discriminating for π, π'")
- To build univocally $\pi = \pi'$ from ρ satisfying $P(\rho, \pi, \pi')$, i.e. to prove the "discriminating power" of such a ρ.

How to prove injectivity of a set-based model?

1. Hypothesis: Let π_1 and π_2 be two normal MELL proof-structures such that $[\![\pi_1]\!] = [\![\pi_2]\!]$ (where $[\![\cdot]\!]$ is a *set-based* semantics: $[\![\cdot]\!]^{\mathbf{Rel}}, [\![\cdot]\!]^{\mathbf{Coh}}, [\![\cdot]\!]^{\mathbf{Fin}}, \dots$).

   Set-based model $\leftrightsquigarrow$ For any MELL proof-structure π, $[\![\pi]\!]$ is a *set*.
2. Key-Lemma:

   For any normal MELL proof-structures π, there is at least *one* "discriminating element" $\rho \in [\![\pi]\!]$, i.e., for any normal MELL-proof-structure π'

   $$\text{if } \rho \in [\![\pi]\!] \cap [\![\pi']\!], \text{ then } \pi = \pi'.$$
3. Conclusion (injectivity): Since $[\![\pi_1]\!] = [\![\pi_2]\!]$, then the "discriminating element" ρ of $\{\pi_1, \pi_2\}$ satisfies $\rho \in [\![\pi_1]\!] \cap [\![\pi_2]\!]$, hence $\pi_1 = \pi_2$ by the Key-Lemma.

The crucial points in this kind of proofs are:

- To define the "discriminating element" ρ of any pair $\{\pi, \pi'\}$ of normal MELL proof-structures.

  Remark: the "structure" of ρ *depends* on π, π' (the Key-Lemma claims: "$\forall \{\pi, \pi'\} \; \exists \rho$ such that if $P(\rho, \pi, \pi')$ then ρ is discriminating for π, π'")
- To build univocally $\pi = \pi'$ from ρ satisfying $P(\rho, \pi, \pi')$, i.e. to prove the "discriminating power" of such a ρ.

Our contribution

Inspired by the proposition relating the relational interpretation and the Taylor expansion in the cut-free case ($[\![\pi]\!]^{\mathbf{Rel}}_{\mathrm{inj}/\sim} \simeq \mathcal{T}(\pi)$), we used DiLL$_0$-proof-structures to study the question of injectivity of **Rel** wrt MELL-proof-structures
$\rightsquigarrow$ we thus obtained a new proof of injectivity of relational semantics, generalizing and simplifying the one of de Carvalho, TdF. [2012].

Theorem: injectivity of relational model (G., P., Tdf.)

1. Given two MELL proof-structures π_1 and π_2 *which are box-connected*, if $\mathcal{T}(\pi_1) = \mathcal{T}(\pi_2)$ then $\pi_1 = \pi_2$.
2. In particular, given two *normal* MELL proof-structures π_1 and π_2 *which are box-connected*, if $[\![\pi_1]\!]^{\mathbf{Rel}} = [\![\pi_2]\!]^{\mathbf{Rel}}$ then $\pi_1 = \pi_2$.

Our novelties:

- it is a simplification, since the "structure" of the discriminating element ρ does not depend on π, π'. Thus the Key-Lemma has a logically less complex claim: "$\forall \rho \in$ DiLL$_0$ such that $P(\rho)$, if $\rho \in \mathcal{T}(\pi) \cap \mathcal{T}(\pi')$, then $\pi = \pi'$.
- generalization because the result holds in presence of cuts: the discriminating element $\rho \in \mathcal{T}(\pi)$ allows to build univocally π also in presence of cuts.

Our contribution

Inspired by the proposition relating the relational interpretation and the Taylor expansion in the cut-free case ($[\![\pi]\!]^{\mathbf{Rel}}_{\mathrm{inj}/\sim} \simeq \mathcal{T}(\pi)$), we used DiLL_0-proof-structures to study the question of injectivity of **Rel** wrt MELL-proof-structures
$\rightsquigarrow$ we thus obtained a new proof of injectivity of relational semantics, generalizing and simplifying the one of de Carvalho, TdF. [2012].

Theorem: injectivity of relational model (G., P., Tdf.)

1. Given two MELL proof-structures π_1 and π_2 *which are box-connected*, if $\mathcal{T}(\pi_1) = \mathcal{T}(\pi_2)$ then $\pi_1 = \pi_2$.
2. In particular, given two *normal* MELL proof-structures π_1 and π_2 *which are box-connected*, if $[\![\pi_1]\!]^{\mathbf{Rel}} = [\![\pi_2]\!]^{\mathbf{Rel}}$ then $\pi_1 = \pi_2$.

Our novelties:

- it is a simplification, since the "structure" of the discriminating element ρ does not depend on π, π'. Thus the Key-Lemma has a logically less complex claim: "$\forall \rho \in \mathrm{DiLL}_0$ such that $P(\rho)$, if $\rho \in \mathcal{T}(\pi) \cap \mathcal{T}(\pi')$, then $\pi = \pi'$.
- generalization because the result holds in presence of cuts: the discriminating element $\rho \in \mathcal{T}(\pi)$ allows to build univocally π also in presence of cuts.

Our contribution

Inspired by the proposition relating the relational interpretation and the Taylor expansion in the cut-free case ($[\![\pi]\!]^{\mathbf{Rel}}_{\mathrm{inj}/\sim} \simeq \mathcal{T}(\pi)$), we used DiLL_0-proof-structures to study the question of injectivity of **Rel** wrt MELL-proof-structures
$\rightsquigarrow$ we thus obtained a new proof of injectivity of relational semantics, generalizing and simplifying the one of de Carvalho, TdF. [2012].

Theorem: injectivity of relational model (G., P., Tdf.)

1. Given two MELL proof-structures π_1 and π_2 *which are box-connected*, if $\mathcal{T}(\pi_1) = \mathcal{T}(\pi_2)$ then $\pi_1 = \pi_2$.
2. In particular, given two *normal* MELL proof-structures π_1 and π_2 *which are box-connected*, if $[\![\pi_1]\!]^{\mathbf{Rel}} = [\![\pi_2]\!]^{\mathbf{Rel}}$ then $\pi_1 = \pi_2$.

Our novelties:

- it is a simplification, since the "structure" of the discriminating element ρ does not depend on π, π'. Thus the Key-Lemma has a logically less complex claim: "$\forall \rho \in \mathsf{DiLL}_0$ such that $P(\rho)$, if $\rho \in \mathcal{T}(\pi) \cap \mathcal{T}(\pi')$, then $\pi = \pi'$.
- generalization because the result holds in presence of cuts: the discriminating element $\rho \in \mathcal{T}(\pi)$ allows to build univocally π also in presence of cuts.

Our contribution

Inspired by the proposition relating the relational interpretation and the Taylor expansion in the cut-free case ($[\![\pi]\!]^{\mathbf{Rel}}_{\mathrm{inj}/\sim} \simeq \mathcal{T}(\pi)$), we used DiLL_0-proof-structures to study the question of injectivity of **Rel** wrt MELL-proof-structures
$\rightsquigarrow$ we thus obtained a new proof of injectivity of relational semantics, generalizing and simplifying the one of de Carvalho, TdF. [2012].

Theorem: injectivity of relational model (G., P., Tdf.)

1. Given two MELL proof-structures π_1 and π_2 *which are box-connected*, if $\mathcal{T}(\pi_1) = \mathcal{T}(\pi_2)$ then $\pi_1 = \pi_2$.
2. In particular, given two *normal* MELL proof-structures π_1 and π_2 *which are box-connected*, if $[\![\pi_1]\!]^{\mathbf{Rel}} = [\![\pi_2]\!]^{\mathbf{Rel}}$ then $\pi_1 = \pi_2$.

Our novelties:

- it is a simplification, since the "structure" of the discriminating element ρ does not depend on π, π'. Thus the Key-Lemma has a logically less complex claim: "$\forall \rho \in \mathsf{DiLL}_0$ such that $P(\rho)$, if $\rho \in \mathcal{T}(\pi) \cap \mathcal{T}(\pi')$, then $\pi = \pi'$.
- generalization because the result holds in presence of cuts: the discriminating element $\rho \in \mathcal{T}(\pi)$ allows to build univocally π also in presence of cuts.