Taylor expansion of proofs and static analysis of time complexity

Daniel de Carvalho

Innopolis University

Yaroslavl, 21st of June, 2018

Some ideas

- Proofs are programs (Curry, Howard, de Bruijn, Girard)

Some ideas

- Proofs are programs (Curry, Howard, de Bruijn, Girard)
- Programs are continuous functions between domains (Scott)

Some ideas

- Proofs are programs (Curry, Howard, de Bruijn, Girard)
- Programs are continuous functions between domains (Scott)
- Sequential programs are stable functions between dI-domains (Berry)

Some ideas

- Proofs are programs (Curry, Howard, de Bruijn, Girard)
- Programs are continuous functions between domains (Scott)
- Sequential programs are stable functions between dI-domains (Berry)
- Sequential programs are stable functions between coherence spaces (Girard)

Some ideas

• Proofs are programs (Curry, Howard, de Bruijn, Girard)
• Programs are continuous functions between domains (Scott)
• Sequential programs are stable functions between dI-domains (Berry)
• Sequential programs are stable functions between coherence spaces (Girard)
• What linear logic teaches us: Stable functions are linear functions (!) (Girard)

Some ideas

• Proofs are programs (Curry, Howard, de Bruijn, Girard)
• Programs are continuous functions between domains (Scott)
• Sequential programs are stable functions between dI-domains (Berry)
• Sequential programs are stable functions between coherence spaces (Girard)
• What linear logic teaches us: Stable functions are linear functions (!) (Girard)
• Some consequences:

Some ideas

- Proofs are programs (Curry, Howard, de Bruijn, Girard)
- Programs are continuous functions between domains (Scott)
- Sequential programs are stable functions between dI-domains (Berry)
- Sequential programs are stable functions between coherence spaces (Girard)
- What linear logic teaches us: Stable functions are linear functions (!) (Girard)
- Some consequences:

  - Refining stability: a) hypercoherences (Ehrhard)
  b) game semantics for linear logic -> full abstraction of PCF (Blass, Abramsky, Jagadeesan, Malacaria) (not covered)

Some ideas

- Proofs are programs (Curry, Howard, de Bruijn, Girard)
- Programs are continuous functions between domains (Scott)
- Sequential programs are stable functions between dI-domains (Berry)
- Sequential programs are stable functions between coherence spaces (Girard)
- What linear logic teaches us: Stable functions are linear functions (!) (Girard)
- Some consequences:
  - Refining stability: a) hypercoherences (Ehrhard) b) game semantics for linear logic -> full abstraction of PCF (Blass, Abramsky, Jagadeesan, Malacaria) (not covered)
  - Programs are smooth functions (Ehrhard)

Some ideas

- Proofs are programs (Curry, Howard, de Bruijn, Girard)
- Programs are continuous functions between domains (Scott)
- Sequential programs are stable functions between dI-domains (Berry)
- Sequential programs are stable functions between coherence spaces (Girard)
- What linear logic teaches us: Stable functions are linear functions (!) (Girard)
- Some consequences:
  - Refining stability: a) hypercoherences (Ehrhard) b) game semantics for linear logic -> full abstraction of PCF (Blass, Abramsky, Jagadeesan, Malacaria) (not covered)
  - Programs are smooth functions (Ehrhard)
  - Foundations for verification of time complexity of programs

Some ideas

- Proofs are programs (Curry, Howard, de Bruijn, Girard)
- Programs are continuous functions between domains (Scott)
- Sequential programs are stable functions between dI-domains (Berry)
- Sequential programs are stable functions between coherence spaces (Girard)
- What linear logic teaches us: Stable functions are linear functions (!) (Girard)
- Some consequences:
  - Refining stability: a) hypercoherences (Ehrhard) b) game semantics for linear logic -> full abstraction of PCF (Blass, Abramsky, Jagadeesan, Malacaria) (not covered)
  - Programs are smooth functions (Ehrhard)
  - Foundations for verification of time complexity of programs
  - Logical implicit complexity (Girard, Lafont, Baillot, Terui, Hofmann) (not covered)

Proofs matter (1): A trivial proof of (Int $\Rightarrow$ Int)

Let us consider natural deduction for second-order intuitionistisc logic.
Let Int be the formula $(\forall X)((X \Rightarrow X) \Rightarrow (X \Rightarrow X))$.
There is a trivial proof of (Int $\Rightarrow$ Int) (let us call it id):

Proofs matter (1): A trivial proof of (Int $\Rightarrow$ Int)

Let us consider natural deduction for second-order intuitionistisc logic.
Let Int be the formula $(\forall X)((X \Rightarrow X) \Rightarrow (X \Rightarrow X))$.
There is a trivial proof of (Int $\Rightarrow$ Int) (let us call it id):

$$\dfrac{\overline{t : \mathrm{Int} \vdash \mathrm{Int}}}{\vdash (\mathrm{Int} \Rightarrow \mathrm{Int})} \Rightarrow_i, t$$

Proofs matter (1): A trivial proof of $(\mathsf{Int} \Rightarrow \mathsf{Int})$

Let us consider natural deduction for second-order intuitionistisc logic.
Let Int be the formula $(\forall X)((X \Rightarrow X) \Rightarrow (X \Rightarrow X))$.
There is a trivial proof of $(\mathsf{Int} \Rightarrow \mathsf{Int})$ (let us call it id):

$$\dfrac{\overline{t : \mathsf{Int} \vdash \mathsf{Int}}}{\vdash (\mathsf{Int} \Rightarrow \mathsf{Int})} \Rightarrow_i, t$$

There are many other proofs of this formula, but why should we take an interest in them after all?!
Let us consider one of them.

Proofs matter (2): A non-trivial proof of $(\mathrm{Int} \Rightarrow \mathrm{Int})$

Another proof of $(\mathrm{Int} \Rightarrow \mathrm{Int})$ (let us call it s):

$$
\begin{prooftree}
\AxiomC{}
\UnaryInfC{$y : Y \vdash Y$}
\AxiomC{}
\UnaryInfC{$t : \mathrm{Int} \vdash \mathrm{Int}$}
\RightLabel{$\forall_e$}
\UnaryInfC{$t : \mathrm{Int} \vdash (Y \Rightarrow Y)$}
\AxiomC{}
\UnaryInfC{$y : Y \vdash Y$}
\RightLabel{$\Rightarrow_e$}
\BinaryInfC{$t : \mathrm{Int}, y : Y \vdash Y$}
\AxiomC{}
\UnaryInfC{$x : X \vdash X$}
\RightLabel{$\Rightarrow_e$}
\BinaryInfC{$\Gamma \vdash X$}
\RightLabel{$\Rightarrow_e$}
\BinaryInfC{$\Gamma \vdash X$}
\RightLabel{$\Rightarrow_i, x$}
\UnaryInfC{$t : \mathrm{Int}, y : Y \vdash Y$}
\RightLabel{$\Rightarrow_i, y$}
\UnaryInfC{$t : \mathrm{Int} \vdash (Y \Rightarrow Y)$}
\RightLabel{$\forall_i$}
\UnaryInfC{$t : \mathrm{Int} \vdash \mathrm{Int}$}
\RightLabel{$\Rightarrow_i, t$}
\UnaryInfC{$\vdash (\mathrm{Int} \Rightarrow \mathrm{Int})$}
\end{prooftree}
$$

where $Y = (X \Rightarrow X)$ and $\Gamma = t : \mathrm{Int}, y : Y, x : X$.
What is the point to consider such a proof?

Proofs matter (3): Many proofs of Int

We have the following proof (let us call it $\underline{0}$):

$$\dfrac{\dfrac{\dfrac{}{x : X \vdash X}}{\vdash (X \Rightarrow X)} \Rightarrow_i, x}{\dfrac{\vdash ((X \Rightarrow X) \Rightarrow (X \Rightarrow X))}{\vdash \mathrm{Int}} \forall_i} \Rightarrow_i, y$$

Proofs matter (3): Many proofs of Int

We have the following proof (let us call it $\underline{0}$):

$$\cfrac{\cfrac{\cfrac{\overline{x : X \vdash X}}{\vdash (X \Rightarrow X)} \Rightarrow_i, x}{\vdash ((X \Rightarrow X) \Rightarrow (X \Rightarrow X))} \Rightarrow_i, y}{\vdash \mathsf{Int}} \forall_i$$

Also, we have the following proof (let us call it $\underline{1}$):

$$\cfrac{\cfrac{\cfrac{\cfrac{\overline{y : (X \Rightarrow X) \vdash (X \Rightarrow X)} \qquad \overline{x : X \vdash X}}{y : (X \Rightarrow X), x : X \vdash X} \Rightarrow_e}{y : (X \Rightarrow X) \vdash (X \Rightarrow X)} \Rightarrow_i, x}{\vdash ((X \Rightarrow X) \Rightarrow (X \Rightarrow X))} \Rightarrow_i, y}{\vdash \mathsf{Int}} \forall_i$$

Proofs matter (4): Cuts

From a proof of (Int $\Rightarrow$ Int) and a proof of Int, we can get a new proof of Int.
For instance, taking $s : (\mathrm{Int} \Rightarrow \mathrm{Int})$ and $\underline{0} : \mathrm{Int}$, we get:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{y : Y \vdash Y} \quad \dfrac{\dfrac{\dfrac{\dfrac{}{t : \mathrm{Int} \vdash \mathrm{Int}}}{t : \mathrm{Int} \vdash (Y \Rightarrow Y)}\;\forall_e \quad \dfrac{}{y : Y \vdash Y}}{t : \mathrm{Int}, y : Y \vdash Y}\;\Rightarrow_e \quad \dfrac{}{x : X \vdash X}}{\Gamma \vdash X}\;\Rightarrow_e}{\Gamma \vdash X}\;\Rightarrow_e}{t : \mathrm{Int}, y : Y \vdash Y}\;\Rightarrow_i, x}{t : \mathrm{Int} \vdash (Y \Rightarrow Y)}\;\Rightarrow_i, y}{t : \mathrm{Int} \vdash \mathrm{Int}}\;\forall_i}{\vdash (\mathrm{Int} \Rightarrow \mathrm{Int})}\;\Rightarrow_i, t \quad \dfrac{\dfrac{\dfrac{\dfrac{}{x : X \vdash X}}{\vdash Y}\;\Rightarrow_i, x}{\vdash (Y \Rightarrow Y)}\;\Rightarrow_i, y}{\vdash \mathrm{Int}}\;\forall_i}{\vdash \mathrm{Int}}\;\Rightarrow_e$$

where $Y = (X \Rightarrow X)$ and $\Gamma = t : \mathrm{Int}, y : Y, x : X$.

Proofs matter (4): Cuts

From a proof of $(\mathsf{Int} \Rightarrow \mathsf{Int})$ and a proof of Int, we can get a new proof of Int.
For instance, taking $s : (\mathsf{Int} \Rightarrow \mathsf{Int})$ and $\underline{0} : \mathsf{Int}$, we get:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{y : Y \vdash Y} \quad \dfrac{\dfrac{\dfrac{\dfrac{}{t : \mathsf{Int} \vdash \mathsf{Int}}}{t : \mathsf{Int} \vdash (Y \Rightarrow Y)}\,\forall_e \quad \dfrac{}{y : Y \vdash Y}}{t : \mathsf{Int}, y : Y \vdash Y}\,\Rightarrow_e \quad \dfrac{}{x : X \vdash X}}{\Gamma \vdash X}\,\Rightarrow_e}{\Gamma \vdash X}\,\Rightarrow_e}{t : \mathsf{Int}, y : Y \vdash Y}\,\Rightarrow_{i}, x}{t : \mathsf{Int} \vdash (Y \Rightarrow Y)}\,\Rightarrow_{i}, y}{t : \mathsf{Int} \vdash \mathsf{Int}}\,\forall_i}{\vdash (\mathsf{Int} \Rightarrow \mathsf{Int})}\,\Rightarrow_{i}, t \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{x : X \vdash X}}{\vdash Y}\,\Rightarrow_{i}, x}{\vdash (Y \Rightarrow Y)}\,\Rightarrow_{i}, y}{\vdash \mathsf{Int}}\,\forall_i}{\vdash \mathsf{Int}}\,\Rightarrow_e$$

where $Y = (X \Rightarrow X)$ and $\Gamma = t : \mathsf{Int}, y : Y, x : X$.

Proofs matter (5): Cut-elimination of s applied to $\underline{0}$

$$
\dfrac{
  \dfrac{}{y : Y \vdash Y}
  \qquad
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{
          \dfrac{
            \dfrac{
              \dfrac{}{x : X \vdash X}
            }{\vdash Y}\;{\Rightarrow_i, x}
          }{\vdash (Y \Rightarrow Y)}\;{\Rightarrow_i, y}
        }{\vdash \mathrm{Int}}\;{\forall_i}
      }{\vdash (Y \Rightarrow Y)}\;{\forall_e}
      \qquad
      \dfrac{}{y : Y \vdash Y}
    }{y : Y \vdash Y}\;{\Rightarrow_e}
    \qquad
    \dfrac{}{x : X \vdash X}
  }{y : Y, x : X \vdash X}\;{\Rightarrow_e}
}{
  \dfrac{
    \dfrac{
      \dfrac{y : Y, x : X \vdash X}{y : Y \vdash Y}\;{\Rightarrow_i, x}
    }{\vdash (Y \Rightarrow Y)}\;{\Rightarrow_i, y}
  }{\vdash \mathrm{Int}}\;{\forall_i}
}\;{\Rightarrow_e}
$$

where $Y = (X \Rightarrow X)$.

Proofs matter (5): Cut-elimination of s applied to $\underline{0}$

$$
\begin{prooftree}
\AxiomC{$y : Y \vdash Y$}
\AxiomC{}
\UnaryInfC{$x : X \vdash X$}
\RightLabel{$\Rightarrow_i, x$}
\UnaryInfC{$\vdash Y$}
\RightLabel{$\Rightarrow_i, y$}
\UnaryInfC{$\color{red}{\vdash (Y \Rightarrow Y)}$}
\RightLabel{$\color{red}{\forall_i}$}
\UnaryInfC{$\color{red}{\vdash \mathrm{Int}}$}
\RightLabel{$\color{red}{\forall_e}$}
\UnaryInfC{$\color{red}{\vdash (Y \Rightarrow Y)}$}
\AxiomC{}
\UnaryInfC{$y : Y \vdash Y$}
\RightLabel{$\Rightarrow_e$}
\BinaryInfC{$y : Y \vdash Y$}
\AxiomC{}
\UnaryInfC{$x : X \vdash X$}
\RightLabel{$\Rightarrow_e$}
\BinaryInfC{$y : Y, x : X \vdash X$}
\RightLabel{$\Rightarrow_e$}
\BinaryInfC{$y : Y, x : X \vdash X$}
\RightLabel{$\Rightarrow_i, x$}
\UnaryInfC{$y : Y \vdash Y$}
\RightLabel{$\Rightarrow_i, y$}
\UnaryInfC{$\vdash (Y \Rightarrow Y)$}
\RightLabel{$\forall_i$}
\UnaryInfC{$\vdash \mathrm{Int}$}
\end{prooftree}
$$

where $Y = (X \Rightarrow X)$.

Proofs matter (5): Cut-elimination of s applied to $\underline{0}$

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{y : Y \vdash Y} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x : X \vdash X}}{\vdash Y}\Rightarrow_i, x}{\vdash (Y \Rightarrow Y)}\Rightarrow_i, y \qquad \dfrac{}{y : Y \vdash Y}}{y : Y \vdash Y}\Rightarrow_e \qquad \dfrac{}{x : X \vdash X}}{y : Y, x : X \vdash X}\Rightarrow_e}{y : Y, x : X \vdash X}\Rightarrow_e}{y : Y \vdash Y}\Rightarrow_i, x}{\vdash (Y \Rightarrow Y)}\Rightarrow_i, y}{\vdash \mathrm{Int}}\forall_i
$$

where $Y = (X \Rightarrow X)$.

Proofs matter (5): Cut-elimination of s applied to $\underline{0}$

$$
\begin{prooftree}
\AxiomC{}
\UnaryInfC{$y : Y \vdash Y$}
\AxiomC{}
\UnaryInfC{$x : X \vdash X$}
\RightLabel{$\Rightarrow_i, x$}
\UnaryInfC{$\vdash Y$}
\RightLabel{$\Rightarrow_i, y$}
\UnaryInfC{$\vdash (Y \Rightarrow Y)$}
\AxiomC{}
\UnaryInfC{$y : Y \vdash Y$}
\RightLabel{$\Rightarrow_e$}
\BinaryInfC{$y : Y \vdash Y$}
\AxiomC{}
\UnaryInfC{$x : X \vdash X$}
\RightLabel{$\Rightarrow_e$}
\BinaryInfC{$y : Y, x : X \vdash X$}
\RightLabel{$\Rightarrow_e$}
\BinaryInfC{$y : Y, x : X \vdash X$}
\RightLabel{$\Rightarrow_i, x$}
\UnaryInfC{$y : Y \vdash Y$}
\RightLabel{$\Rightarrow_i, y$}
\UnaryInfC{$\vdash (Y \Rightarrow Y)$}
\RightLabel{$\forall_i$}
\UnaryInfC{$\vdash \mathrm{Int}$}
\end{prooftree}
$$

where $Y = (X \Rightarrow X)$.

Proofs matter (5): Cut-elimination of s applied to $\underline{0}$

$$
\begin{prooftree}
\AxiomC{}
\UnaryInfC{$y : Y \vdash Y$}
\AxiomC{}
\UnaryInfC{$x : X \vdash X$}
\RightLabel{$\Rightarrow_i, x$}
\UnaryInfC{$\vdash Y$}
\AxiomC{}
\UnaryInfC{$x : X \vdash X$}
\RightLabel{$\Rightarrow_e$}
\BinaryInfC{$x : X \vdash X$}
\RightLabel{$\Rightarrow_e$}
\BinaryInfC{$y : Y, x : X \vdash X$}
\RightLabel{$\Rightarrow_i, x$}
\UnaryInfC{$y : Y \vdash Y$}
\RightLabel{$\Rightarrow_i, y$}
\UnaryInfC{$\vdash (Y \Rightarrow Y)$}
\RightLabel{$\forall_i$}
\UnaryInfC{$\vdash \mathrm{Int}$}
\end{prooftree}
$$

where $Y = (X \Rightarrow X)$.

Proofs matter (5): Cut-elimination of s applied to $\underline{0}$

$$
\begin{prooftree}
\AxiomC{}
\UnaryInfC{$y : Y \vdash Y$}
\AxiomC{}
\UnaryInfC{$x : X \vdash X$}
\RightLabel{$\Rightarrow_i, x$}
\UnaryInfC{$\vdash Y$}
\AxiomC{}
\UnaryInfC{$x : X \vdash X$}
\RightLabel{$\Rightarrow_e$}
\BinaryInfC{$x : X \vdash X$}
\RightLabel{$\Rightarrow_e$}
\BinaryInfC{$y : Y, x : X \vdash X$}
\RightLabel{$\Rightarrow_i, x$}
\UnaryInfC{$y : Y \vdash Y$}
\RightLabel{$\Rightarrow_i, y$}
\UnaryInfC{$\vdash (Y \Rightarrow Y)$}
\RightLabel{$\forall_i$}
\UnaryInfC{$\vdash \mathrm{Int}$}
\end{prooftree}
$$

where $Y = (X \Rightarrow X)$.

Proofs matter (5): Cut-elimination of s applied to $\underline{0}$

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{y : Y \vdash Y} \qquad \overline{x : X \vdash X}}{y : Y, x : X \vdash X} \Rightarrow_e}{y : Y \vdash Y} \Rightarrow_i, x}{\vdash (Y \Rightarrow Y)} \Rightarrow_i, y}{\vdash \mathrm{Int}} \forall_i$$

where $Y = (X \Rightarrow X)$.

This is the proof we called $\underline{1}$.

Proofs matter (6): Cut-elimination of *id* applied to $\underline{0}$

$$\dfrac{\dfrac{\overline{t : \mathsf{Int} \vdash \mathsf{Int}}}{\vdash (\mathsf{Int} \Rightarrow \mathsf{Int})}\ {\Rightarrow_i, t} \qquad \dfrac{\dfrac{\dfrac{\overline{x : X \vdash X}}{\vdash (X \Rightarrow X)}\ {\Rightarrow_i, x}}{\vdash ((X \Rightarrow X) \Rightarrow (X \Rightarrow X))}\ {\Rightarrow_i, y}}{\vdash \mathsf{Int}}\ {\forall_i}}{\vdash \mathsf{Int}}\ {\Rightarrow_e}$$

Proofs matter (6): Cut-elimination of *id* applied to $\underline{0}$

$$
\dfrac{
  \dfrac{\dfrac{}{t : \mathrm{Int} \vdash \mathrm{Int}}}{\vdash (\mathrm{Int} \Rightarrow \mathrm{Int})}\Rightarrow_i, t
  \qquad
  \dfrac{
    \dfrac{
      \dfrac{\dfrac{}{x : X \vdash X}}{\vdash (X \Rightarrow X)}\Rightarrow_i, x
    }{\vdash ((X \Rightarrow X) \Rightarrow (X \Rightarrow X))}\Rightarrow_i, y
  }{\vdash \mathrm{Int}}\forall_i
}{\vdash \mathrm{Int}}\Rightarrow_e
$$

Proofs matter (6): Cut-elimination of *id* applied to $\underline{0}$

$$\dfrac{\dfrac{\dfrac{\overline{x : X \vdash X}}{\vdash (X \Rightarrow X)}\Rightarrow_i, x}{\vdash ((X \Rightarrow X) \Rightarrow (X \Rightarrow X))}\Rightarrow_i, y}{\vdash \mathsf{Int}}\forall_i$$

This is the proof we called $\underline{0}$.

Proofs matter (7): Cut-elimination

We saw that

- $s : (\mathsf{Int} \Rightarrow \mathsf{Int})$ applied to $\underline{0} : \mathsf{Int}$ reduces to $\underline{1} : \mathsf{Int}$.
- $id : (\mathsf{Int} \Rightarrow \mathsf{Int})$ applied to $\underline{0} : \mathsf{Int}$ reduces to the $\underline{0} : \mathsf{Int}$.

Proofs matter (7): Cut-elimination

We saw that

- $s : (\mathrm{Int} \Rightarrow \mathrm{Int})$ applied to $\underline{0} : \mathrm{Int}$ reduces to $\underline{1} : \mathrm{Int}$.
- $id : (\mathrm{Int} \Rightarrow \mathrm{Int})$ applied to $\underline{0} : \mathrm{Int}$ reduces to the $\underline{0} : \mathrm{Int}$.

More generally, we could define $\underline{n} : \mathrm{Int}$ for any $n \in \mathbb{N}$ and show that

- $s : (\mathrm{Int} \Rightarrow \mathrm{Int})$ applied to $\underline{n} : \mathrm{Int}$ reduces to $\underline{n+1} : \mathrm{Int}$.
- $id : (\mathrm{Int} \Rightarrow \mathrm{Int})$ applied to $\underline{n} : \mathrm{Int}$ reduces to $\underline{n} : \mathrm{Int}$.

Proofs matter (7): Cut-elimination

We saw that

- $s : (\mathrm{Int} \Rightarrow \mathrm{Int})$ applied to $\underline{0} : \mathrm{Int}$ reduces to $\underline{1} : \mathrm{Int}$.
- $id : (\mathrm{Int} \Rightarrow \mathrm{Int})$ applied to $\underline{0} : \mathrm{Int}$ reduces to the $\underline{0} : \mathrm{Int}$.

More generally, we could define $\underline{n} : \mathrm{Int}$ for any $n \in \mathbb{N}$ and show that

- $s : (\mathrm{Int} \Rightarrow \mathrm{Int})$ applied to $\underline{n} : \mathrm{Int}$ reduces to $\underline{n+1} : \mathrm{Int}$.
- $id : (\mathrm{Int} \Rightarrow \mathrm{Int})$ applied to $\underline{n} : \mathrm{Int}$ reduces to $\underline{n} : \mathrm{Int}$.

- The proof $s : (\mathrm{Int} \Rightarrow \mathrm{Int})$ behaves like a program that computes the successor of any integer.
- The proof $id : (\mathrm{Int} \Rightarrow \mathrm{Int})$ behaves like a program that returns its argument.

Proofs matter (8): The formulae-as-types correspondence

Cut-elimination always terminates (Girard 1971). Before, it was shown by Gentzen (1934) that cut-elimination terminates in (classical) propositional logic.

Proofs matter (8): The formulae-as-types correspondence

Cut-elimination always terminates (Girard 1971). Before, it was shown by Gentzen (1934) that cut-elimination terminates in (classical) propositional logic. Nevertheless, notice that:

- Cut-elimination terminates in natural deduction for intuitionistic propositional logic and is confluent.
- Cut-elimination terminates in sequent calculus for intuitionistic propositional logic (LJ), but is not confluent.

Proofs matter (8): The formulae-as-types correspondence

Cut-elimination always terminates (Girard 1971). Before, it was shown by Gentzen (1934) that cut-elimination terminates in (classical) propositional logic. Nevertheless, notice that:

- Cut-elimination terminates in natural deduction for intuitionistic propositional logic and is confluent.
- Cut-elimination terminates in sequent calculus for intuitionistic propositional logic (LJ), but is not confluent.

Cut-elimination in LJ behaves bad because equality between LJ proofs is "evil" (i.e. too fine).

Proofs matter (8): The formulae-as-types correspondence

Cut-elimination always terminates (Girard 1971). Before, it was shown by Gentzen (1934) that cut-elimination terminates in (classical) propositional logic. Nevertheless, notice that:

- Cut-elimination terminates in natural deduction for intuitionistic propositional logic and is confluent.
- Cut-elimination terminates in sequent calculus for intuitionistic propositional logic (LJ), but is not confluent.

Cut-elimination in LJ behaves bad because equality between LJ proofs is "evil" (i.e. too fine).

Conclusion: Proofs in (intuitionistic) natural deduction are programs, where formulae are types and cut-elimination is their execution.

Untyped λ-calculus

Proofs in intuitionistic natural deduction can be represented by typed λ-terms.

Untyped λ-calculus

Proofs in intuitionistic natural deduction can be represented by typed λ-terms.

But what about *untyped* λ-calculus? It was introduced in 1936 (Church) without any denotational semantics.

Untyped λ-calculus

Proofs in intuitionistic natural deduction can be represented by typed λ-terms.

But what about *untyped* λ-calculus? It was introduced in 1936 (Church) without any denotational semantics.
Simply typed λ-calculus has a standard denotational semantics: types are sets and terms are functions.
But in *untyped* λ-calculus terms can be applied to themselves and the set of functions $D \to D$ cannot be embedded into D (unless $D \simeq \{*\}$).

Untyped λ-calculus

Proofs in intuitionistic natural deduction can be represented by typed λ-terms.

But what about *untyped* λ-calculus? It was introduced in 1936 (Church) without any denotational semantics.
Simply typed λ-calculus has a standard denotational semantics: types are sets and terms are functions.
But in *untyped* λ-calculus terms can be applied to themselves and the set of functions $D \to D$ cannot be embedded into D (unless $D \simeq \{*\}$).

Scott (1972): Building a *topological space* D that is homeomorphic to the space of *continuous functions* $D \to D$.

Kolmogorov spaces

A Kolmogorov space X is a topological space where distinct points are topologically distinguishable:

- $(\forall x, y \in X)(\mathcal{N}_X(x) = \mathcal{N}_X(y) \Rightarrow x = y)$
- i.e. $(\forall x, y \in X)(\overline{\{x\}} = \overline{\{y\}} \Rightarrow x = y)$

Kolmogorov spaces

A Kolmogorov space X is a topological space where distinct points are topologically distinguishable:

- $(\forall x, y \in X)(\mathcal{N}_X(x) = \mathcal{N}_X(y) \Rightarrow x = y)$
- i.e. $(\forall x, y \in X)(\overline{\{x\}} = \overline{\{y\}} \Rightarrow x = y)$

We have a faifthul functor from the category of Kolmogorov spaces to the category of posets by the *specialisation functor*:

$$(x \leq_X y \Leftrightarrow \mathcal{N}_X(x) \subseteq \mathcal{N}_X(y))$$

$$i.e.(x \leq_X y \Leftrightarrow \overline{\{x\}} \subseteq \overline{\{y\}})$$

Kolmogorov spaces

A Kolmogorov space X is a topological space where distinct points are topologically distinguishable:

- $(\forall x, y \in X)(\mathcal{N}_X(x) = \mathcal{N}_X(y) \Rightarrow x = y)$
- i.e. $(\forall x, y \in X)(\overline{\{x\}} = \overline{\{y\}} \Rightarrow x = y)$

We have a faifthul functor from the category of Kolmogorov spaces to the category of posets by the *specialisation functor*:

$$(x \leq_X y \Leftrightarrow \mathcal{N}_X(x) \subseteq \mathcal{N}_X(y))$$

$$i.e.(x \leq_X y \Leftrightarrow \overline{\{x\}} \subseteq \overline{\{y\}})$$

Given a poset $(E, \leq)$, the set E can be endowed with several topologies Ω such that

- (E, Ω) is a Kolmogorov space
- and the specialisation order on (E, Ω) is $(E, \leq)$.

Kolmogorov spaces

A Kolmogorov space X is a topological space where distinct points are topologically distinguishable:

- $(\forall x, y \in X)(\mathcal{N}_X(x) = \mathcal{N}_X(y) \Rightarrow x = y)$
- i.e. $(\forall x, y \in X)(\overline{\{x\}} = \overline{\{y\}} \Rightarrow x = y)$

We have a faifthul functor from the category of Kolmogorov spaces to the category of posets by the *specialisation functor*:

$$(x \leq_X y \Leftrightarrow \mathcal{N}_X(x) \subseteq \mathcal{N}_X(y))$$

$$i.e. (x \leq_X y \Leftrightarrow \overline{\{x\}} \subseteq \overline{\{y\}})$$

Given a poset $(E, \leq)$, the set E can be endowed with several topologies Ω such that

- (E, Ω) is a Kolmogorov space
- and the specialisation order on (E, Ω) is $(E, \leq)$.

One of these topologies is the *Scott topology*.

Scott topology

Scott opens of $(E, \leq)$ are upper sets that are inaccessible by directed joins, i.e. subsets U of E such that

- $(\forall x \in U)(\forall y \in E)(x \leq y \Rightarrow y \in U)$
- $(\forall \Delta \subseteq_{\mathrm{dir}} E)(\bigvee \Delta \in U \Rightarrow \Delta \cap U \neq \emptyset)$

Scott topology

Scott opens of $(E, \leq)$ are upper sets that are inaccessible by directed joins, i.e. subsets U of E such that

- $(\forall x \in U)(\forall y \in E)(x \leq y \Rightarrow y \in U)$
- $(\forall \Delta \subseteq_{\mathrm{dir}} E)(\bigvee \Delta \in U \Rightarrow \Delta \cap U \neq \emptyset)$

Theorem.

(i) Scott opens of $(E, \leq)$ form a topology Ω on E.

(ii) The specialisation order on this topology is the order $\leq$.

Proof. (i) is trivial.
For (ii), notice $(\forall x \in E)\{y \in E; \neg y \leq x\} \in \Omega$.

Scott topology

Scott opens of $(E, \leq)$ are upper sets that are inaccessible by directed joins, i.e. subsets U of E such that

- $(\forall x \in U)(\forall y \in E)(x \leq y \Rightarrow y \in U)$
- $(\forall \Delta \subseteq_{\mathrm{dir}} E)(\bigvee \Delta \in U \Rightarrow \Delta \cap U \neq \emptyset)$

Theorem.

(i) Scott opens of $(E, \leq)$ form a topology Ω on E.

(ii) The specialisation order on this topology is the order $\leq$.

Proof. (i) is trivial.
For (ii), notice $(\forall x \in E)\{y \in E; \neg y \leq x\} \in \Omega$.
Example. $B = \{\mathrm{T}, \mathrm{F}, \bot\}$, $\mathrm{B} = (B, \leq_{\mathrm{B}})$, where $(x \leq_{\mathrm{B}} y \Leftrightarrow (x = y \vee x = \bot))$, and $\mathbb{B} = (B, \Omega)$ with $\Omega = \mathfrak{P}(\{\mathrm{T}, \mathrm{F}, \bot\}) \setminus \{\{\bot\}\}$ the Scott topology of B.
If $f : \{\mathrm{T}, \mathrm{F}, \bot\} \to \{\mathrm{T}, \mathrm{F}, \bot\}$ s.t. $f(\bot) \neq \bot$ and $f(\mathrm{T}) = \bot$, then f is not a continuous function $\mathbb{B} \to \mathbb{B}$.
Intuition: If $f(\bot) \neq \bot$, then f denotes a program that does not read its argument and thus should be constant.

A model of untyped λ-calculus

Scott (1972) has been able to build a special lattice D endowed with the Scott topology that has the property

$$(D \Rightarrow D) \simeq D$$

where $(D \Rightarrow D)$ is the space of continuous functions $D \to D$.

Stability

Let $\mathbb{B} \times \mathbb{B}$ be the Scott topology on the product order $\mathsf{B} \times \mathsf{B}$. We have a continuous function $p : \mathbb{B} \times \mathbb{B} \to \mathbb{B}$ as follows:

- $(p(x, y) = \mathsf{T} \Leftrightarrow \mathsf{T} \in \{x, y\})$
- $(p(x, y) = \mathsf{F} \Leftrightarrow \{x, y\} = \{\mathsf{F}\})$

But there is no *sequential* program denoted by p.

Stability

Let $\mathbb{B} \times \mathbb{B}$ be the Scott topology on the product order $\mathrm{B} \times \mathrm{B}$. We have a continuous function $p : \mathbb{B} \times \mathbb{B} \to \mathbb{B}$ as follows:

- $(p(x, y) = \mathrm{T} \Leftrightarrow \mathrm{T} \in \{x, y\})$
- $(p(x, y) = \mathrm{F} \Leftrightarrow \{x, y\} = \{\mathrm{F}\})$

But there is no *sequential* program denoted by p.

Berry (1978) introduced *dI-domains* (which are posets with some "good" properties) and *stable* functions between them, which are continuous functions with some "good" property, as a model of PCF, which is a sequential programming language based on λ-calculus.

Stability

Let $\mathbb{B} \times \mathbb{B}$ be the Scott topology on the product order $\mathrm{B} \times \mathrm{B}$. We have a continuous function $p : \mathbb{B} \times \mathbb{B} \to \mathbb{B}$ as follows:

- $(p(x, y) = \mathrm{T} \Leftrightarrow \mathrm{T} \in \{x, y\})$
- $(p(x, y) = \mathrm{F} \Leftrightarrow \{x, y\} = \{\mathrm{F}\})$

But there is no *sequential* program denoted by p.

Berry (1978) introduced *dI-domains* (which are posets with some "good" properties) and *stable* functions between them, which are continuous functions with some "good" property, as a model of PCF, which is a sequential programming language based on λ-calculus.

$\mathbb{B} \times \mathbb{B}$ and $\mathbb{B}$ are dI-domains. The function p is *not* stable, because

- $(\mathrm{T}, \perp) \vee (\perp, \mathrm{T})$ exists
- and $p(\mathrm{T}, \perp) \wedge p(\perp, \mathrm{T}) = \mathrm{T} \neq \perp = p((\mathrm{T}, \perp) \wedge (\perp, \mathrm{T}))$

Coherence spaces

Girard (1986) introduced *coherence spaces*, which are special dI-domains, and showed that coherence spaces with stable functions are a model of second-order intuitionistisc logic.

A coherence space $(A, \overset{\frown}{\smile})$ is a set A endowed with a symmetric reflexive relation $\overset{\frown}{\smile}$ (a *coherence* relation) on A. Thet set $\mathcal{C}(A, \overset{\frown}{\smile})$ of its *cliques* (i.e. complete subgraphs) endowed with the inclusion is a dI-domain (and, in particular, a poset).

Coherence spaces

Girard (1986) introduced *coherence spaces*, which are special dI-domains, and showed that coherence spaces with stable functions are a model of second-order intuitionistisc logic.

A coherence space $(A, \frown)$ is a set A endowed with a symmetric reflexive relation $\frown$ (a *coherence* relation) on A. Thet set $\mathcal{C}(A, \frown)$ of its *cliques* (i.e. complete subgraphs) endowed with the inclusion is a dI-domain (and, in particular, a poset).

Example. The binary relation $\frown_{\mathbb{B}}$ defined on $\{\mathrm{t}, \mathrm{f}\}$ by $x \frown_{\mathbb{B}} y$ iff $x = y$ is reflexive and symmetric. The cliques of $(\{\mathrm{t}, \mathrm{f}\}, \frown_{\mathbb{B}})$ are:

- $\perp = \emptyset$
- $\mathsf{T} = \{\mathrm{t}\}$
- $\mathsf{F} = \{\mathrm{f}\}$

We recover the poset $\mathsf{B} = (B, \leq_{\mathsf{B}})$ by taking $\leq_{\mathsf{B}} = \subseteq$.

Stable functions between coherence spaces

Proposition. Given two coherence spaces $\mathcal{A}$ and $\mathcal{B}$, a continuous function $(\mathcal{C}(\mathcal{A}), \Omega_{\mathcal{A}}) \to (\mathcal{C}(\mathcal{B}), \Omega_{\mathcal{B}})$, where $\Omega_{\mathcal{A}}$ is the Scott topology of $(\mathcal{C}(\mathcal{A}), \subseteq)$, is a function $f\colon \mathcal{C}(\mathcal{A}) \to \mathcal{C}(\mathcal{B})$ such that

- $(a' \subseteq a \Rightarrow f(a') \subseteq f(a))$
- and, if Δ is a directed subset of $(\mathcal{C}(\mathcal{A}), \subseteq)$, then $f(\bigcup \Delta) = \bigcup f[\Delta]$

Stable functions between coherence spaces

Proposition. Given two coherence spaces $\mathcal{A}$ and $\mathcal{B}$, a continuous function $(\mathcal{C}(\mathcal{A}), \Omega_{\mathcal{A}}) \to (\mathcal{C}(\mathcal{B}), \Omega_{\mathcal{B}})$, where $\Omega_{\mathcal{A}}$ is the Scott topology of $(\mathcal{C}(\mathcal{A}), \subseteq)$, is a function $f\colon \mathcal{C}(\mathcal{A}) \to \mathcal{C}(\mathcal{B})$ such that

- $(a' \subseteq a \Rightarrow f(a') \subseteq f(a))$
- and, if Δ is a directed subset of $(\mathcal{C}(\mathcal{A}), \subseteq)$, then $f(\bigcup \Delta) = \bigcup f[\Delta]$

Definition. A *stable* function $\mathcal{A} \to \mathcal{B}$ is a continuous function $f\colon (\mathcal{C}(\mathcal{A}), \Omega_{\mathcal{A}}) \to (\mathcal{C}(\mathcal{B}), \Omega_{\mathcal{B}})$ such that

$$(\forall a, a' \in \mathcal{C}(\mathcal{A}))(a \cup a' \in \mathcal{C}(\mathcal{A}) \Rightarrow f(a \cap a') = f(a) \cap f(a'))$$

Product of coherence spaces

Given two coherence spaces $\mathcal{A}_1 = (A_1, \frown_1)$ and $\mathcal{A}_2 = (A_2, \frown_2)$, the product $\mathcal{A}_1 \& \mathcal{A}_2$ is $((\{1\} \times A_1) \cup (\{2\} \times A_2), \frown)$, where

$$((i, a) \frown (j, b) \Leftrightarrow (i = j \Rightarrow a \frown_i b))$$

Product of coherence spaces

Given two coherence spaces $\mathcal{A}_1 = (A_1, \coh_1)$ and $\mathcal{A}_2 = (A_2, \coh_2)$, the product $\mathcal{A}_1 \& \mathcal{A}_2$ is $((\{1\} \times A_1) \cup (\{2\} \times A_2), \coh)$, where

$$((i, a) \coh (j, b) \Leftrightarrow (i = j \Rightarrow a \coh_i b))$$

Example. The cliques of $(\{\mathrm{t}, \mathrm{f}\}, \coh_{\mathbb{B}}) \& (\{\mathrm{t}, \mathrm{f}\}, \coh_{\mathbb{B}})$ are:

- $(\bot, \bot) = \emptyset$
- $(\mathrm{T}, \bot) = \{(1, \mathrm{t})\}$
- $(\bot, \mathrm{T}) = \{(2, \mathrm{t})\}$
- $(\mathrm{F}, \bot) = \{(1, \mathrm{f})\}$
- $(\bot, \mathrm{F}) = \{(2, \mathrm{f})\}$
- $(\mathrm{T}, \mathrm{T}) = \{(1, \mathrm{t}), (2, \mathrm{t})\}$
- $(\mathrm{T}, \mathrm{F}) = \{(1, \mathrm{t}), (2, \mathrm{f})\}$
- $(\mathrm{F}, \mathrm{T}) = \{(1, \mathrm{f}), (2, \mathrm{t})\}$
- $(\mathrm{F}, \mathrm{F}) = \{(1, \mathrm{f}), (2, \mathrm{f})\}$

The function p is *not* a stable function

$$(\{\mathrm{t}, \mathrm{f}\}, \coh_{\mathbb{B}}) \& (\{\mathrm{t}, \mathrm{f}\}, \coh_{\mathbb{B}}) \to (\{\mathrm{t}, \mathrm{f}\}, \coh_{\mathbb{B}})$$

The coherence space of stable functions

Proposition. Let $f\colon \mathcal{A} \to \mathcal{B}$ be a stable function. Then, for any $a \in \mathcal{C}(\mathcal{A})$, for any $\beta \in f(a)$, there exists $a_0 \subseteq_{\text{fin}} a$ such that

- $\beta \in f(a_0)$
- and $(\forall a' \subseteq a_o)(\beta \in f(a') \Rightarrow a' = a_0)$.

The coherence space of stable functions

Proposition. Let $f\colon \mathcal{A} \to \mathcal{B}$ be a stable function. Then, for any $a \in \mathcal{C}(\mathcal{A})$, for any $\beta \in f(a)$, there exists $a_0 \subseteq_{\mathsf{fin}} a$ such that

- $\beta \in f(a_0)$
- and $(\forall a' \subseteq a_o)(\beta \in f(a') \Rightarrow a' = a_0)$.

One can then endow the set $\mathcal{C}_{\mathsf{fin}}(\mathcal{A}) \times B$ with a coherence relation $\frown$ to define the space $\mathcal{A} \Rightarrow \mathcal{B}$. We thus get a cartesian closed category (i.e. a model of the simply typed λ-calculus). What is striking is that this construction can be made up of two constructions:

- Given a coherence space $\mathcal{A}$, one can get a new coherence space $!\mathcal{A}$ on the set $\mathcal{C}_{\mathsf{fin}}(\mathcal{A})$.
- Given two coherence spaces $\mathcal{A} = (A, \frown_{\mathcal{A}})$ and $\mathcal{B} = (B, \frown_{\mathcal{B}})$, one can get a new coherence space $\mathcal{A} \multimap \mathcal{B}$ on the set $A \times B$.

The decomposition of the intuitionnistic arrow $A \Rightarrow B$ into $!A \multimap B$ gave rise to the discovery of linear logic (LL).

Linear logic

The linear implication $A \multimap B$ can itself be decomposed into $A^\perp \parr B$ (like in *classical* logic!) with an involutive linear negation. The negation corresponds to reversing the coherence relation $\frown$ and the two implications ($\Rightarrow$ and $\multimap$) to two closed categories:

- The category Stab of stable functions between coherence spaces: A model of intuitionistic logic.
- And the category Lin of linear functions between coherence spaces: A model of linear logic.

$$\mathrm{Lin}(!(A, \mathcal{A}), (B, \mathcal{B})) \simeq \mathrm{Stab}((A, \mathcal{A}), (B, \mathcal{B}))$$

Linear logic

The linear implication $A \multimap B$ can itself be decomposed into $A^{\perp} \parr B$ (like in *classical* logic!) with an involutive linear negation. The negation corresponds to reversing the coherence relation $\frown$ and the two implications ($\Rightarrow$ and $\multimap$) to two closed categories:

- The category Stab of stable functions between coherence spaces: A model of intuitionistic logic.
- And the category Lin of linear functions between coherence spaces: A model of linear logic.

$$\mathrm{Lin}(!(A,\mathcal{A}),(B,\mathcal{B})) \simeq \mathrm{Stab}((A,\mathcal{A}),(B,\mathcal{B}))$$

Grammar of the formulae of propositional linear logic:

$$\mathbb{T} ::= X \mid X^{\perp} \mid 1 \mid \perp \mid (\mathbb{T} \otimes \mathbb{T}) \mid (\mathbb{T} \parr \mathbb{T}) \mid !\mathbb{T} \mid ?\mathbb{T} \mid (\mathbb{T} \& \mathbb{T}) \mid (\mathbb{T} \oplus \mathbb{T}) \mid 0 \mid \top$$

with the de Morgan laws:

- $(A \otimes B)^{\perp} = (A^{\perp} \parr B^{\perp})$ and $(A \parr B)^{\perp} = (A^{\perp} \otimes B^{\perp})$
- $(!A)^{\perp} = ?A^{\perp}$ and $(?A)^{\perp} = !A^{\perp}$
- $(A \& B)^{\perp} = (A^{\perp} \oplus B^{\perp})$ and $(A \oplus B)^{\perp} = (A^{\perp} \& B^{\perp})$
- $(X^{\perp})^{\perp} = X$, $1^{\perp} = \perp$, $\perp^{\perp} = 1$, $0^{\perp} = \top$ and $\top^{\perp} = 0$

The problem of canonicity of proofs

Intuitionistic sequent calculus (LJ)	Natural deduction
MELL sequent calculus	Girard proof-nets?

The problem of canonicity of proofs

Danos-Regnier proof-nets (1995) are an improvement of Girard proof-nets.

Intuitionistic sequent calculus (LJ)	Natural deduction
MELL sequent calculus	Danos-Regnier proof-nets?

Sequent calculus proofs

$$
\dfrac{\dfrac{\dfrac{\vdash A, A^{\perp} \qquad \vdash B, B^{\perp}}{\vdash (A \otimes B), A^{\perp}, B^{\perp}}\,\otimes}{\vdash (A \otimes B), (A^{\perp} \parr B^{\perp})}\,\parr \qquad \vdash \underline{A}, \underline{A}^{\perp}}{\vdash ((A \otimes B) \otimes \underline{A}), (A^{\perp} \parr B^{\perp}), \underline{A}^{\perp}}\,\otimes
$$

Figure: Proof π_1

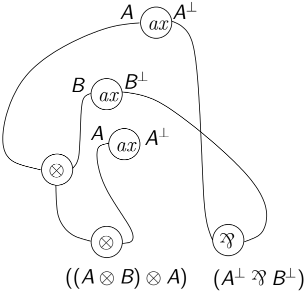
$$
\dfrac{\dfrac{\dfrac{\vdash A, A^{\perp} \qquad \vdash B, B^{\perp}}{\vdash (A \otimes B), A^{\perp}, B^{\perp}}\,\otimes \qquad \vdash \underline{A}, \underline{A}^{\perp}}{\vdash ((A \otimes B) \otimes \underline{A}), A^{\perp}, B^{\perp}, \underline{A}^{\perp}}\,\otimes}{\vdash ((A \otimes B) \otimes \underline{A}), (A^{\perp} \parr B^{\perp}), \underline{A}^{\perp}}\,\parr
$$

Figure: Proof π_2

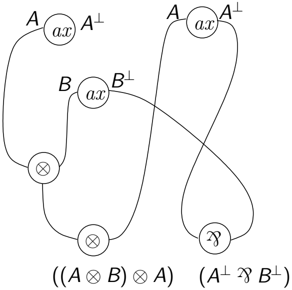
$$
\dfrac{\dfrac{\dfrac{\vdash A, A^{\perp} \qquad \vdash B, B^{\perp}}{\vdash (A \otimes B), A^{\perp}, B^{\perp}}\,\otimes \qquad \vdash \underline{A}, \underline{A}^{\perp}}{\vdash ((A \otimes B) \otimes \underline{A}), A^{\perp}, B^{\perp}, \underline{A}^{\perp}}\,\otimes}{\vdash ((A \otimes B) \otimes \underline{A}), (\underline{A}^{\perp} \parr B^{\perp}), A^{\perp}}\,\parr
$$

Figure: Proof π_3

Proof-nets



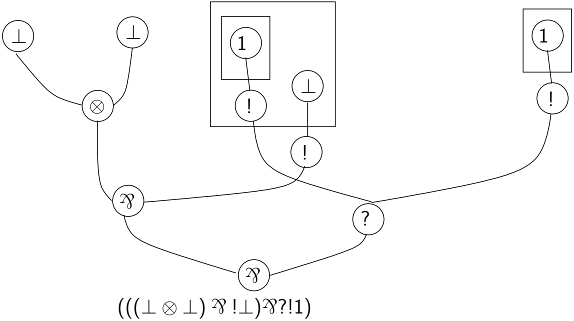
A ax $A^\perp$
B ax $B^\perp$
A ax $A^\perp$
$\otimes$
$\otimes$
$⅋$
$((A \otimes B) \otimes A)$
$(A^\perp ⅋ B^\perp)$



A ax $A^\perp$
A ax $A^\perp$
B ax $B^\perp$
$\otimes$
$\otimes$
$⅋$
$((A \otimes B) \otimes A)$
$(A^\perp ⅋ B^\perp)$

A proof-net with boxes

A proof of $(((\perp \otimes \perp) \parr\, !\perp) \parr ?!1)$:



Finiteness spaces

On a set A, one defines the binary relation $\perp$ on $\mathfrak{P}(A)$ by $a \perp b$ iff $\mathrm{Card}\,(a \cap b) \leq 1$. For any $\mathcal{A} \in \mathfrak{P}(\mathfrak{P}(A))$, we set $\mathcal{A}^{\perp} = \{b \subseteq A; (\forall a \in \mathcal{A}) a \perp b\}$.
A coherence space $(A, \frown)$ is a set $\mathcal{A} \in \mathfrak{P}(\mathfrak{P}(A))$ such that $\mathcal{A} = \mathcal{A}^{\perp\perp}$ (with $\mathcal{C}(A, \frown) = \mathcal{A}$).

Finiteness spaces

On a set A, one defines the binary relation $\perp$ on $\mathfrak{P}(A)$ by $a \perp b$ iff $\mathrm{Card}\,(a \cap b) \leq 1$. For any $\mathcal{A} \in \mathfrak{P}(\mathfrak{P}(A))$, we set $\mathcal{A}^\perp = \{b \subseteq A; (\forall a \in \mathcal{A}) a \perp b\}$.
A coherence space $(A, \frown)$ is a set $\mathcal{A} \in \mathfrak{P}(\mathfrak{P}(A))$ such that $\mathcal{A} = \mathcal{A}^{\perp\perp}$ (with $\mathcal{C}(A, \frown) = \mathcal{A}$).
Definition. (Ehrhard 2005) Taking instead for $\perp$ the relation defined by $a \perp b$ iff $a \cap b$ is finite, one gets *finiteness spaces* $(A, \mathcal{A})$ instead of coherence spaces, which provide a new model of LL.
Examples.

- If A is finite, then $(A, \mathcal{A})$ is a finiteness space iff $\mathcal{A} = \mathfrak{P}(\mathcal{A})$.
- $\mathsf{N} = (\mathbb{N}, \mathfrak{P}_{\mathsf{fin}}(\mathbb{N}))$
- $\mathsf{N}^\perp = (\mathbb{N}, \mathfrak{P}(\mathbb{N}))$
- $!\mathsf{N} = (\mathfrak{M}_{\mathsf{fin}}(\mathbb{N}), \{a \subseteq \mathfrak{M}_{\mathsf{fin}}(\mathbb{N}); (\exists u \in \mathfrak{P}_{\mathsf{fin}}(\mathsf{N}))(\forall \mu \in a)\mathrm{Supp}(\mu) \subseteq u\})$
- $(!N)^\perp = (\mathfrak{M}_{\mathsf{fin}}(\mathbb{N}), \{a \subseteq \mathfrak{M}_{\mathsf{fin}}(\mathbb{N}); (\forall u \in \mathfrak{P}_{\mathsf{fin}}(\mathsf{N}))\#\{\mu \in a; \mathrm{Supp}(\mu) \subseteq u\} < \infty\})$

Topological modules associated with finiteness spaces

Given a commutative (semi-)field R endowed with the discrete topology, each finiteness space $(A, \mathcal{A})$ gives rise to a topological R-module $R\langle\mathcal{A}\rangle$: vectors are the $v \in k^A$ s.t. $\mathrm{Supp}(v) = \{\alpha \in a; v(\alpha) \neq 0\} \in \mathcal{A}$ and the topology is the Lefschetz topology (1942).

For any vector v, we have $v = \sum_{\alpha \in A} v(\alpha) \cdot \alpha$.

Topological modules associated with finiteness spaces

Given a commutative (semi-)field R endowed with the discrete topology, each finiteness space $(A, \mathcal{A})$ gives rise to a topological R-module $R\langle\mathcal{A}\rangle$: vectors are the $v \in k^A$ s.t. $\mathrm{Supp}(v) = \{\alpha \in a; v(\alpha) \neq 0\} \in \mathcal{A}$ and the topology is the Lefschetz topology (1942).
For any vector v, we have $v = \sum_{\alpha \in A} v(\alpha) \cdot \alpha$.

Example. Let 2 be the semi-ring $\{0, 1\}$ with $1 + 1 = 1$.
We have the following continuous linear function

$$succ : 2\langle !\mathrm{N}\rangle \to 2\langle \mathrm{N}\rangle$$

For any $u \in \mathfrak{P}_{\mathrm{fin}}(\mathbb{N})$, for any $(\lambda_\mu)_{\mu \in \mathfrak{M}_{\mathrm{fin}}(u)} \in \{0, 1\}^{\mathfrak{M}_{\mathrm{fin}}(u)}$, we have

$$succ\Big(\sum_{\mu \in \mathfrak{M}_{\mathrm{fin}}(u)} \lambda_\mu \cdot \mu\Big) = \sum_{\substack{n \in \mathbb{N} \\ \lambda_{[n]} = 1}} 1 \cdot [n + 1]$$

It denotes a program that computes the successor function and reads exactly once its argument.

Non-uniformity

Consider the following program:

$\lambda x.$if x then True
else if x then True else False

Non-uniformity

Consider the following program:

$\lambda x.\text{if } x$ then True
else if x then True else False

It can be seen as a continuous linear function $g : 2\langle !\mathsf{B}\rangle \to 2\langle \mathsf{B}\rangle$, where B is the finiteness space $(\{\mathsf{T},\mathsf{F}\}, \mathfrak{P}(\{\mathsf{T},\mathsf{F}\}))$: We have

- $g([\mathsf{T}]) = \mathsf{T}$
- $g([\mathsf{F}]) = 0$
- $g([\mathsf{T},\mathsf{T}]) = 0$
- $g([\mathsf{F},\mathsf{F}]) = \mathsf{F}$
- $g([\mathsf{T}] + [\mathsf{F},\mathsf{F}]) = \mathsf{T} + \mathsf{F}$
- $g([\mathsf{F},\mathsf{T}]) = \mathsf{T}$: non-uniformity of the semantics
- etc...

The Kleisli category

Let us recall the situation with coherence spaces:

$$\mathsf{Lin}(!(A,\mathcal{A}),(B,\mathcal{B})) \simeq \mathsf{Stab}((A,\mathcal{A}),(B,\mathcal{B}))$$

The category Stab is the Kleisli category of the comonad !.

Same situation with finiteness spaces:

The Kleisli category

Let us recall the situation with coherence spaces:

$$\mathsf{Lin}(!(A,\mathcal{A}),(B,\mathcal{B})) \simeq \mathsf{Stab}((A,\mathcal{A}),(B,\mathcal{B}))$$

The category Stab is the Kleisli category of the comonad !.

Same situation with finiteness spaces:
A continuous linear function $f\colon R\langle !(A,\mathcal{A})\rangle \to R\langle (B,\mathcal{B})\rangle$ can be seen as a power series $\underline{f}$ from $R\langle (A,\mathcal{A})\rangle$ to $R\langle (B,\mathcal{B})\rangle$ such that $\underline{f}(0) = f([])$.

Derivatives

Given a continuous linear function $f\colon R\langle !(A,\mathcal{A})\rangle \to R\langle (B,\mathcal{B})\rangle$, the derivative at 0 of $\underline{f}$ is $\underline{f}'(0) = f \circ \mathrm{cod} : R\langle (A,\mathcal{A})\rangle \to R\langle (B,\mathcal{B})\rangle$, where $\mathrm{cod} : R\langle (A,\mathcal{A})\rangle \to R\langle !(A,\mathcal{A})\rangle$ is defined by

$$(\forall v \in R\langle (A,\mathcal{A})\rangle)\mathrm{cod}(v) = \sum_{\alpha \in \mathrm{Supp}(v)} v(\alpha) \cdot [\alpha]$$

Derivatives

Given a continuous linear function $f\colon R\langle !(A,\mathcal{A})\rangle \to R\langle (B,\mathcal{B})\rangle$, the derivative at 0 of $\underline{f}$ is $\underline{f}'(0) = f \circ \mathsf{cod} : R\langle (A,\mathcal{A})\rangle \to R\langle (B,\mathcal{B})\rangle$, where $\mathsf{cod} : R\langle (A,\mathcal{A})\rangle \to R\langle !(A,\mathcal{A})\rangle$ is defined by

$$(\forall v \in R\langle (A,\mathcal{A})\rangle)\mathsf{cod}(v) = \sum_{\alpha \in \mathsf{Supp}(v)} v(\alpha)\cdot[\alpha]$$

Examples.

- The derivative at 0 of $\underline{succ}$ is $\underline{succ}'(0) : 2\langle \mathsf{N}\rangle \to 2\langle \mathsf{N}\rangle$ defined by

$$\underline{succ}'(0)(\sum_{n\in\mathbb{N}} \lambda_n \cdot n) = \sum_{n\in\mathbb{N}} \lambda_n \cdot (n+1)$$

Derivatives

Given a continuous linear function $f\colon R\langle !(A,\mathcal{A})\rangle \to R\langle (B,\mathcal{B})\rangle$, the derivative at 0 of $\underline{f}$ is $\underline{f}'(0) = f \circ \mathrm{cod} : R\langle (A,\mathcal{A})\rangle \to R\langle (B,\mathcal{B})\rangle$, where $\mathrm{cod} : R\langle (A,\mathcal{A})\rangle \to R\langle !(A,\mathcal{A})\rangle$ is defined by

$$(\forall v \in R\langle (A,\mathcal{A})\rangle)\mathrm{cod}(v) = \sum_{\alpha \in \mathrm{Supp}(v)} v(\alpha) \cdot [\alpha]$$

Examples.

- The derivative at 0 of $\underline{succ}$ is $\underline{succ}'(0) : 2\langle \mathrm{N}\rangle \to 2\langle \mathrm{N}\rangle$ defined by

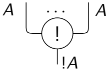
$$\underline{succ}'(0)(\sum_{n\in\mathbb{N}} \lambda_n \cdot n) = \sum_{n\in\mathbb{N}} \lambda_n \cdot (n+1)$$

- The derivative at 0 of $\underline{g}$ is $\underline{g}'(0) : 2\langle B\rangle \to 2\langle B\rangle$ defined by

$$\underline{g}'(0)(\lambda_1 \cdot \mathrm{T} + \lambda_2 \cdot \mathrm{F})) = \lambda_1 \cdot \mathrm{T}$$

Differential nets

Differential nets (Ehrhard-Regnier 2006) allow to express the Taylor expansion of any linear logic proof in the syntax.
We have no box any more but a new kind of cells (cocontractions):



(with 0 premises, we get coderelictions)
and we have sums of nets (which express non-determinism). For Taylor expansion, we need *infinite* sums: infinite sums are not strictly speaking syntactical objets but lie in between syntax (we have cut-elimination) and semantics (we have infinite objects).

Derivatives of constant functions semantically

The continuous linear function $w_{(A,\mathcal{A})} : R\langle !(A,\mathcal{A})\rangle \to R$ is defined by $(\forall v \in R\langle !(A,\mathcal{A})\rangle) w_{(A,\mathcal{A})}(v) = v([])$.
If $f : R \to R\langle (B,\mathcal{B})\rangle$, then we have the continuous linear function $f \circ w_{!(A,\mathcal{A})} : R\langle !(A,\mathcal{A})\rangle \to R\langle (B,\mathcal{B})\rangle$ with

$$(\forall v \in R\langle !(A,\mathcal{A})\rangle)(f \circ w_{(A,\mathcal{A})})(v) = v([]) \cdot f(1)$$

which corresponds to the constant power series $\underline{f \circ w_{!(A,\mathcal{A})}}$ from $R\langle (A,\mathcal{A})\rangle$ to $R\langle (B,\mathcal{B})\rangle$ with

$$(\forall v \in R\langle (A,\mathcal{A})\rangle)\underline{(f \circ w_{(A,\mathcal{A})})}(v) = f(1)$$

Derivatives of constant functions semantically

The continuous linear function $w_{(A,\mathcal{A})} : R\langle !(A,\mathcal{A})\rangle \to R$ is defined by $(\forall v \in R\langle !(A,\mathcal{A})\rangle) w_{(A,\mathcal{A})}(v) = v([])$.
If $f \colon R \to R\langle (B,\mathcal{B})\rangle$, then we have the continuous linear function $f \circ w_{!(A,\mathcal{A})} : R\langle !(A,\mathcal{A})\rangle \to R\langle (B,\mathcal{B})\rangle$ with

$$(\forall v \in R\langle !(A,\mathcal{A})\rangle)(f \circ w_{(A,\mathcal{A})})(v) = v([]) \cdot f(1)$$

which corresponds to the constant power series $\underline{f \circ w_{!(A,\mathcal{A})}}$ from $R\langle (A,\mathcal{A})\rangle$ to $R\langle (B,\mathcal{B})\rangle$ with

$$(\forall v \in R\langle (A,\mathcal{A})\rangle)\underline{(f \circ w_{(A,\mathcal{A})})}(v) = f(1)$$

The derivative at 0 of a constant function should be the zero function. Let us check:

Derivatives of constant functions semantically

The continuous linear function $w_{(A,\mathcal{A})} : R\langle !(A,\mathcal{A})\rangle \to R$ is defined by $(\forall v \in R\langle !(A,\mathcal{A})\rangle) w_{(A,\mathcal{A})}(v) = v([])$.
If $f : R \to R\langle (B,\mathcal{B})\rangle$, then we have the continuous linear function $f \circ w_{!(A,\mathcal{A})} : R\langle !(A,\mathcal{A})\rangle \to R\langle (B,\mathcal{B})\rangle$ with

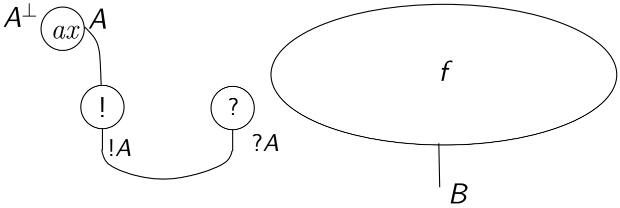
$$(\forall v \in R\langle !(A,\mathcal{A})\rangle)(f \circ w_{(A,\mathcal{A})})(v) = v([]) \cdot f(1)$$

which corresponds to the constant power series $\underline{f \circ w_{!(A,\mathcal{A})}}$ from $R\langle (A,\mathcal{A})\rangle$ to $R\langle (B,\mathcal{B})\rangle$ with

$$(\forall v \in R\langle (A,\mathcal{A})\rangle)\underline{(f \circ w_{(A,\mathcal{A})})}(v) = f(1)$$

The derivative at 0 of a constant function should be the zero function. Let us check:For any $v \in R\langle (A,\mathcal{A})\rangle$, one has $(f \circ w_{(A,\mathcal{A})} \circ cod_{(A,\mathcal{A})})(v) = (f \circ w_{(A,\mathcal{A})})(\sum_{\alpha \in \mathsf{Supp}(v)} v(\alpha) \cdot [\alpha])$
$= 0 \cdot f(1) = 0$.

Derivatives of constant functions syntactically

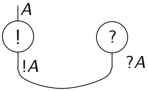


Cut-elimination of differential nets

The continuous linear function $w_{(A,\mathcal{A})} \circ cod_{(A,\mathcal{A})} : (A, \mathcal{A}) \to R$ is the zero function, which corresponds to the fact that derivatives of constant functions are zero functions.

Cut-elimination of differential nets

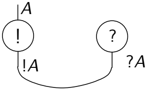
The continuous linear function $w_{(A,\mathcal{A})} \circ cod_{(A,\mathcal{A})} : (A, \mathcal{A}) \to R$ is the zero function, which corresponds to the fact that derivatives of constant functions are zero functions.

This explains that the cut 

reduces to 0.

Cut-elimination of differential nets

The continuous linear function $w_{(A,\mathcal{A})} \circ cod_{(A,\mathcal{A})} : (A, \mathcal{A}) \to R$ is the zero function, which corresponds to the fact that derivatives of constant functions are zero functions.

This explains that the cut 

reduces to 0.
And so on...

Taylor expansion

For lambda-terms u and v, we have

$$(u)v = \sum_{n \in \mathbb{N}} \frac{1}{n!} (D^n u)(0) \cdot v^n$$

The Taylor expansion of any linear logic proof can be defined in the syntax of differential nets. Then a natural question arises:

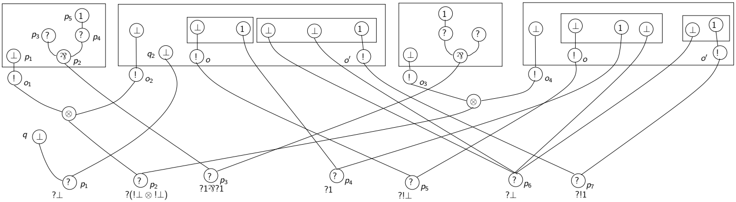
Taylor expansion

For lambda-terms u and v, we have

$$(u)v = \sum_{n \in \mathbb{N}} \frac{1}{n!} (D^n u)(0) \cdot v^n$$

The Taylor expansion of any linear logic proof can be defined in the syntax of differential nets. Then a natural question arises: Are two linear logic proofs having the same Taylor expansion equal? (the *invertibility problem of Taylor expansion*)

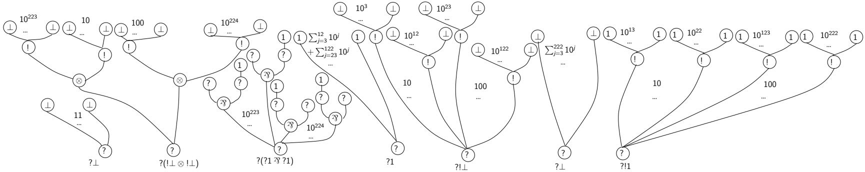
An example



There exists a 10-heterogeneous experiment f of this proof-net π s.t.

- $f^{\#}(o_1) = \{10^{223}\}$
- $f^{\#}(o_2) = \{10\}$
- $f^{\#}(o_3) = \{10^{224}\}$
- $f^{\#}(o_4) = \{100\}$
- $f^{\#}((o_2, o)) = \{10^3, 10^4, 10^5, 10^6, 10^7, 10^8, 10^9, 10^{10}, 10^{11}, 10^{12}\}$
- $f^{\#}((o_2, o')) =$ $\{10^{13}, 10^{14}, 10^{15}, 10^{16}, 10^{17}, 10^{18}, 10^{19}, 10^{20}, 10^{21}, 10^{22}\}$
- $f^{\#}((o_4, o)) = \{10^{23}, \ldots, 10^{122}\}$
- $f^{\#}((o_4, o')) = \{10^{123}, \ldots, 10^{222}\}$

$\mathcal{T}(f)[0]$



10^{223}
10
100
10^{224}
$\sum_{j=3}^{12} 10^j + \sum_{j=23}^{122} 10^j$
10^3
10^{23}
10^{12}
10^{122}
$\sum_{j=3}^{222} 10^j$
10^{13}
10^{22}
10^{123}
10^{222}
11
10^{223}
10^{224}
10
100
10
100
$?\bot$
$?(!\bot \otimes !\bot)$
$?(?1 \parr ?1)$
$?1$
$?!\bot$
$?\bot$
$?!1$

Relational model

By taking for R the semi-ring $2 = \{0, 1\}$ with $1 + 1 = 1$, a continuous linear function $R\langle(A, \mathcal{A})\rangle \to R\langle(B, \mathcal{B})\rangle$ is essentially a subset of $A \times B$. One thus retrieves a well-known model of linear logic since the 90's: the *relational model*.

Relational model

By taking for R the semi-ring $2 = \{0, 1\}$ with $1 + 1 = 1$, a continuous linear function $R\langle(A, \mathcal{A})\rangle \to R\langle(B, \mathcal{B})\rangle$ is essentially a subset of $A \times B$. One thus retrieves a well-known model of linear logic since the 90's: the *relational model*.

At that time, it was conjectured that two linear logic proofs are β-equivalent iff they are equal in the relational model (the *injectivity problem of the relational model*).

Relational model

By taking for R the semi-ring $2 = \{0, 1\}$ with $1 + 1 = 1$, a continuous linear function $R\langle(A, \mathcal{A})\rangle \to R\langle(B, \mathcal{B})\rangle$ is essentially a subset of $A \times B$. One thus retrieves a well-known model of linear logic since the 90's: the *relational model*.

At that time, it was conjectured that two linear logic proofs are β-equivalent iff they are equal in the relational model (the *injectivity problem of the relational model*).
Cf. Friedman's completeness result for λ-calculus (1975): For any two simply typed λ-terms v and u, we have

$$(v \simeq_{\beta\eta} u \Leftrightarrow [\![v]\!] = [\![u]\!])$$

where $[\![-]\!]$ is the interpretation in the full typed structure $\mathcal{M}_X$ over an infinite set X (i.e. the standard model of sets and functions, where propositional variables are interpreted by an infinite set).

Injectivity of the relational model and invertibility of the Taylor expansion

Remark. If the invertibility of Taylor expansion holds, then the injectivity of the relational model trivially holds, which shows that the invertibility problem of Taylor expansion is *not* trivial.

Injectivity of the relational model and invertibility of the Taylor expansion

Remark. If the invertibility of Taylor expansion holds, then the injectivity of the relational model trivially holds, which shows that the invertibility problem of Taylor expansion is *not* trivial.

Theorem. (C. 2018) The Taylor expansion of linear logic proofs is invertible.
Corollary. (C. 2016) The relational model is injective.

Non-idempotent intersection types

Idempotent intersection types have been introduced in the 70's by Coppo and Dezani to characterise normalisable untyped λ-terms.

Non-idempotent intersection types

Idempotent intersection types have been introduced in the 70's by Coppo and Dezani to characterise normalisable untyped λ-terms. The relational model of linear logic induces a model of the simply typed λ-calculus, which induces, through the resolution of the equation $(D \Rightarrow D) \trianglelefteq D$, a model of the untyped λ-calculus, which induces *non-idempotent* intersection types:
$D := A \mid (\mathfrak{M}_{\mathrm{fin}}(D) \times D)$

$$\frac{}{x : [\alpha] \vdash_R x : \alpha}$$

$$\frac{\Gamma, x : a \vdash_R v : \alpha}{\Gamma \vdash_R \lambda x.v : (a, \alpha)}$$

$$\frac{\Gamma_0 \vdash_R v : ([\alpha_1, \ldots, \alpha_n], \alpha) \qquad \Gamma_1 \vdash_R u : \alpha_1, \ldots, \Gamma_n \vdash_R u : \alpha_n}{\Gamma_0 + \Gamma_1 + \ldots + \Gamma_n \vdash_R (v)u : \alpha}\ n \in \mathbb{N}$$

If t is closed, then $[\![t]\!]$ is the set of its types.

Execution time

The relation between Taylor expansion and the Krivine machine inspired the following theorem:

Theorem. (C. 2007, C. 2017) For any two closed normal λ-terms u and v, the number of steps of the Krivine machine to compute $(v)u$ is

$$\inf\{|(a,\alpha)| + |a'| + 1; ((a,\alpha), a') \in \mathcal{U}^e([\![v]\!], [\![u]\!])\}$$

where $\mathcal{U}^e(X, Y)$ is the set

$$\{((a,\alpha), a') \in (X \backslash A) \times \mathfrak{M}_{\mathrm{fin}}(Y); (\exists \sigma \in \mathcal{S})(\sigma(a) = \sigma(a') \wedge \sigma(\alpha) \in D^e\}$$

with D^e the set of intersection types with no [] in positive position.

A short bibliography

- Scott, Continuous lattices, 1972
- Girard, The System F of variables types fifteen years later, 1986
- Girard, Linear logic, 1987
- Girard, Linear logic: its syntax and semantics, 1995
- Ehrhard, Finiteness spaces, 2005
- Ehrhard and Regnier, Differential interaction nets, 2006
- C., The Relational Model Is Injective for Multiplicative Exponential Linear Logic, CSL 2016
- C., Execution time of λ-terms via denotational semantics and intersection types, MSCS, 2017
- C., Taylor expansion in linear logic is invertible, LMCS, 2018
- Grellois and Melliès, Relational Semantics of Linear Logic and Higher-order Model Checking, 2015
- Vial's PhD thesis, 2017
- Chouquet and Vaux, An application of parallel cut elimination in unit-free multiplicative linear logic to the Taylor expansion of proof nets, 2018

Acknowledgement

I acknowledge Lionel Vaux for his comments about a preliminary version of these slides.