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

Giulio Guerrieri\textsuperscript{1}  
Luc Pellissier\textsuperscript{2}  
Lorenzo Tortora de Falco\textsuperscript{3}

\textsuperscript{1}PPS, Université Paris Diderot
\textsuperscript{2}LIPN, Université Paris 13
\textsuperscript{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  

- **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 \( \rightarrow_{\text{cut}} \). This led to:
  - Curry-Howard correspondence (~1960): e.g. propositional intuitionistic minimal logic/simply typed \( \lambda \)-calculus, 2\textsuperscript{nd} order intuitionistic logic/system \( \mathcal{F} \), propositional classical logic/\( \lambda \mu \)-calculus, etc.
    * formula \( A \rightsquigarrow \) type \( A \)
    * proof \( \pi \) of \( A \Rightarrow B \rightsquigarrow \) program \( \pi \) 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 \( A \) is some category \( C \)
* proof/program \( \pi \) of \( A \Rightarrow B \rightsquigarrow \) a morphism \( \llbracket \pi \rrbracket : A \rightarrow B \) in \( C \)
* invariance under cut-elimination/execution: if \( \pi \rightarrow_{\text{cut}} \pi' \) then \( \llbracket \pi \rrbracket = \llbracket \pi' \rrbracket \)
Proofs representations: from numbers to graphs  

- **Hilbert/Gödel** (∼1920-30): a proof is a finite sequence of formulas (through Gödelization a finite sequence of integers and thus an integer)  
  \[\sim\] 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 (∼1960): e.g. propositional intuitionistic minimal logic/simply typed \(\lambda\)-calculus, 2\(^{\text{nd}}\) order intuitionistic logic/system \(\mathcal{F}\), propositional classical logic/\(\lambda\mu\)-calculus, etc.
    * formula \(A\) \(\sim\) type \(A\)
    * proof \(\pi\) of \(A \Rightarrow B\) \(\sim\) program \(\pi\) with input of type \(A\) and output of type \(B\)
    * cut-elimination \(\sim\) 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 \(A\) \(\sim\) an object \(A\) is some category \(C\)
* proof/program \(\pi\) of \(A \Rightarrow B\) \(\sim\) a morphism \([\pi]\) : \(A \rightarrow B\) in \(C\)
* 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)
  ⇝ no structure, no dynamics

- **Gentzen** (1936): a proof (in sequent calculus, natural deduction) is a finite tree with an internal *dynamics*, cut-elimination → cut. This led to:
  - Curry-Howard correspondence (~1960): e.g. propositional intuitionistic minimal logic/simply typed λ-calculus, 2nd order intuitionistic logic/system $\mathcal{F}$, propositional classical logic/λμ-calculus, etc.
    - formula $A$ ⇝ type $A$
    - proof $\pi$ of $A \Rightarrow B$ ⇝ program $\pi$ with input of type $A$ and output of type $B$
    - cut-elimination ⇝ 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 $A$ ⇝ an object $A$ is some category $C$
- proof/program $\pi$ of $A \Rightarrow B$ ⇝ a morphism $\llbracket \pi \rrbracket : A \to B$ in $C$
- invariance under cut-elimination/execution: if $\pi \to \text{cut} \ \pi'$ then $\llbracket \pi \rrbracket = \llbracket \pi' \rrbracket$
Proofs representations: from numbers to graphs

- **Hilbert/Gödel** (approximately 1920-30): a proof is a finite sequence of formulas (through Gödelization a finite sequence of integers and thus an integer).
  \[ \Rightarrow \text{no structure, no dynamics} \]

- **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** (approximately 1960): e.g. propositional intuitionistic minimal logic/simply typed \( \lambda \)-calculus, 2\(^{nd}\) order intuitionistic logic/system \( \mathcal{F} \), propositional classical logic/\( \lambda \mu \)-calculus, etc.
    - formula \( A \Rightarrow A \)
    - proof \( \pi \) of \( A \Rightarrow B \Rightarrow B \)
    - cut-elimination \( \Rightarrow \) program \( \pi \) with input of type \( A \) and output of type \( B \)
  - denotational semantics (approximately 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 \( A \Rightarrow A \Rightarrow C \)
- proof/program \( \pi \) of \( A \Rightarrow B \Rightarrow B \) a morphism \( \llbracket \pi \rrbracket : A \rightarrow B \) in \( C \)
- invariance under cut-elimination/execution: if \( \pi \rightarrow_{\text{cut}} \pi' \) then \( \llbracket \pi \rrbracket = \llbracket \pi' \rrbracket \)
Girard (1987): a proof is a graph, cut-elimination is graph rewriting

- denotational semantics of intuitionistic logic/λ-calculus: let \( \text{Coh} \) (resp. \( \text{Coh}_l \)) be the category of coherence spaces and stable \( \rightarrow_{st} \) (resp. linear \( \rightarrow_{o} \)) functions.
  - formula/type \( A \rightsquigarrow \) a coherence space \( \mathcal{A} \)
  - proof/program \( \pi \) of \( A \Rightarrow B \rightsquigarrow \) a stable function \( \llbracket \pi \rrbracket_{\text{Coh}} : \mathcal{A} \rightarrow_{st} \mathcal{B} \)
  - invariance under cut-elimination/execution: if \( \pi \rightarrow_{\text{cut}} \pi' \) then \( \llbracket \pi \rrbracket_{\text{Coh}} = \llbracket \pi' \rrbracket_{\text{Coh}} \)

\[
\begin{array}{c}
\text{Coh} \\
\downarrow
\end{array} \quad \begin{array}{c}
\downarrow \\
\text{Coh}_l
\end{array}
\]

\( \rightsquigarrow \text{Coh}[A, B] = \text{Coh}_l[!A, B] \), more precisely \( f : A \rightarrow_{st} B = f : !A \rightarrow_{o} B \)

- logical formulas: the operations \( ! \) and \( \rightarrow_{o} \) 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

\[
A \Rightarrow B = !A \rightarrow_{o} B
\]
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 \( \text{Coh} \) (resp. \( \text{Coh}_l \)) be the category of coherence spaces and stable \( \rightarrow_{st} \) (resp. linear \( \rightarrow \)) functions.
    - formula/type \( A \rightarrow \) a coherence space \( \mathcal{A} \)
    - proof/program \( \pi \) of \( A \Rightarrow B \rightarrow \) a stable function \( \llbracket \pi \rrbracket^{\text{Coh}}: \mathcal{A} \rightarrow_{st} \mathcal{B} \)
    - invariance under cut-elimination/execution: if \( \pi \rightarrow_{\text{cut}} \pi' \) then \( \llbracket \pi \rrbracket^{\text{Coh}} = \llbracket \pi' \rrbracket^{\text{Coh}} \)

\[
\begin{array}{ccc}
\text{Coh} & \downarrow & \text{Coh}_l \\
\rotatebox{90}{\( \supseteq \)} & \text{\rotatebox{90}{\( \supseteq \)}}
\end{array}
\]

\( \llbracket A, B \rrbracket = \text{Coh}_l[!A, B] \), more precisely \( f : A \rightarrow_{st} B = f : !A \rightarrow B \)

- logical formulas: the operations \( ! \) and \( \rightarrow \) can be internalized in the syntax
  \( \llbracket \) Linear Logic (LL) \( \rrbracket \) 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

\[ A \Rightarrow B = !A \rightarrow B \]
Girard (1987): a proof is a graph, cut-elimination is graph rewriting
(continued)

- semantics of proofs in LL:
  - proof/program \( \pi \) of \( !A \rightarrow B \) \( \rightsquigarrow \) a stable function \( \llbracket \pi \rrbracket^{\text{Coh}}_{\text{st}} : A \rightarrow B \)
    - Idea: the proof \( \pi \) may use the hypothesis \( A \) an arbitrary number of times
      i.e. the program \( \pi \) may call its arguments of type \( A \) at will
  - proof/program \( \pi \) of \( A \rightarrow B \) \( \rightsquigarrow \) a linear function \( \llbracket \pi \rrbracket^{\text{Coh}!} : !A \rightarrow B \)
    - Idea: the proof \( \pi \) uses the hypothesis \( A \) linearly (exactly once)
      i.e. the program \( \pi \) 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 $\pi$ of $!A \rightarrow B \rightsquigarrow$ a stable function $[\pi]^{Coh} : A \rightarrow_{st} B$
    - Idea: the proof $\pi$ may use the hypothesis $A$ an arbitrary number of times, i.e. the program $\pi$ may call its arguments of type $A$ at will
  - proof/program $\pi$ of $A \rightarrow B \rightsquigarrow$ a linear function $[\pi]^{Coh/} : !A \rightarrow B$
    - Idea: the proof $\pi$ uses the hypothesis $A$ linearly (exactly once), i.e. the program $\pi$ 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 $\pi$ of $!A \rightarrow B \rightsquigarrow$ a stable function $\llbracket \pi \rrbracket^{\text{Coh}}_{\text{st}}: A \rightarrow \text{st } B$
    - Idea: the proof $\pi$ may use the hypothesis $A$ an arbitrary number of times
    i.e. the program $\pi$ may call its arguments of type $A$ at will
  - Proof/program $\pi$ of $A \rightarrow B \rightsquigarrow$ a linear function $\llbracket \pi \rrbracket^{\text{Coh}}_{\text{l}}: !A \rightarrow B$
    - Idea: the proof $\pi$ uses the hypothesis $A$ linearly (exactly once)
    i.e. the program $\pi$ 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 $A ⇝ _\bowtie A$

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

\[
\begin{array}{c}
\wedge \\
\otimes \\
\forall \\
\forall \\
T \\
1 \\
\bot \\
\top \\
\ominus \\
\oplus \\
&
\end{array}
\]

\begin{array}{cc}
\text{multiplicative} & \text{additive} \\
\text{monoidal structure} & \text{cartesian structure}
\end{array}

\text{MELL} = \text{multiplicative and exponential } (\otimes, \&) \text{ fragment of LL (no additives)}

\[
A, B ::= X \mid X^{\bot} \mid 1 \mid \bot \mid A \otimes B \mid A \forall B \mid !A \mid ?A
\]

Negation is involutive. $A^{\bot}$ 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 $\vdash A$

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

\[ \begin{array}{cccc}
\& \quad \land \quad \lor \\
\otimes \quad \lor \quad \& \\
\exists \quad \top \quad \forall \\
\perp \quad \bot \quad 1 \\
0
\end{array} \]

- multiplicative monoidal structure
- additive cartesian structure

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

\[ A, B ::= X \mid X^\perp \mid 1 \mid \perp \mid A \otimes B \mid A \otimes B \mid !A \mid ?A \]

Negation is involutive. $A^\perp$ is defined by induction using the usual De Morgan identities
Example: the following proof of MELL sequent calculus...

\[
\begin{align*}
\vdash X, X_\bot & \quad \text{ax} \\
\vdash X, X_\bot & \quad \text{ax} \\
\vdash X \otimes X, X_\bot, X_\bot & \quad \otimes \\
\vdash Y, Y_\bot & \quad \text{ax} \\
\vdash (X \otimes X) \otimes Y, X_\bot, X_\bot, Y_\bot & \quad \otimes \\
\vdash (X \otimes X) \otimes Y, ?X_\bot, X_\bot, Y_\bot & \quad ?d \\
\vdash (X \otimes X) \otimes Y, ?X_\bot, ?X_\bot, Y_\bot & \quad ?d \\
\vdash (X \otimes X) \otimes Y, ?X_\bot, Y_\bot & \quad ?c \\
\vdash (X \otimes X) \otimes Y, ?X_\bot, Y_\bot & \quad \vdash X, ?X_\bot & \quad ?d \\
\vdash !X, ?X_\bot & \quad ! \\
\vdash (X \otimes X) \otimes Y, ?X_\bot, Y_\bot & \quad \text{cut} \\
\end{align*}
\]
Example: ...corresponds to the following MELL proof-net
cut-elimination steps in MELL-proof structures

contraction/promotion: duplication of a resource

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]

\[
\begin{array}{c}
\pi' \\
\vdash A \perp \\
\vdash c \\
\vdash A \perp \\
\pi \\
\vdash A \\
\vdash \Gamma \\
\vdash \text{cut}
\end{array}
\]
cut-elimination steps in MELL-proof structures

contraction/promotion: duplication of a resource
weakening/promotion: erasure of a resource

dereliction/promotion: access to a resource
weakening/promotion: erasure of a resource

dereliction/promotion: access to a resource
weakening/promotion: erasure of a resource

dereliction/promotion: access to a resource
 weakening/promotion: erasure of a resource

 dereliction/promotion: access to a resource
Outline

1. Introduction: linear logic in a nutshell

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

3. The question of injectivity
Concrete denotational semantics of LL: the relational model

The simplest denotational semantics of LL is the *relational model*

\[ \ll \rightarrow \text{LL is interpreted in } \text{Rel}, \text{ the category of sets and relations} \]

- formula/type \( A \rightarrow a \text{ set } A \)
- proof/program \( \pi \text{ of } A \rightarrow B \rightarrow \text{a relation } \llbracket \pi \rrbracket_{\text{Rel}} \subseteq A \times B \)
- invariance under cut-elimination/execution: if \( \pi \rightarrow_{\text{cut}} \pi' \) then \( \llbracket \pi \rrbracket_{\text{Rel}} = \llbracket \pi' \rrbracket_{\text{Rel}} \)

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 \( \text{Rel} \).

\[ \llbracket A \rrbracket_{\text{Rel}} \rightarrow \text{relational model can be seen as coherence semantics “without coherence”} \]

\[ \llbracket A \rrbracket_{\text{Rel}} \rightarrow \text{the price to pay is that negation is invisible in } \text{Rel}: A = A_{\perp} \]

Apparently, \( \llbracket \pi \rrbracket_{\text{Rel}} \) 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* ⇝ LL is interpreted in **Rel**, the category of sets and relations

- **formula/type** \( A \) ⇝ a set \( A \)
- **proof/program** \( \pi \) of \( A \rightarrow B \) ⇝ a relation \( \llbracket \pi \rrbracket_{\text{Rel}} \subseteq A \times B \)
- invariance under cut-elimination/execution: if \( \pi \rightarrow_{\text{cut}} \pi' \) then \( \llbracket \pi \rrbracket_{\text{Rel}} = \llbracket \pi' \rrbracket_{\text{Rel}} \)

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

⇝ relational model can be seen as coherence semantics “without coherence”
⇝ the price to pay is that negation is invisible in **Rel**: \( A = A\perp \)

Apparently, \( \llbracket \pi \rrbracket_{\text{Rel}} \) 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

\( \leadsto \) LL is interpreted in \textbf{Rel}, the category of sets and relations

- formula/type \( A \leadsto \) a set \( A \)
- proof/program \( \pi \) of \( A \rightarrow B \leadsto \) a relation \( \llbracket \pi \rrbracket^{\text{Rel}} \subseteq A \times B \)
- invariance under cut-elimination/execution: if \( \pi \rightarrow_{\text{cut}} \pi' \) then \( \llbracket \pi \rrbracket^{\text{Rel}} = \llbracket \pi' \rrbracket^{\text{Rel}} \)

**Remark:** A coherence space \( A \) (interpreting the formula \( A \)) can be seen as a set of elements (\textit{points}) endowed with a reflexive and symmetric relation (\textit{coherence}).

The set \( A \) without the coherence relation is the interpretation of \( A \) in \textbf{Rel}.

\( \leadsto \) relational model can be seen as coherence semantics “without coherence”

\( \leadsto \) the price to pay is that negation is invisible in \textbf{Rel}: \( A = A^{\perp} \)

Apparently, \( \llbracket \pi \rrbracket^{\text{Rel}} \) is just a set of points without any structure...
Concrete denotational semantics of LL: the finiteness spaces

Ehrhard (2005) introduced \textit{finiteness spaces}, a new denotational semantics of LL:

\[
\text{Fin} = \text{the category of finiteness spaces and continuous linear functions } \rightarrow_{\text{lin}}
\]

- formula/type \( A \mapsto \text{a finiteness space } A \) (topological vector space over a field)
- proof/program \( \pi \) of \( A \rightarrow B \mapsto \text{a continuous linear map } [\pi]_{\text{Fin}} : A \rightarrow_{\text{lin}} B \)
- invariance under cut-elimination/execution: if \( \pi \rightarrow_{\text{cut}} \pi' \) then \( [\pi]_{\text{Fin}} = [\pi']_{\text{Fin}} \)

In \text{Fin}, \( ! \) is an endofunctor such that:

- proof/program \( \pi \) of \( !A \rightarrow B \mapsto \text{a continuous linear map } [\pi]_{\text{Fin}} : !A \rightarrow_{\text{lin}} B \)
  \[
  = \text{a continuous analytical map } [\pi]_{\text{Fin}} : A \rightarrow_{\text{an}} B
  \]
- the categorical structure of \( ! \) corresponds to the differential operations on these continuous analytical maps

\( \mapsto \) Any analytical map is equal to its \textit{Taylor expansion} at any point of its domain

These differential operations and the notion of Taylor expansion can be internalized in the syntax

\( \mapsto \) \textbf{Differential Linear Logic} (\textbf{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

\[ \text{Fin} = \text{the category of finiteness spaces and continuous linear functions} \]

- formula/type \( A \leadsto A \) (topological vector space over a field)
- proof/program \( \pi \) of \( A \to B \leadsto \) a continuous linear map \( \llbracket \pi \rrbracket_{\text{Fin}} : A \to_{\text{lin}} B \)
- invariance under cut-elimination/execution: if \( \pi \to_{\text{cut}} \pi' \) then \( \llbracket \pi \rrbracket_{\text{Fin}} = \llbracket \pi' \rrbracket_{\text{Fin}} \)

In \( \text{Fin} \), \( ! \) is an endofunctor such that:

- proof/program \( \pi \) of \( !A \to !B \leadsto \) a continuous linear map \( \llbracket \pi \rrbracket_{\text{Fin}} : !A \to_{\text{lin}} B \)
  \[ = \text{a continuous analytical map} \llbracket \pi \rrbracket_{\text{Fin}} : A \to_{\text{an}} B \]
- the categorical structure of \( ! \) corresponds to the *differential operations* on these continuous analytical maps

\( \leadsto \) 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 internalized in the syntax

\( \leadsto \) *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

$\textbf{Fin} = \text{the category of finiteness spaces and continuous linear functions } \rightarrow_{\text{lin}}$

- formula/type $A \rightsquigarrow$ a finiteness space $A$ (topological vector space over a field)
- proof/program $\pi$ of $A \rightarrow B \rightsquigarrow$ a continuous linear map $\llbracket \pi \rrbracket_{\text{Fin}} : A \rightarrow_{\text{lin}} B$
- invariance under cut-elimination/execution: if $\pi \rightarrow_{\text{cut}} \pi'$ then $\llbracket \pi \rrbracket_{\text{Fin}} = \llbracket \pi' \rrbracket_{\text{Fin}}$

In $\textbf{Fin}$, $!$ is an endofunctor such that:

- proof/program $\pi$ of $!A \rightarrow B \rightsquigarrow$ a continuous linear map $\llbracket \pi \rrbracket_{\text{Fin}} : !A \rightarrow_{\text{lin}} B$
  
  $= \text{a continuous analytical map } \llbracket \pi \rrbracket_{\text{Fin}} : A \rightarrow_{\text{an}} B$

- 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 internalized in the syntax

$\rightsquigarrow$ Differential Linear Logic (DiLL$_0$), introduced by Ehrhard and Regnier (2006)
Differential Linear Logic (Ehrhard & Regnier [2006])

\[ \text{DiLL}_0 \text{ formulas } = \text{MELL formulas} \]

\[ \text{DiLL}_0 \text{ proof-structures } = \text{the same as MELL except for the rules introducing } \Rightarrow \text{(infinitary) boxes are replaced by three new kind of (finitary) nodes:} \]

- Co-dereliction (\(!d\)) releases inputs of type \(!A\) that can be called \textit{exactly once} (i.e. \textit{linearly}) during the cut elimination process
- Thanks to co-contraction (\(!c\)) and co-weakening (\(!w\)), in DiLL$_0$ every resource is available only \textit{finitely many times}.

\textbf{Idea:} a proof is an \textit{infinite} (formal) \textit{sum} of graphs, cut elimination is a \textit{local} graph rewriting.
Differential Linear Logic (Ehrhard & Regnier [2006])

\(\text{DiLL}_0\) formulas = MELL formulas
\(\text{DiLL}_0\) proof-structures = the same as MELL except for the rules introducing \(!\)\(\rightsquigarrow\) (infinitary) boxes are replaced by three new kind of (finitary) nodes:

- Co-dereliction (\(!d\)) releases inputs of type \(!A\) that can be called \textit{exactly once} (i.e. \textit{linearly}) during the cut elimination process
- Thanks to co-contraction (\(!c\)) and co-weakening (\(!w\)), in \(\text{DiLL}_0\) every resource is available only \textit{finitely many times}.

**Idea:** a proof is an \textit{infinite} (formal) \textit{sum} of graphs, cut elimination is a \textit{local} graph rewriting
Examples of co-structural cut-elimination steps  (1 of 2)

A DiLL₀ proof-structure reduces to a finite set of DiLL₀ proof-structures

![Diagram showing co-structural cut-elimination steps]

Guerrieri, Pellissier, Tortora de Falco  Injectivity of REL for connected MELL via TACL, 21/6/2015 15 / 24
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

\[
\pi \\
\downarrow \quad \downarrow \\
A \perp \quad A \perp \\
\downarrow \quad \downarrow \\
?d \quad ?d \\
\downarrow \quad \downarrow \\
?A \perp \quad ?A \perp \\
\downarrow \quad \downarrow \\
?c \quad ?c \\
\downarrow \quad \downarrow \\
?A \perp \\
\end{align*}

\[
\pi_1 \\
\downarrow \\
A \\
\downarrow \\
!d \\
\downarrow \\
!A \\
\downarrow \\
!c \\
\downarrow \\
!A \\
\end{align*}

\[
\pi_2 \\
\downarrow \\
A \\
\downarrow \\
!d \\
\downarrow \\
!A \\
\downarrow \\
!c \\
\downarrow \\
!A \\
\end{align*}

\[
\pi \\
\downarrow \\
A \perp \\
\downarrow \\
!A \\
\downarrow \\
\begin{array}{c}
\text{cut}
\end{array} \\
\downarrow \\
A \perp \\
\end{align*}

\[
\pi_1 \\
\downarrow \\
A \perp \\
\downarrow \\
\begin{array}{c}
\text{cut}
\end{array} \\
\downarrow \\
A \\
\end{align*}

\[
\pi_2 \\
\downarrow \\
A \\
\downarrow \\
\begin{array}{c}
\text{cut}
\end{array} \\
\downarrow \\
A \\
\end{align*}

\[
\pi \\
\downarrow \\
A \perp \\
\downarrow \\
\begin{array}{c}
\text{cut}
\end{array} \\
\downarrow \\
A \perp \\
\end{align*}

\[
\pi_1 \\
\downarrow \\
A \perp \\
\downarrow \\
\begin{array}{c}
\text{cut}
\end{array} \\
\downarrow \\
A \\
\end{align*}

\[
\pi_2 \\
\downarrow \\
A \\
\downarrow \\
\begin{array}{c}
\text{cut}
\end{array} \\
\downarrow \\
A \\
\end{align*}

\[
\pi \\
\downarrow \\
A \perp \\
\downarrow \\
\begin{array}{c}
\text{cut}
\end{array} \\
\downarrow \\
A \perp \\
\end{align*}

\[
\pi_1 \\
\downarrow \\
A \perp \\
\downarrow \\
\begin{array}{c}
\text{cut}
\end{array} \\
\downarrow \\
A \\
\end{align*}

\[
\pi_2 \\
\downarrow \\
A \\
\downarrow \\
\begin{array}{c}
\text{cut}
\end{array} \\
\downarrow \\
A \\
\end{align*}
Examples of co-structural cut-elimination steps (2 of 2)

Why? Because there is a mismatch: $\pi$ ask for 2 copies of a resource, but only 1 copy ($\pi_1$) is available and it cannot be duplicated.
Examples of co-structural cut-elimination steps  (2 of 2)

Why? Because there is a mismatch: $\pi$ ask for 2 copies of a resource, but only 1 copy ($\pi_1$) is available and it cannot be duplicated.
Taylor expansion of a MELL proof-structure

Taylor expansion $\mathcal{T} : \ \text{MELL} \rightarrow \mathcal{P}(\text{DiLL}_0)\quad \pi \mapsto \mathcal{T}(\pi)$

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 $\pi$)
Taylor expansion of a MELL proof-structure

Taylor expansion $\mathcal{T} : \text{MELL} \to \mathcal{P}(\text{DiLL}_0)$

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

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

Example: $\pi = \begin{array}{c}
\text{ax} \\
A \\
?d \\
?A \\
?d \\
??A \\
adux \\
??A \\
\end{array} \begin{array}{c}
\downarrow \\
\downarrow \\
\downarrow \\
\downarrow \\
\downarrow \\
\downarrow \\
\text{!} \\
\end{array} \begin{array}{c}
A \downarrow \\
?A \downarrow \\
?d \downarrow \\
??A \downarrow \\
!A \downarrow \\
\end{array}$
Taylor expansion of a MELL proof-structure

Taylor expansion $\mathcal{T}: \text{MELL} \rightarrow \mathcal{P}(\text{DiLL}_0)$

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

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

**Example:**

$\mathcal{T}(\pi) = \left\{ \begin{array}{c}
?w & !w, \\
?A & !A^\perp \\
?d & !d, \\
?A & !A^\perp \\
??A & \\
\vdots \\
\end{array} \right\}$
Remark: A finiteness space (interpreting the formula $A$ in $\text{Fin}$) 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 $\text{Rel}$, and

$$[[\pi]]_{\text{Fin}} = [[\pi]]_{\text{Rel}} (\neq [[\pi]]_{\text{Coh}})$$

Summing up:
- $[[\pi]]_{\text{Fin}} : \mathcal{A} \rightarrow_{\text{an}} \mathcal{B}$ is its Taylor expansion
- the Taylor expansion can be internalized in the syntax
- the interpretation of a proof in $\text{Fin}$ and in $\text{Rel}$ is the same

It is natural to expect a relationship between the Taylor expansion of a MELL-proof-structure $\pi$ and the interpretation of $\pi$ in $\text{Rel}$.
Taylor expansion: bridge between syntax and semantics

Remark: A finiteness space (interpreting the formula $A$ in $\text{Fin}$) 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 $\text{Rel}$, and

$$\llbracket \pi \rrbracket^\text{Fin} = \llbracket \pi \rrbracket^\text{Rel} (\neq \llbracket \pi \rrbracket^\text{Coh})$$

Summing up:

- $\llbracket \pi \rrbracket^\text{Fin} : \mathcal{A} \to \text{an } \mathcal{B}$ is its Taylor expansion
- the Taylor expansion can be internalized in the syntax
- the interpretation of a proof in $\text{Fin}$ and in $\text{Rel}$ is the same

$\Rightarrow$ It is natural to expect a relationship between the Taylor expansion of a MELL-proof-structure $\pi$ and the interpretation of $\pi$ in $\text{Rel}$
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 \(\pi\):

\[
\llbracket \pi \rrbracket_{\text{Rel inj}/\sim} \cong \mathcal{T}(\pi).
\]

where \(\llbracket \pi \rrbracket_{\text{Rel inj}/\sim} \subseteq \llbracket \pi \rrbracket_{\text{Rel}}\), more precisely:

- \(\llbracket \pi \rrbracket_{\text{inj}}\) = the set containing the “most informative” points of \(\llbracket \pi \rrbracket_{\text{Rel}}\)
- \(\sim\) is an equivalence relation on the points based on renaming of atoms

Given a normal MELL proof-structure \(\pi\), the proposition above allows us to deal with the elements of \(\mathcal{T}(\pi)\) instead the points of \(\llbracket \pi \rrbracket_{\text{Rel}}\)

\(\Rightarrow\) a geometrical representation of the points of the relational semantics of \(\pi\).

\([\pi] = \begin{cases} 
\text{a morphism in some category (Rel, Fin, Coh, \ldots)} \\
\text{a (infinite) set of points (of a set, finiteness space, coherence space, \ldots)} \\
\text{a (infinite) set of graphs (the DiLL}_{0}\text{-proof-structures of } \mathcal{T}(\pi))
\end{cases}\)
Our first “contribution”:

Proposition (G., P., TdF., but also “folklore”)

For every normal (= cut-free, with atomic axioms) MELL-proof-structure $\pi$:

$$\llbracket \pi \rrbracket_{rel}^{\text{inj}/\sim} \cong \mathcal{T}(\pi).$$

where $\llbracket \pi \rrbracket_{\text{inj}/\sim} \subseteq \llbracket \pi \rrbracket_{\text{rel}}$, more precisely:

- $\llbracket \pi \rrbracket_{\text{inj}}$ = the set containing the “most informative” points of $\llbracket \pi \rrbracket_{\text{rel}}$
- $\sim$ is an equivalence relation on the points based on renaming of atoms

Given a normal MELL proof-structure $\pi$, the proposition above allows us to deal with the elements of $\mathcal{T}(\pi)$ instead the points of $\llbracket \pi \rrbracket_{\text{rel}}$

$\leadsto$ a geometrical representation of the points of the relational semantics of $\pi$. 

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

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

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

- $[\pi]^\text{Rel}_{\text{inj}} = \text{the set containing the “most informative” points of } [\pi]^\text{Rel}$
- $\sim = \text{an equivalence relation on the points based on renaming of atoms}$

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

$\Rightarrow$ a geometrical representation of the points of the relational semantics of $\pi$.

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

2 From linear logic to differential linear logic and Taylor expansion

3 The question of injectivity
The question of injectivity and its motivations

The question of injectivity

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

$$[\pi] = [\pi'] \Rightarrow \pi = \pi'.$$

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 $\lambda$-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 $\eta$-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 $\pi_1, \pi_2$ are normal

$$\Rightarrow [\pi_1] = [\pi] = [\pi_2] \Rightarrow \pi_1 = \pi_2.$$
The question of injectivity and its motivations

The question of injectivity

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

$$[\pi] = [\pi'] \implies \pi = \pi'.$$

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 $\lambda$-calculi and term rewriting systems (Friedman '75, Statman '82).

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

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

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

where $\pi_1, \pi_2$ are normal $\Rightarrow$ $[\pi_1] = [\pi] = [\pi_2] \implies \pi_1 = \pi_2$
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 $\pi_1$ and $\pi_2$ such that $[\pi_1]^{\text{Coh}} = [\pi_2]^{\text{Coh}}$ 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 $\pi_1$ and $\pi_2$ without $\\w$ nodes, if $[\pi_1]^{\text{Rel}} = [\pi_2]^{\text{Rel}}$ 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 $\pi_1$ and $\pi_2$ such that $[\pi_1]^{\text{Coh}} = [\pi_2]^{\text{Coh}}$ 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 $\pi_1$ and $\pi_2$ without $?w$ nodes, if $[\pi_1]^{\text{Rel}} = [\pi_2]^{\text{Rel}}$ 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 $\pi_1$ and $\pi_2$ be two normal MELL proof-structures such that $[\pi_1] = [\pi_2]$ (where $[\cdot]$ is a set-based semantics: $[\cdot]^{\text{Rel}}, [\cdot]^{\text{Coh}}, [\cdot]^{\text{Fin}}, \ldots$). Set-based model $\iff$ For any MELL proof-structure $\pi$, $[\pi]$ is a set.

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

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

3 **Conclusion (injectivity):** Since $[\pi_1] = [\pi_2]$, then the "discriminating element" $\rho$ 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" $\rho$ of any pair $\{\pi, \pi'\}$ of normal MELL proof-structures.

**Remark:** the "structure" of $\rho$ depends on $\pi, \pi'$ (the Key-Lemma claims: "$\forall \{\pi, \pi'\} \exists \rho$ such that if $P(\rho, \pi, \pi')$ then $\rho$ is discriminating for $\pi, \pi'$")

- To build univocally $\pi = \pi'$ from $\rho$ satisfying $P(\rho, \pi, \pi')$, i.e. to prove the "discriminating power" of such a $\rho$. 

How to prove injectivity of a set-based model?

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

Set-based model \( \rightsquigarrow \) For any MELL proof-structure \( \pi \), \([\pi]\) is a set.

2 Key-Lemma:

For any normal MELL proof-structures \( \pi \), there is at least one “discriminating element” \( \rho \in [\pi] \), i.e., for any normal MELL-proof-structure \( \pi' \)

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

3 Conclusion (injectivity): Since \( [\pi_1] = [\pi_2] \), then the “discriminating element” \( \rho \) 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” \( \rho \) of any pair \( \{\pi, \pi'\} \) of normal MELL proof-structures.

Remark: the “structure” of \( \rho \) depends on \( \pi, \pi' \) (the Key-Lemma claims: “\( \forall \{\pi, \pi'\} \exists \rho \) such that if \( P(\rho, \pi, \pi') \) then \( \rho \) is discriminating for \( \pi, \pi' \)”)

- To build univocally \( \pi = \pi' \) from \( \rho \) satisfying \( P(\rho, \pi, \pi') \), i.e. to prove the “discriminating power” of such a \( \rho \).
How to prove injectivity of a set-based model?

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

Set-based model $\iff$ For any MELL proof-structure $\pi$, $[[\pi]]$ is a set.

2 Key-Lemma:
For any normal MELL proof-structures $\pi$, there is at least one “discriminating element” $\rho \in [[\pi]]$, i.e., for any normal MELL-proof-structure $\pi'$

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

3 Conclusion (injectivity): Since $[[\pi_1]] = [[\pi_2]]$, then the “discriminating element” $\rho$ 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” $\rho$ of any pair $\{\pi, \pi'\}$ of normal MELL proof-structures.

Remark: the “structure” of $\rho$ depends on $\pi, \pi'$ (the Key-Lemma claims: “$\forall \{\pi, \pi'\} \exists \rho$ such that if $P(\rho, \pi, \pi')$ then $\rho$ is discriminating for $\pi, \pi'$”)

- To build univocally $\pi = \pi'$ from $\rho$ satisfying $P(\rho, \pi, \pi')$, i.e. to prove the “discriminating power” of such a $\rho$. 

How to prove injectivity of a set-based model?

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

Set-based model $\iff$ For any MELL proof-structure $\pi$, $[\pi]$ is a set.

2. **Key-Lemma:**

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

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

3. **Conclusion (injectivity):** Since $[\pi_1] = [\pi_2]$, then the “discriminating element” $\rho$ 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” $\rho$ of any pair $\{\pi, \pi'\}$ of normal MELL proof-structures.

**Remark:** the “structure” of $\rho$ *depends* on $\pi, \pi'$ (the Key-Lemma claims: “$\forall \{\pi, \pi'\} \exists \rho$ such that if $P(\rho, \pi, \pi')$ then $\rho$ is discriminating for $\pi, \pi'$”)

- To build univocally $\pi = \pi'$ from $\rho$ satisfying $P(\rho, \pi, \pi')$, i.e. to prove the “discriminating power” of such a $\rho$. 

Guerrieri, Pellissier, Tortora de Falco
Injectivity of REL for connected MELL via T
TACL, 21/6/2015
How to prove injectivity of a set-based model?

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

   Set-based model $\iff$ For any MELL proof-structure $\pi$, $[\pi]$ is a set.

2. **Key-Lemma:**
   For any normal MELL proof-structures $\pi$, there is at least one “discriminating element” $\rho \in [\pi]$, i.e., for any normal MELL-proof-structure $\pi'$
   
   \[
   \text{if } \rho \in [\pi] \cap [\pi'], \text{ then } \pi = \pi'.
   \]

3. **Conclusion (injectivity):** Since $[\pi_1] = [\pi_2]$, then the “discriminating element” $\rho$ 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” $\rho$ of any pair $\{\pi, \pi'\}$ of normal MELL proof-structures.

  **Remark:** the “structure” of $\rho$ *depends* on $\pi, \pi'$ (the Key-Lemma claims: “$\forall \{\pi, \pi'\} \exists \rho$ such that if $P(\rho, \pi, \pi')$ then $\rho$ is discriminating for $\pi, \pi'$”)

- To build univocally $\pi = \pi'$ from $\rho$ satisfying $P(\rho, \pi, \pi')$, i.e. to prove the “discriminating power” of such a $\rho$. 
Our contribution

Inspired by the proposition relating the relational interpretation and the Taylor expansion in the cut-free case \((\llbracket \pi \rrbracket_{\text{inj}} / \sim \cong \mathcal{T}(\pi))\), we used DiLL\(_0\)-proof-structures to study the question of injectivity of \(\text{Rel} \) 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 \(\pi_1\) and \(\pi_2\) 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 \(\pi_1\) and \(\pi_2\) which are box-connected, if \(\llbracket \pi_1 \rrbracket_{\text{Rel}} = \llbracket \pi_2 \rrbracket_{\text{Rel}}\) then \(\pi_1 = \pi_2\).

Our novelties:

- it is a simplification, since the “structure” of the discriminating element \(\rho\) does not depend on \(\pi, \pi'\). Thus the Key-Lemma has a logically less complex claim: “\(\forall \rho \in \text{DiLL}_0\) 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 \(\pi\) also in presence of cuts.
Our contribution

Inspired by the proposition relating the relational interpretation and the Taylor expansion in the cut-free case \( (\mathbb{[\pi]}_{\text{inj}} \sim \mathcal{T}(\pi)) \), we used DiLL\(_0\)-proof-structures to study the question of injectivity of Rel wrt MELL-proof-structures 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 \( \pi_1 \) and \( \pi_2 \) 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 \( \pi_1 \) and \( \pi_2 \) which are box-connected, if \( \mathbb{[\pi_1]}_{\text{Rel}} = \mathbb{[\pi_2]}_{\text{Rel}} \) then \( \pi_1 = \pi_2 \).

Our novelties:

- it is a simplification, since the “structure” of the discriminating element \( \rho \) does not depend on \( \pi, \pi' \). Thus the Key-Lemma has a logically less complex claim: “\( \forall \rho \in \text{DiLL}_0 \) 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 \( \pi \) also in presence of cuts.
Our contribution

Inspired by the proposition relating the relational interpretation and the Taylor expansion in the cut-free case ($\frac{[\pi]_{\text{inj}}^{\text{Rel}}}{\sim} \equiv \mathcal{T}(\pi)$), 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 $\pi_1$ and $\pi_2$ 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 $\pi_1$ and $\pi_2$ which are box-connected, if $[\pi_1]^{\text{Rel}} = [\pi_2]^{\text{Rel}}$ then $\pi_1 = \pi_2$.

Our novelties:

- it is a simplification, since the “structure” of the discriminating element $\rho$ does not depend on $\pi, \pi'$. Thus the Key-Lemma has a logically less complex claim: “$\forall \rho \in \text{DiLL}_0$ 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 $\pi$ also in presence of cuts.
Our contribution

Inspired by the proposition relating the relational interpretation and the Taylor expansion in the cut-free case (\(\llbracket \pi \rrbracket_{\text{inj/\sim}} \simeq \mathcal{T}(\pi)\)), we used DiLL\(_0\)-proof-structures to study the question of injectivity of Rel wrt MELL-proof-structures \(\Rightarrow\) 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 \(\pi_1\) and \(\pi_2\) 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 \(\pi_1\) and \(\pi_2\) which are box-connected, if \(\llbracket \pi_1 \rrbracket_{\text{Rel}} = \llbracket \pi_2 \rrbracket_{\text{Rel}}\) then \(\pi_1 = \pi_2\).

Our novelties:

- it is a simplification, since the “structure” of the discriminating element \(\rho\) does not depend on \(\pi, \pi'\). Thus the Key-Lemma has a logically less complex claim: “\(\forall \rho \in \text{DiLL}_0\) 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 \(\pi\) also in presence of cuts.