Normalization by evaluation in linear logic proof-nets

Jules Chouquet

IRIF, Université Paris-Diderot

TLLA, Oxford, September, 3, 2017

Joint work with
G. Guerrieri\textsuperscript{1}, L. Pellissier\textsuperscript{2}, L. Tortora de Falco\textsuperscript{3}, L. Vaux\textsuperscript{4}

\textsuperscript{1}Department of computer science, University of Oxford
\textsuperscript{2}LIPN, Université Paris-Nord
\textsuperscript{3}Dipartimento di matematica e fisica, Universita Roma tre.
\textsuperscript{4}I2M, Aix-Marseille Université
1 Preliminaries

2 Method for obtaining $R = nf(P)$

3 Conclusion
1 Preliminaries
   - Principles
   - Tools

2 Method for obtaining $R = \text{nf}(P)$
   - Guideline
   - Bound the complexity of the reduction and the size of $\text{nf}(P)$
   - Isolate a finite subset of $T(P)$
   - Evaluate, looking for $r[2]$

3 Conclusion
Implement a method for normalizing a proof, without computing it but looking into its denotational semantics.
Implement a method for normalizing a proof, without computing it but looking into its denotational semantics.

In our case, we give such a method for Linear Logic proof-nets, multiplicative exponential fragment, without weakenings (for now).
Plan

- Preliminaries
  - Method for obtaining \( R = \text{nf}(P) \)
  - Conclusion

**Plan**

- **P**
- **T(P)**
- **X \subset_{\text{fin}} T(P)**
- **r[2]**

**Tools**

- Expanse
- Isolate a finite subset
- Extract a well-chosen element

**Cut Elimination**

- **P**
- **R**
- **r[2]**

- **Plan**
- **Expanse**
- **Cut Elimination**

**Jules Chouquet**

Normalization by evaluation in linear logic proof-nets
Preliminaries

1. Principles
2. Tools

Method for obtaining $R = nf(P)$

- Guideline
- Bound the complexity of the reduction and the size of $nf(P)$
- Isolate a finite subset of $T(P)$
- Evaluate, looking for $r[2]$

Conclusion

Jules Chouquet
Normalization by evaluation in linear logic proof-nets
Starting point

- Computation: Cut-elimination in multiplicative exponential proof-nets.
Starting point

- **Computation**: Cut-elimination in multiplicative exponential proof-nets.
- **Denotation**: relational semantics, the proofs are interpreted in the category of sets and relations.
Starting point

- Computation: Cut-elimination in multiplicative exponential proof-nets.
- Denotation: relational semantics, the proofs are interpreted in the category of sets and relations.

The Taylor expansion of proof-nets is central, in the method used for finding the normal form of a net.
In DLL₀ : constructors are those of MELL, plus codereliction/cocontraction, and no boxes (that is the point). We consider the following nodes and reduction rules:
In DLL₀: constructors are those of MELL, plus codereliction/cocontraction, and no boxes (that is the point). We consider the following nodes and reduction rules:

```
\[ \rightarrow \]
```

Jules Chouquet  |  Normalization by evaluation in linear logic proof-nets
In \( \text{DLL}_0 \): constructors are those of MELL, plus codereliction/cocontraction, and no boxes (that is the point). We consider the following nodes and reduction rules:
In DLL$_0$: constructors are those of MELL, plus codereliction/cocontraction, and no boxes (that is the point). We consider the following nodes and reduction rules:

\[
\begin{align*}
\rightarrow & \\
\otimes & \\
\rightarrow & \\
? & \\
\sigma & \\
\end{align*}
\]
Taylor expansion of MELL proof-nets
A resource construction

\( T(P) \) is (here) a set of approximations of \( P \in \text{MELL} \).

For a promoted net \(!P\), the Taylor expansion consists in, for all \( n \in \mathbb{N} \), \( n \) copies of the content of the box, so that the conclusions of the net are the same.

We consider infinite unions of resource nets (i.e. differential nets without exponential boxes).
Taylor expansion of a box

\[ \bigcup_{n \in \mathbb{N}} (p_1 \in T(P)) \quad \cdots \quad n \quad \cdots \quad p_n \in T(P) \]

becomes

\[ !P \quad \cdots \]

\[ \vdots \]

\[ ? \quad ? \quad ! \quad \cdots \quad ? \quad \cdots \]
Preliminaries

Method for obtaining $R = nf(P)$

Conclusion

Guideline
Bound the complexity of the reduction and the size of $nf(P)$
Isolate a finite subset of $T(P)$
Evaluate, looking for $r[2]$
1 Preliminaries
   - Principles
   - Tools

2 Method for obtaining $R = \text{nf}(P)$
   - Guideline
     - Bound the complexity of the reduction and the size of $\text{nf}(P)$
     - Isolate a finite subset of $T(P)$
     - Evaluate, looking for $r[2]$

3 Conclusion
Consider now a MELL net $P$, of the shape:

$P_1 \quad \cdots \quad \text{cut} \quad \cdots \quad P_2$

where $P_1$ and $P_2$ are in normal form.
Consider now a MELL net $P$, of the shape:

\[
\begin{array}{c}
\cdots
\end{array}
\begin{array}{c}
\text{cut}
\end{array}
\begin{array}{c}
\cdots
\end{array}
\]

where $P_1$ and $P_2$ are in normal form.

We describe a method for finding the normal form of $P$ without proceeding to the cut elimination over $P$. 

\[ P_1 \quad P_2 \]
Method for obtaining $R = nf(P)$

Guideline
Bound the complexity of the reduction and the size of $nf(P)$
Isolate a finite subset of $T(P)$
Evaluate, looking for $r[2]$

Plan

- $P$ → $T(P)$
  - expanse
  - isolate a finite subset
    - $X \subset_{fin} T(P)$
    - extract a well chosen element
- $R$ → $r[2]$
  - cut elimination
  - reconstruct
$P = \text{cut}(P_1, P_2)$

- $P$ expands to $T(P)$
- $T(P)$ isolates a finite subset $X \subseteq_{\text{fin}} T(P)$
- $X$ extracts a well-chosen element
- $R$ reconstructs $r[2]$

**Plan**

1. Preliminaries
   - Method for obtaining $R = \text{nf}(P)$
2. Conclusion
   - Isolate a finite subset of $T(P)$
   - Evaluate, looking for $r[2]$
Method for obtaining $R = \text{nf}(P)$

Guideline

Bound the complexity of the reduction and the size of $\text{nf}(P)$

Isolate a finite subset of $T(P)$

Evaluate, looking for $r[2]$

Conclusion

Normalization by evaluation in linear logic proof-nets
We first apply the results of De Carvalho, Pagani, Tortora de Falco\textsuperscript{5}, establishing that:

- There exists a bound to $\#nf(P)$ depending on $\llbracket P_1 \rrbracket$ and $\llbracket P_2 \rrbracket$.

- There exists a bound to $t$, the number of cut-elimination steps needed to compute $nf(P)$, also depending on $\llbracket P_1 \rrbracket$ and $\llbracket P_2 \rrbracket$.

\textsuperscript{5}A semantic measure of the execution time in linear logic, Theoretical Computer Science 412., 2011
Plan

$P = \text{cut}(P_1, P_2)$

- **Cut Elimination**
- $R = \text{cut}(P_1, P_2)$
  - **Expanse**
  - $T(P)$
    - Isolate a finite subset
    - $X \subset_{\text{fin}} T(P)$
      - Extract a well chosen element
      - Evaluate, looking for $r[2]$
Plan

\[ P = \text{cut}(P_1, P_2) \]

- \( P \) → expanse → \( T(P) \)
- \( P \) → cut elimination → max : t steps → \( R \)
- \( R \) → reconstruct → \( r[2] \)
- \( X \subset_{\text{fin}} T(P) \) → isolate a finite subset
- \( X \subset_{\text{fin}} T(P) \) → extract a well chosen element
P = cut(P₁, P₂)

\[ \#R \leq \varphi([P₁], [P₂]) \]
1 Preliminaries
   • Principles
   • Tools

2 Method for obtaining $R = \text{nf}(P)$
   • Guideline
   • Bound the complexity of the reduction and the size of $\text{nf}(P)$
   • Isolate a finite subset of $T(P)$
   • Evaluate, looking for $r[2]$

3 Conclusion
We want to find in $T(P)$ the ressource nets that reduce into $r[2]$.

It appears that given a simple ressource net of $T(R)$, and $R' \in MELL$ s.t. $R' \rightarrow R$, we can bound the size of the element(s) of $T(R')$ that reduce into it.

This point comes from an adaptation of a technique used by Vaux\(^6\), in algebraic lambda calculus. We can’t go into the details here, but we can try to give an idea.

\(^{6}\) *Taylor expansion, $\beta$-reduction and normalization, CSL 2017*
We consider in $\text{DLL}_0$ a parallel reduction. So, the wire, of size 1, can be reduced from a net of arbitrary size:

```
  [ ]  ⋮  [ ]  ⇒  
```

Antireducts in DLL₀ proof-nets
Antireducts in DLL₀ proof-nets

We consider in DLL₀ a parallel reduction. So, the wire, of size 1, can be reduced from a net of arbitrary size:

\[
\begin{array}{ccc}
& & \\
& & \\
\end{array}
\implies
\begin{array}{c}
\end{array}
\]

**BUT** we can characterize a saving notion of depth:

**Definition**

\(\text{depth}(p) = \text{the maximal number of cuts in a switching path of } p.\)

That is bounded and preserved under reduction, in the nets that interest us.
Remark This is analogous to the structural depth in $\lambda$-calculus. It permits to go through the lack of inductive structure in proof nets.

Theorem

If $p \Rightarrow q$, and $\text{depth}(p)$ is defined, then $\#p \leq 12(\text{depth}(p))! \#q$. 
Remark This is analogous to the structural depth in \(\lambda\)-calculus. It permits to go through the lack of inductive structure in proof nets.

**Theorem**

If \(p \Rightarrow q\), and \(\text{depth}(p)\) is defined, then \(#p \leq 12(\text{depth}(p))! \#q\).

We need here \(p\) to be acyclic. Otherwise, \(\text{depth}(p)\) might be infinite.
**Remark** This is analogous to the structural depth in $\lambda$-calculus. It permits to go through the lack of inductive structure in proof nets.

**Theorem**
If $p \Rightarrow q$, and $\text{depth}(p)$ is defined, then $\#p \leq 12(\text{depth}(p))! \#q$.

We need here $p$ to be acyclic. Otherwise, $\text{depth}(p)$ might be infinite.

**Proposition**
For all $p \in T(P)$, $\text{depth}(p) \leq 2\#P$. 
Let’s then examine the one-step reduction: $R_1 \rightarrow R$, $r[2] \in T(R)$, we have:

$$\uparrow r[2] \cap T(R_1) \subseteq \{ r_1 \in T(R_1) ; \# r_1 \leq 12 \times 2^{\text{depth}(R_1)!} \times \# r[2] \}$$
Let’s then examine the one-step reduction: $R_1 \rightarrow R$, $r[2] \in T(R)$, we have:

$$\uparrow r[2] \cap T(R_1) \subset \{ r_1 \in T(R_1) ; \# r_1 \leq 12 \times 2^{\text{depth}(R_1)!} \times \# r[2] \}$$

If we remember that we have a bound on the $t \in \mathbb{N}$ s.t. $P \rightarrow^t R$, we can extend the argument to:

$$P = R_t \rightarrow \cdots \rightarrow R_3 \rightarrow R_2 \rightarrow R_1 \rightarrow R$$

and establish that $T(P) \cap \uparrow r[2]$ is included in a finite subset of $T(P)$. 

Jules Chouquet
Normalization by evaluation in linear logic proof-nets
Plan

$P = \text{cut}(P_1, P_2)$

- **Cut elimination**
- $\max : t$ steps
- **Reconstruct**

$R = \text{cut}(P_1, P_2)$

**Expanse**

$T(P)$

- **Isolate a finite subset**

$X \subset_{\text{fin}} T(P)$

- **Extract a well chosen element**

$\#R \leq \varphi([P_1], [P_2])$
$P = \text{cut}(P_1, P_2)$

$\#R \leq \varphi([P_1], [P_2])$

 Guidelines

- Bound the complexity of the reduction and the size of $\text{nf}(P)$
- Isolate a finite subset of $T(P)$
- Evaluate, looking for $r[2]$

$\text{expanse}$

$\text{cut elimination}$

$\text{max : t steps}$

$\text{reconstruct}$

$\text{isolate a finite subset}$

$X \subseteq \text{fin } T(P)$

$\text{extract a well chosen element}$

$\#r[2] \leq 2\#R$
Plan

$P = \text{cut}(P_1, P_2)$

$R = nf(P)$

Guideline
Bound the complexity of the reduction and the size of $nf(P)$
Isolate a finite subset of $T(P)$
Evaluate, looking for $r[2]$

$X \subseteq \text{fin} T(P)$

$X = \{ p \in T(P); \#p \leq \psi(P, \#r[2], t) \}$

$\#R \leq \varphi([P_1], [P_2])$

$\#r[2] \leq 2\#R$

Jules Chouquet
Normalization by evaluation in linear logic proof-nets
1 Preliminaries
   • Principles
   • Tools

2 Method for obtaining \( R = nf(P) \)
   • Guideline
   • Bound the complexity of the reduction and the size of \( nf(P) \)
   • Isolate a finite subset of \( T(P) \)
   • Evaluate, looking for \( r[2] \)

3 Conclusion
Evaluation

We now have a finite set \( \{ p_1, \ldots, p_k \} \subseteq T(P) \) of simple ressource nets, such that one of them at least\(^7\), say \( p_n \), reduces into \( r[2] \).

\(^7\)We conjecture unicity in the connected case.
Evaluation

We now have a finite set \( \{ p_1, \ldots, p_k \} \subseteq T(P) \) of simple ressource nets, such that one of them at least\(^7\), say \( p_n \), reduces into \( r[2] \).

\( p_n \) being a ressource net, its semantic is finite (no boxes), and the evaluation is linear.

\(^7\)We conjecture unicity in the connected case.

\(^8\)We have here a finite sum due to the ressource reduction letting appear possible choices of permutations.
Evaluation

We now have a finite set \( \{p_1, \ldots, p_k\} \subseteq T(P) \) of simple ressource nets, such that one of them at least\(^7\), say \( p_n \), reduces into \( r[2] \).

\( p_n \) being a ressource net, its semantic is finite (no boxes), and the evaluation is linear.

And since \( p_n \rightarrow r[2] + \cdots \)\(^8\), we have \( \llbracket r[2]\rrbracket \in \llbracket p_n \rrbracket \).

\(^7\)We conjecture unicity in the connected case.

\(^8\)We have here a finite sum due to the ressource reduction letting appear possible choices of permutations.
The injectivity result proved recently\(^9\) states in particular that, from the 2-expansion of any box-connected net in normal form, we can build the original net.

\(^9\)Guerrieri, Pellissier, Tortora de Falco: *Computing connected proof(-structure)s from their Taylor expansion*. FSCD 2016
The injectivity result proved recently\textsuperscript{9} states in particular that, from the 2-expansion of any box-connected net in normal form, we can build the original net.

In our case, from $r[2]$, we can finally construct $R = nf(P)$.

\textsuperscript{9}Guerrieri, Pellissier, Tortora de Falco : \textit{Computing connected proof(-structure)s from their Taylor expansion}. FSCD 2016
The injectivity result proved recently\(^9\) states in particular that, from the 2-expansion of any box-connected net in normal form, we can build the original net.

In our case, from \(r[2]\), we can finally construct \(R = \text{n}f(P)\).

and we are happy.

\(^9\)Guerrieri, Pellissier, Tortora de Falco: *Computing connected proof(-structure)s from their Taylor expansion*. FSCD 2016
1 Preliminaries

2 Method for obtaining $R = nf(P)$

3 Conclusion
\[ P = \text{cut}(P_1, P_2) \]

\[ X \subset_{\text{fin}} T(P) \]

\[ X = \{ p \in T(P); \#p \leq \psi(P, \#r[2], t) \} \]
Plan

\[ P = \text{cut}(P_1, P_2) \]

- Expanse: \( T(P) \)
- Acyclicity: \( X \subset \text{fin} T(P) \)
- Isolate a finite subset
- Extract a well chosen element
- Reconstruct
- Max: \( t \) steps
- \( \#R \leq \varphi([P_1], [P_2]) \)
- \( \#r[2] \leq 2\#R \)

\[ X = \{ p \in T(P); \#p \leq \psi(P, \#r[2], t) \} \]
Plan

\[ P = \text{cut}(P_1, P_2) \]

- **Expanse**
  - \( P \) \( \rightarrow \) \( T(P) \)
  - **Acyclicity**
  - \( X \subset \text{fin} \ T(P) \)
  - **Isolate a finite subset**
  - \( X \subset \text{fin} \ T(P) \)
  - **Extract a well chosen element**
  - \( \#r[2] \leq 2\#R \)
  - \( r[2] \)
  - \( \#r[2] \leq 2\#R \)

- **Box-connexity**
  - \( R \leftarrow \) \( \text{Box-connexity} \)
  - **Reconstruct**
  - \( \#R \leq \varphi([P_1], [P_2]) \)
  - \( \#R \leq \varphi([P_1], [P_2]) \)

- **Cut Elimination**
  - \( \text{max : t steps} \)
  - \( \text{cut elimination} \)
  - \( \text{cut elimination} \)

- **Preiminaries**
  - **Method for obtaining** \( R = \text{nf}(P) \)

- **Conclusion**

- **Plan**
  - **Jules Chouquet**
  - **Normalization by evaluation in linear logic proof-nets**
$P = \text{cut}(P_1, P_2)$

\[ R = nf(P) \]

Plan

- Expanse

- Acyclicity
  - Isolate a finite subset
  - Extract a well chosen element

- Cut elimination
  - Max: t steps

- Box-connexity
  - Reconstruct

\[ \#R \leq \varphi([P_1], [P_2]) \]

\[ \#r[2] \leq 2\#R \]

\[ X = \{ p \in T(P); \#p \leq \gamma(P, [P_1], [P_2]) \} \]
Establish the unicity of $p_n \rightarrow r[2] + \cdots$ in the box-connected case.

Investigate the notion of box-connexity. (ongoing work of Guerrieri, Pellissier, Tortora de Falco).

Extend the result concerning bounding the size of antireducts to all MELL, and DLL. (ongoing works of C. and Vaux).

Unify the results in a common and convenient syntax for the proof-structures, in order to giving life to the method presented (the four authors, Tortora de Falco).
Thank you for your attention.