Reversing Taylor expansion

Christine Tasson

Laboratoire Preuves Programmes Systèmes - Université Paris Diderot
Common work with Michele Pagani

March 16, 2009

Table of contents

Multiplicative Exponential Linear Logic

Formulæ

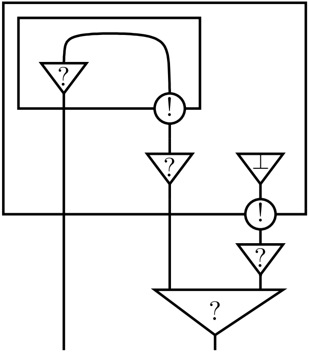
$$A, B := \ 1 \mid \bot \mid A^{\perp} \mid A \otimes B \mid A \parr B \mid !A \mid ?A$$

- Duality: $A^{\perp\perp} = A, \quad (A \parr B)^{\perp} = A^{\perp} \otimes B^{\perp}, \quad !A^{\perp} = ?A^{\perp}$.
- Linear implication: $A \multimap B = A^{\perp} \parr B$.
- Intuitionist implication: $A \Rightarrow B = !A \multimap B$.

Proof system

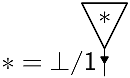
- Sequent calculus
- Nets (geometrical presentation)

An example

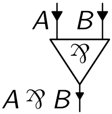


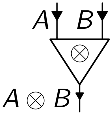
Linear logic nets

Units

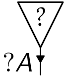


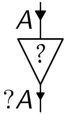
Functions

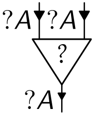




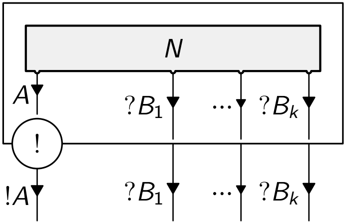
Resources







Boxes around subprograms

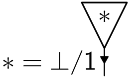


The most complex rule:

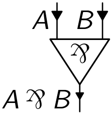
- Can be erased or duplicated during evaluation.
- No symmetry with resources.

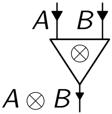
Differential nets

Units

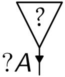


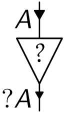
Functions

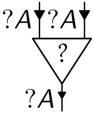




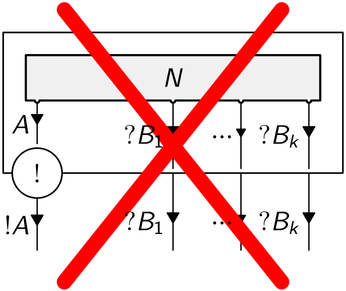
Resources







Boxes



Differential linear logic

Formulæ

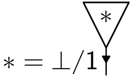
$$A, B := \ 1 \mid \bot \mid A^{\perp} \mid A \otimes B \mid A \parr B \mid {!A} \mid {?A}$$

Some differences?

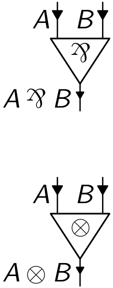
- Boxes are replaced by three new cells.
- Symmetry between $!$ and $?$ modalities.
- Erasing and duplication not on will but explicit.

Differential nets

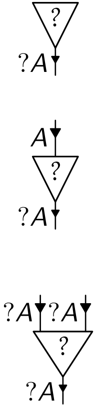
Units



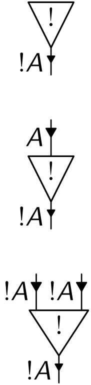
Functions



Resources



Co-resources



Taylor expansion origins

Linear logic seminal idea [Girard87]

Decompose program with respect to their resource handling:

$$\mathrm{Prog}(x) = \sum_n \mathrm{Prog}_n(\underbrace{x, \ldots, x}_{n}).$$

Semantic realization [Ehrhard05]

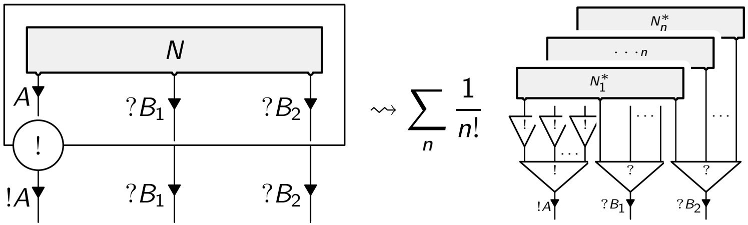
Taylor expansion in the finiteness spaces semantics.

Lambda-calculus implementation [Ehrhard-Regnier05]

- λ-terms are decomposed into sums of resource λ-terms.
- Uniform relation between resource λ-terms in the sum.
- Evaluation in λ-calculus is simulated by evaluation in resource λ-calculus.

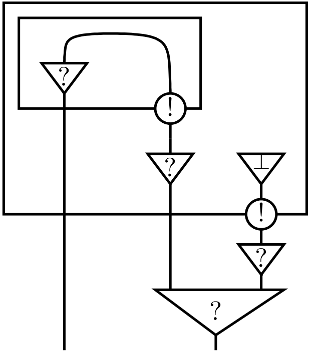
Recursive principle

Each linear logic box is mapped to a sum of differential nets.

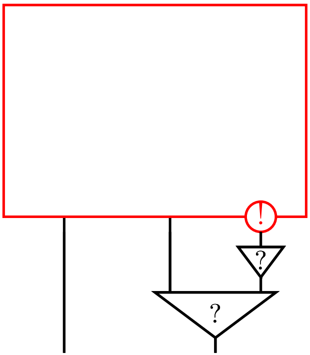


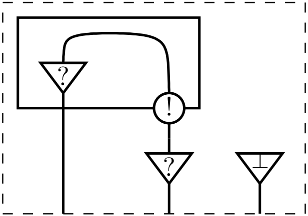
where N_k^*s appear in the Taylor expansion of N.

Taylor expansion, an example



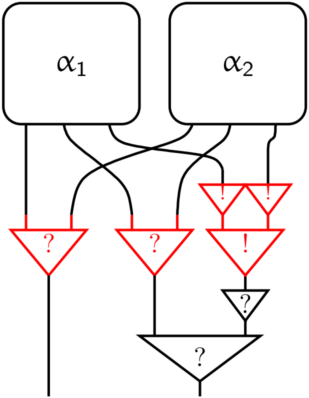
Taylor expansion, an example



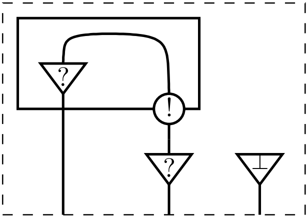


Take 2 copies.

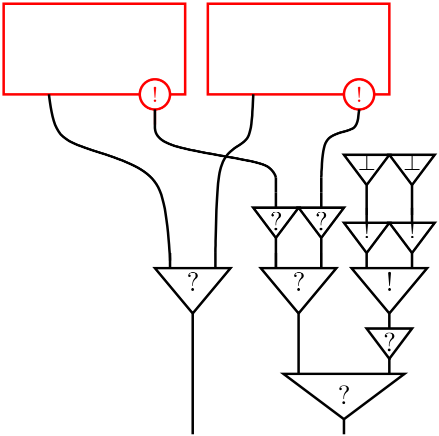
Taylor expansion, an example

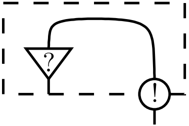


$\alpha_i \in \mathcal{T}$



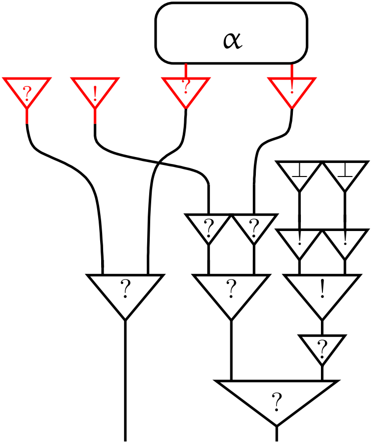
Taylor expansion, an example



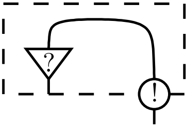


Take 0/1 copy.

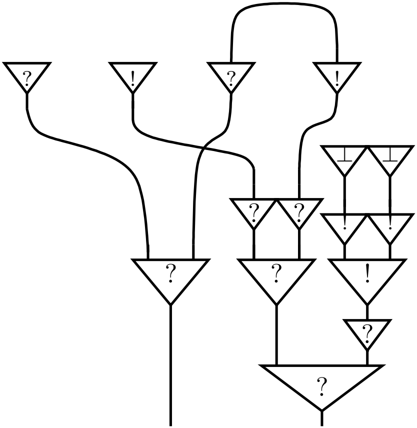
Taylor expansion, an example



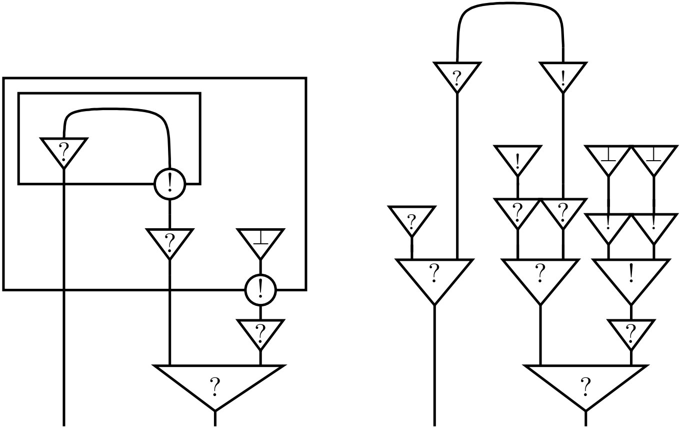
$\alpha \in \mathcal{T}$



Taylor expansion, an example



Taylor expansion, an example

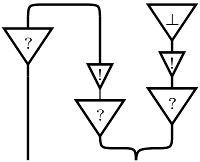


A linear logic net and a differential net in its Taylor expansion

Table of contents

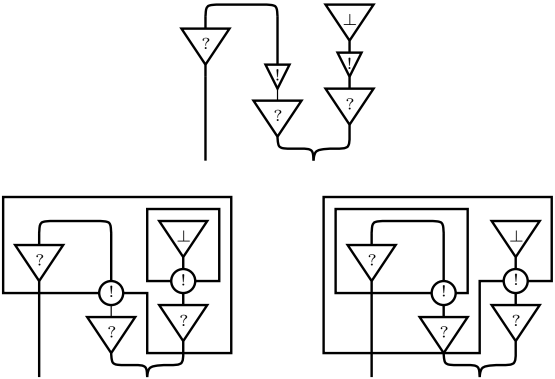
How to get back boxes?

Example



Notice that in lambda-calculus, we don't have this problem.

Example



Notice that in lambda-calculus, we don't have this problem.

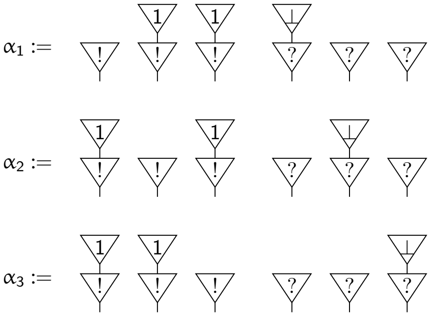
Uniformity

Definition

A collection $(\alpha_i)_{i \in J}$ of differential nets is uniform iff there is a linear logic net π such that $\forall i \in J,\ \alpha_i \in \mathcal{T}(\pi)$.

Hypercoherence

The uniformity relation is not a binary relation:



Injectivity

Relational semantics injectivity [Falco03]

In the relational semantics of linear logic, types are interpreted by sets and proofs by relations.

Question: Can we always differentiate two proofs of linear logic with their relational semantics?

$$\forall \pi \neq \pi', \exists [\cdot]_{\mathrm{Rel}} \; ; \; [\pi]_{\mathrm{Rel}} \neq [\pi']_{\mathrm{Rel}}$$

Separation Roughly speaking [Mazza-Pagani07]

- $\forall \mu \neq \mu'$ differential nets with same conclusion, there is ν such that: $\langle \mu, \nu \rangle \rightarrow_{\beta}^{*} 0$ and $\langle \mu', \nu \rangle \rightarrow_{\beta}^{*} 1$.
- Every denotational semantics of differential nets separates every two nets with different interpretations.
- Through Taylor expansion, every semantics of differential nets generates a semantics of linear logic.

Question and naive solution

Taylor inverse

We are looking for an algorithm with:

Input: A finite collection of differential nets: $(\alpha_i)_{i \in J}$.

Output: $(\alpha_i)_{i \in J}$ is

Either Dead-lock.

Or uniform and a linear logic net π such that $\forall i \in J,\ \alpha_i \in \mathcal{T}(\pi)$.

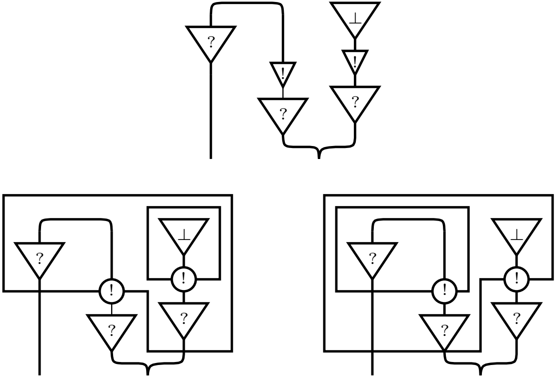
Naive solution

Try every possible subgraphs and check wanted properties.

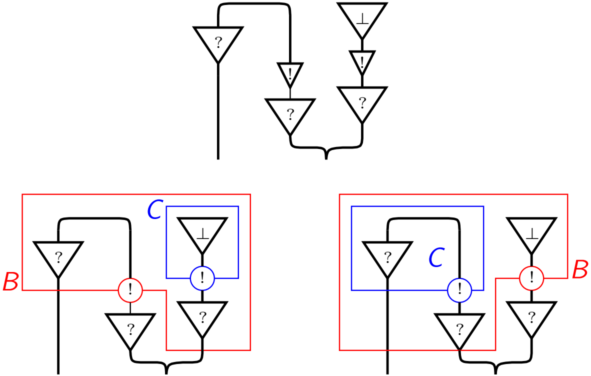
Locality

We wish to rebuild boxes locally.

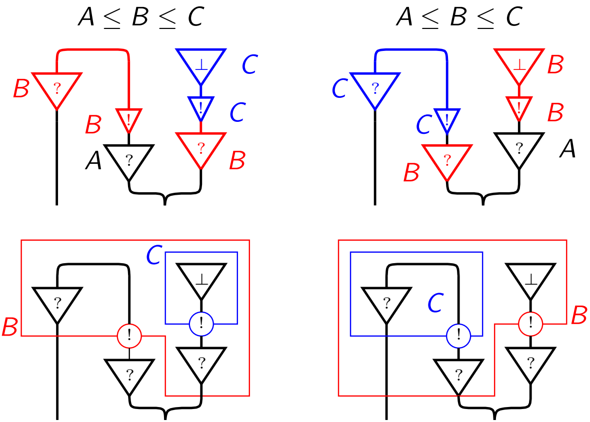
Labelling



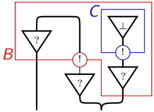
Labelling

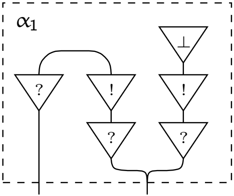


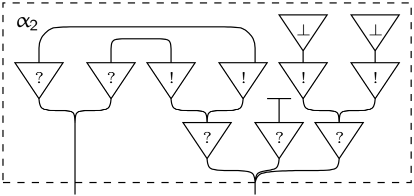
Labelling



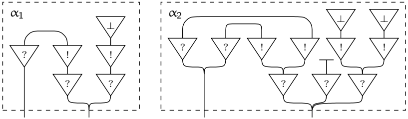
Merging on an example

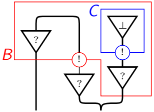


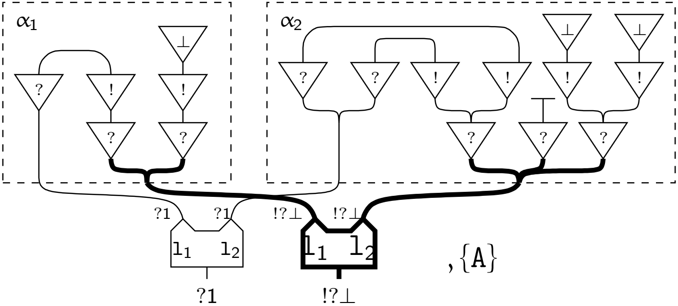




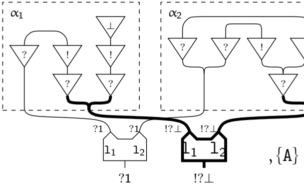
Merging on an example

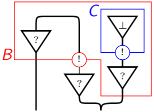


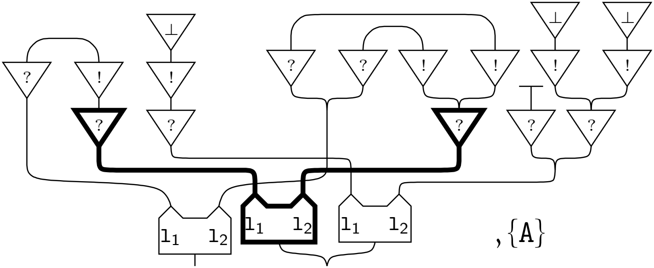




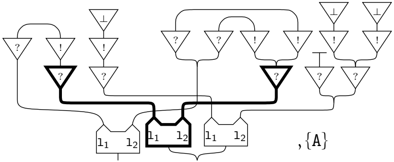
Merging on an example

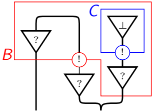


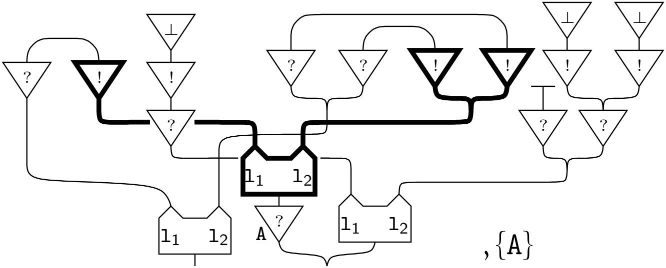




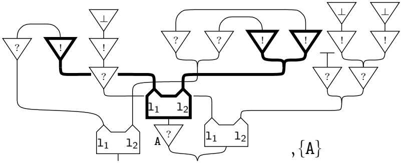
Merging on an example

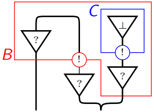


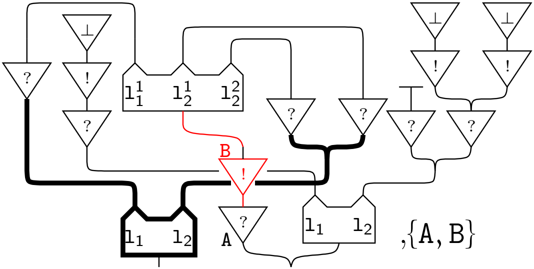




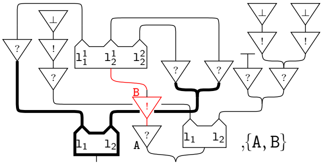
Merging on an example

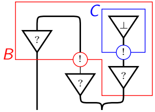


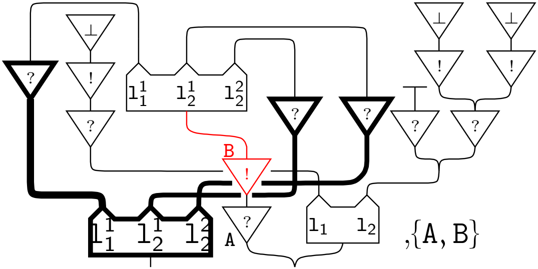




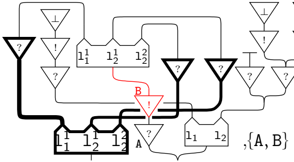
Merging on an example

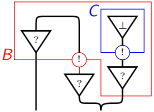


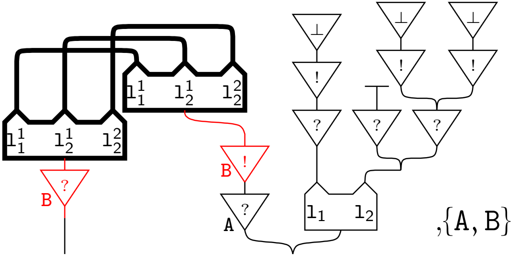




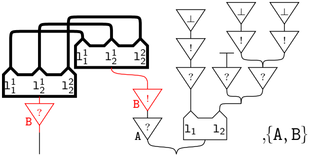
Merging on an example

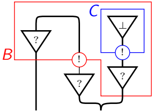


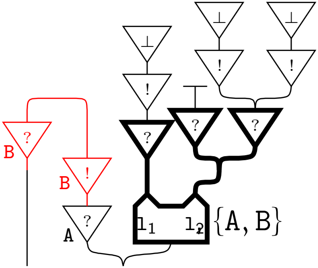




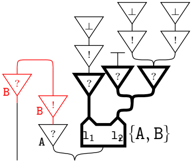
Merging on an example

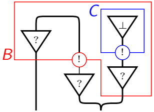


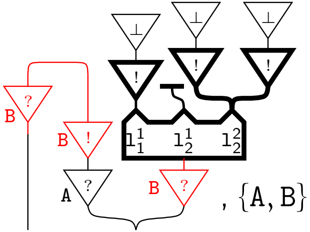




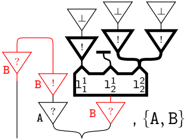
Merging on an example

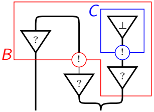


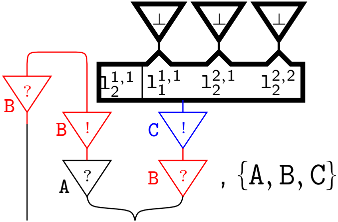




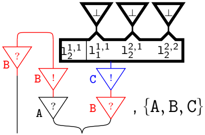
Merging on an example

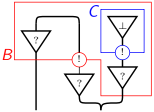




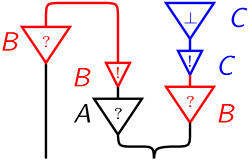


Merging on an example

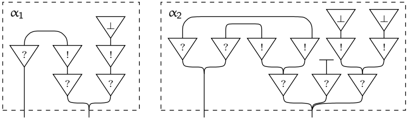


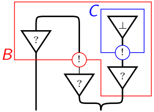


$$A \leq B \leq C$$

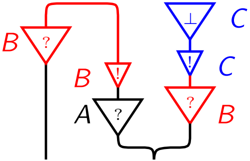


Merging on an example





$$A \leq B \leq C$$



Algorithm

Principle

Counters go through nets, merge compatible cells, and label with respect to boxes order.

Properties

- If succeeds give a witness.
- Not completely local because of the alphabet.
- Efficiency: more efficient than the naive try.
- Order: describe boxes order and Taylor expansion order.

Conclusion

Summary

We tackled the inverse Taylor expansion problem in the setting of nets and described an algorithm taking benefits from both boxes order and Taylor order.

And after

- Injectivity?
- Reduction, cut elimination.
- Coefficients and simulation.

Questions ?

Bibliographie

- [Girard] *Linear Logic*, Theoretical Computer Science, 1987
- [Ehrhard and Regnier] *The differential λ-calculus*, Theoretical Computer Science, 2003
- [Ehrhard and Regnier] *Differential Interaction Nets*, Electronic Notes in Theoretical Computer Science, 2005
- [Ehrhard] *Finiteness spaces*, Mathematical Structures in Computer Science, 2005
- [Pagani and Mazza] *The separation theorem for differential interaction nets*, LNAI, 2007
- [Falco] *Obsessional experiments for linear logic proof-nets*, Mathematical Structures in Computer Science, 2003
- [De Carvalho] *Sémantique de la logique linéaire et temps de calcul*, Ph.D. Thesis, 2007