

Realizability Proof for Normalization of Full Differential Linear Logic

Stéphane Gimenez

PPS - Université Paris Diderot

30 Juin 2011 (based on Novi Sad's talk)

Reducibility
Realizability Proof for Normalization of
Full Differential Linear Logic

Stéphane Gimenez

PPS - Université Paris Diderot

30 Juin 2011 (based on Novi Sad's talk)

Plan

Introduction

- Differentials and λ -terms

- Interaction nets, Boxes, and Reduction

- Differential constructions

- Full-DiLL

Normalization

- Previous results

- A new proof

- WN of LL (proof by Girard)

- WN of Full-DiLL

Conclusion

Introduction / Differentials and λ -terms

Analytic functions.

$$\begin{aligned} h &: \mathbb{R} \rightarrow \mathbb{R} \\ x &\mapsto 2x g(x) \end{aligned}$$

$$\begin{aligned} Dh &: \mathbb{R} \rightarrow \mathcal{L}(\mathbb{R}, \mathbb{R}) \\ x &\mapsto u \mapsto Dh(x)(u) \end{aligned}$$

$$\begin{aligned} Dh(x)(u) &= \frac{\partial}{\partial x} (2x g(x)) \cdot u \\ &= 2 g(x) \cdot u + 2x \frac{\partial}{\partial x} (g(x)) \cdot u \\ &= 2 g(x) \cdot u + 2x Dg(x)(u) \end{aligned}$$

Introduction / Differentials and λ -terms

Analytic functions.

$$\begin{aligned} h &: \mathbb{R} \rightarrow \mathbb{R} \\ x &\mapsto 2x \, g(x) \end{aligned}$$

$$\begin{aligned} Dh &: \mathbb{R} \rightarrow \mathcal{L}(\mathbb{R}, \mathbb{R}) \\ x &\mapsto u \mapsto Dh(x)(u) \end{aligned}$$

$$\begin{aligned} Dh(x)(u) &= \frac{\partial}{\partial x} (2x \, g(x)) \cdot u \\ &= 2 \, g(x) \cdot u + 2x \, \frac{\partial}{\partial x} (g(x)) \cdot u \\ &= 2 \, g(x) \cdot u + 2x \, Dg(x)(u) \end{aligned}$$

λ -terms.

$$\begin{aligned} t &: A \rightarrow B \\ &= \lambda x. x(g \, x) \end{aligned}$$

$$\begin{aligned} Dt &: A \rightarrow A \multimap B \\ &= \lambda x. \lambda u. Dt \, x \, u \end{aligned}$$

Introduction / Differentials and λ -terms

Analytic functions.

$$\begin{aligned} h &: \mathbb{R} \rightarrow \mathbb{R} \\ x &\mapsto 2x \, g(x) \end{aligned}$$

$$\begin{aligned} Dh &: \mathbb{R} \rightarrow \mathcal{L}(\mathbb{R}, \mathbb{R}) \\ x &\mapsto u \mapsto Dh(x)(u) \end{aligned}$$

$$\begin{aligned} Dh(x)(u) &= \frac{\partial}{\partial x} (2x \, g(x)) \cdot u \\ &= 2 \, g(x) \cdot u + 2x \, \frac{\partial}{\partial x} (g(x)) \cdot u \\ &= 2 \, g(x) \cdot u + 2x \, Dg(x)(u) \end{aligned}$$

λ -terms.

$$\begin{aligned} t &: A \rightarrow B \\ &= \lambda x. x(g \, x) \end{aligned}$$

$$\begin{aligned} Dt &: A \rightarrow A \multimap B \\ &= \lambda x. \lambda u. Dt \, x \, u \end{aligned}$$

$$Dt = \lambda x. \lambda u. \frac{\partial}{\partial x} (x(g \, x)) \cdot u$$

Introduction / Differentials and λ -terms

Analytic functions.

$$\begin{aligned} h &: \mathbb{R} \rightarrow \mathbb{R} \\ x &\mapsto 2x g(x) \end{aligned}$$

$$\begin{aligned} Dh &: \mathbb{R} \rightarrow \mathcal{L}(\mathbb{R}, \mathbb{R}) \\ x &\mapsto u \mapsto Dh(x)(u) \end{aligned}$$

$$\begin{aligned} Dh(x)(u) &= \frac{\partial}{\partial x} (2x g(x)) \cdot u \\ &= 2 g(x) \cdot u + 2x \frac{\partial}{\partial x} (g(x)) \cdot u \\ &= 2 g(x) \cdot u + 2x Dg(x)(u) \end{aligned}$$

λ -terms.

$$\begin{aligned} t &: A \rightarrow B \\ &= \lambda x. x(g x) \end{aligned}$$

$$\begin{aligned} Dt &: A \rightarrow A \multimap B \\ &= \lambda x. \lambda u. Dt x u \end{aligned}$$

$$\begin{aligned} Dt &= \lambda x. \lambda u. \frac{\partial}{\partial x} (x(g x)) \cdot u \\ &= \lambda x. \lambda u. (u(g x) + Dx(g(x))(\frac{\partial}{\partial x} g(x) \cdot u)) \\ &= \lambda x. \lambda u. (u(g x) + Dx(g(x))(Dg x u)) \end{aligned}$$

Introduction / Differentials and λ -terms

Analytic functions.

$$\begin{aligned} h &: \mathbb{R} \rightarrow \mathbb{R} \\ x &\mapsto 2x \, g(x) \end{aligned}$$

$$\begin{aligned} Dh &: \mathbb{R} \rightarrow \mathcal{L}(\mathbb{R}, \mathbb{R}) \\ x &\mapsto u \mapsto Dh(x)(u) \end{aligned}$$

$$\begin{aligned} Dh(x)(u) &= \frac{\partial}{\partial x} (2x \, g(x)) \cdot u \\ &= 2 \, g(x) \cdot u + 2x \, \frac{\partial}{\partial x} (g(x)) \cdot u \\ &= 2 \, g(x) \cdot u + 2x \, Dg(x)(u) \end{aligned}$$

λ -terms.

$$\begin{aligned} t &: A \rightarrow B \\ &= \lambda x. x(g \, x) \end{aligned}$$

$$\begin{aligned} Dt &: A \rightarrow A \multimap B \\ &= \lambda x. \lambda u. Dt \, x \, u \end{aligned}$$

$$\begin{aligned} Dt &= \lambda x. \lambda u. \frac{\partial}{\partial x} (x(g \, x)) \cdot u \\ &= \lambda x. \lambda u. (u(g \, x) + Dx(g(x))(\frac{\partial}{\partial x} g(x) \cdot u)) \\ &= \lambda x. \lambda u. (u(g \, x) + \textcolor{red}{D}x(g(x))(\textcolor{red}{D}g \, x \, u)) \end{aligned}$$

Introduction / Differentials and λ -terms

Differential λ -calculus [Ehrhard, Regnier, Vaux]:

$$t ::= x \mid \lambda x. t \mid 0 \mid \frac{t \, t'}{t + t'} \mid D t \cdot t'$$

β -reduction:

$$(\lambda z. t) u \longrightarrow t[u/z]$$

differential reduction:

$$D(\lambda z. t) \cdot u \longrightarrow \lambda z. \frac{\partial}{\partial z} t \cdot u$$

where $\frac{\partial}{\partial z} t \cdot u$ accounts for “linear substitution of z by u in t ”

- Resource aware calculus.

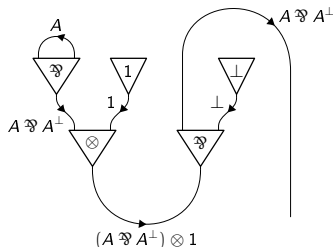
on the other side of the Curry-Howard mirror...

Introduction / Interaction nets, Boxes, and Reduction

LL in Sequent Calculus:

$$\frac{
 \frac{
 \frac{}{\vdash A, A^\perp} \text{ax}
 }{\vdash A \wp A^\perp} \text{par}
 \quad
 \frac{}{\vdash 1} \text{one}
 }{\vdash (A \wp A^\perp) \otimes 1} \text{ten}
 \quad
 \frac{
 \frac{
 \frac{}{\vdash (A^\perp \otimes A), A \wp A^\perp} \text{ax}
 }{\vdash (A^\perp \otimes A), \perp, A \wp A^\perp} \text{bot}
 }{\vdash (A^\perp \otimes A) \wp \perp, A \wp A^\perp} \text{par}
 }{\vdash A \wp A^\perp} \text{cut}$$

LL using Interaction Nets [Lafont]:

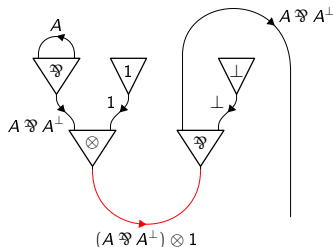


Introduction / Interaction nets, Boxes, and Reduction

LL in Sequent Calculus:

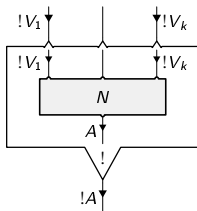
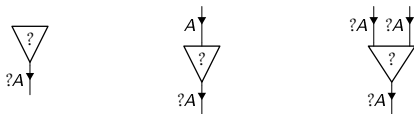
$$\frac{
 \frac{
 \frac{}{\vdash A, A^\perp} \text{ax}
 }{\vdash A \wp A^\perp} \text{par}
 \quad
 \frac{}{\vdash 1} \text{one}
 }{\vdash (A \wp A^\perp) \otimes 1} \text{ten}
 \quad
 \frac{
 \frac{
 \frac{}{\vdash (A^\perp \otimes A), A \wp A^\perp} \text{ax}
 }{\vdash (A^\perp \otimes A), \perp, A \wp A^\perp} \text{bot}
 }{\vdash (A^\perp \otimes A) \wp \perp, A \wp A^\perp} \text{par}
 }{\vdash A \wp A^\perp} \text{cut}$$

LL using Interaction Nets [Lafont]:



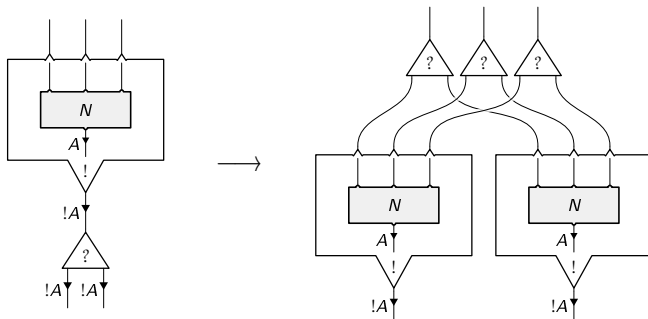
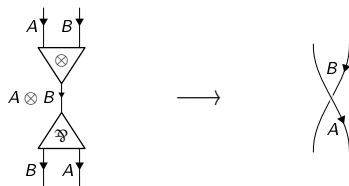
Introduction / Interaction nets, Boxes, and Reduction

Exponentials:



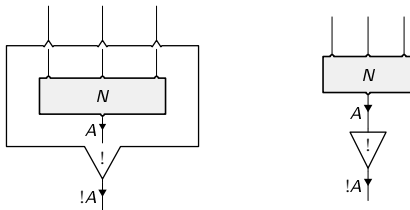
$$\frac{! \Gamma \vdash A}{! \Gamma \vdash ! A} \text{ promotion}$$

Introduction / Interaction nets, Boxes, and Reduction



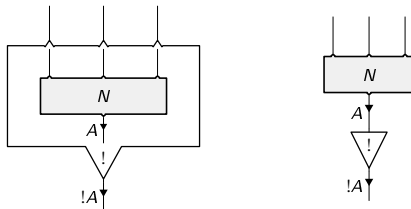
Introduction / Differential constructions

In a differential setting, we want both replicable resources and single-use resources (co-dereliction):

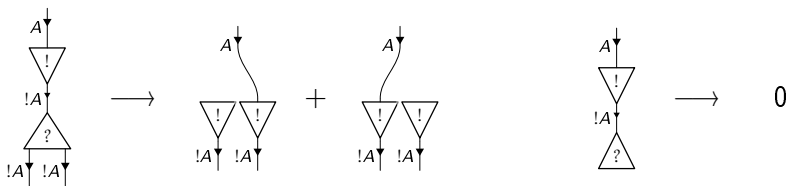


Introduction / Differential constructions

In a differential setting, we want both replicable resources and single-use resources (co-dereliction):



Routing of single-use resources:



Introduction / Differential constructions

We need a way to merge bags of resources (co-contraction):



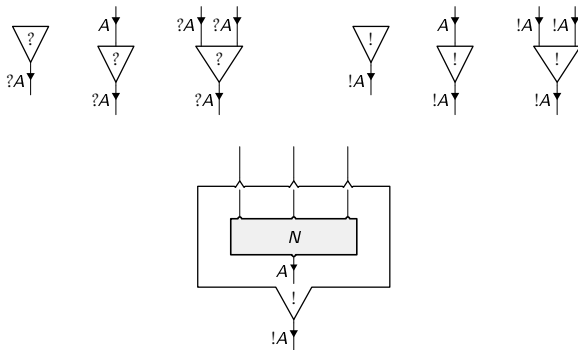
and to build empty bags of resources (co-weakening):



(Illustration with some concrete examples on the blackboard!)

Introduction / Full-DiLL

Full-DiLL constructions:

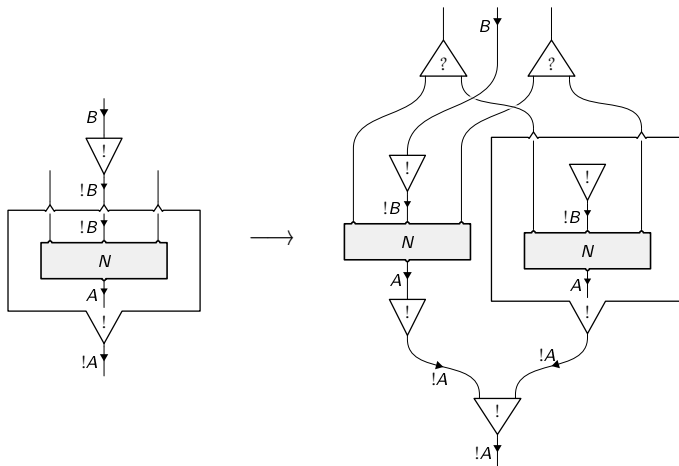


Reduction rules:

- ▶ standard *Linear Logic* reductions (4).
- ▶ *Differential Interaction Nets* reductions (9).
- ▶ 3 more rules.

Introduction / Full-DiLL

Chain rule:



Introduction / Full-DiLL

Motivations for Full-DiLL:

- ▶ Embeds Differential λ -calculus.
- ▶ Duality in consuming and producing resources.
- ▶ Links with concurrency (finite π -calculus):
 $a(x).P$ and $\bar{a}(z).P$ processes can be seen as resource producers and resource consumers.
- ▶ Promotion brings guarded replication into this picture:
 $!a(x).P$ is a replicable resource.

Does this logic enjoy the WN property? (cut-elimination theorem)
the SN property?

Normalization / Previous results

Previous normalisation results:

- ▶ Differential Interaction Nets
[Ehrhard, Regnier, 2006]
Combinatorial proof for **SN** of **Promotion free DiLL**.
- ▶ The cut-elimination theorem for Differential Linear Logic
[Pagani, 2009]
Combinatorial proof for **WN** of **Full-DiLL**.
- ▶ The Conservation Theorem for Differential Nets
[Pagani, Tranquilli, to appear]
“This turns the quest for strong normalization into one for non-erasing weak normalization”
Upgrades the previous proof to **SN** of **Full-DiLL**.

Normalization / A new proof

Other approach:

- ▶ extend Girard's proof for WN of LL.

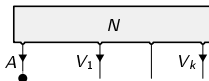
Using a reducibility technique, we obtain a modular proof. Further extensions comes for free:

- ▶ second order fragment
- ▶ additive fragment...

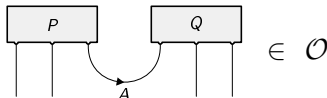
The reducibility technique (Girard).

Normalization / WN of LL (proof by Girard)

A *pointed net* is a net with a distinguished port:



Let $\mathcal{O}$ be a chosen set of nets called *observable*, $P \star Q$ holds when:



If $\mathcal{P}$ is a set of pointed nets of type A , its *dual* is defined as:

$$\mathcal{P}^* := \{ Q \mid \forall P \in \mathcal{P}, P \star Q \}$$

Normalization / WN of LL (proof by Girard)

For any set of pointed nets $\mathcal{P} \subseteq \mathcal{N}_A$ and $\mathcal{Q} \subseteq \mathcal{N}_A$, the following properties hold:

$$\mathcal{P} \subseteq \mathcal{Q} \implies \mathcal{Q}^* \subseteq \mathcal{P}^* \qquad \mathcal{P} \subseteq \mathcal{P}^{**} \qquad \mathcal{P}^{***} = \mathcal{P}^*$$

A *behavior* at type A is any set $\mathcal{P}$ of pointed nets of type A which is equal to its double dual:

$$\mathcal{P} = \mathcal{P}^{**}$$

Normalization / WN of LL (proof by Girard)

We can define a behaviour $\llbracket 1 \rrbracket \subseteq \mathcal{N}_1$ as follows:

$$\llbracket 1 \rrbracket := \left\{ \begin{array}{c} \text{1} \\ \nabla \\ \text{1} \downarrow \\ \bullet \end{array} \right\}^{**}$$

Behavior $\llbracket A \otimes B \rrbracket \subseteq \mathcal{N}_{A \otimes B}$ is defined from behaviors $\llbracket A \rrbracket \subseteq \mathcal{N}_A$ and $\llbracket B \rrbracket \subseteq \mathcal{N}_B$:

$$\llbracket A \otimes B \rrbracket := \left\{ \begin{array}{c} \begin{array}{cc} \begin{array}{|c|} \hline P \\ \hline \end{array} & \begin{array}{|c|} \hline Q \\ \hline \end{array} \\ \begin{array}{c} \text{A} \swarrow \quad \searrow \text{B} \\ \otimes \\ \text{A} \otimes \text{B} \downarrow \\ \bullet \end{array} \end{array} \middle| P \in \llbracket A \rrbracket, Q \in \llbracket B \rrbracket \right\}^{**}$$

Normalization / WN of LL (proof by Girard)

The behavior $\llbracket !A \rrbracket \subseteq \mathcal{N}_{!A}$ is defined from $\llbracket A \rrbracket \subseteq \mathcal{N}_A$ as follows:

$$\llbracket !A \rrbracket := \left\{ \left(\begin{array}{c} \text{Diagram: A box labeled } P \text{ with three input wires from above and one output wire labeled } A \text{ pointing down to a junction. The junction has a downward arrow labeled } !A \text{ leading to a black dot.} \\ P \in \llbracket A \rrbracket \\ \text{with adequate} \\ \text{interface} \end{array} \right) \mid \right\}^{**}$$

And $\llbracket ?A \rrbracket \subseteq \mathcal{N}_{?A}$ is defined by:

$$\llbracket ?A \rrbracket := \llbracket !A^\perp \rrbracket^*$$

Normalization / WN of LL (proof by Girard)

We now chose for $\mathcal{O}$ the set of all weakly normalizing nets. So, $P \star Q$ means that the interaction between P and Q reduces to a normal form.

Property

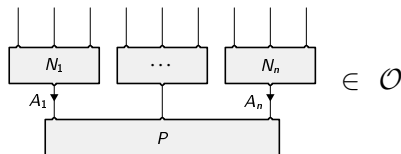
Behaviors $\llbracket A \rrbracket$ previously defined are reducibility candidates:

$$\text{Axiom}_A \in \llbracket A \rrbracket \qquad \llbracket A \rrbracket \subseteq \mathcal{O}_A$$

Normalization / WN of LL (proof by Girard)

Definition (Net reducibility)

A net P is *reducible* when for all $N_i \in \llbracket A_i \rrbracket$, we have:



Main lemma:

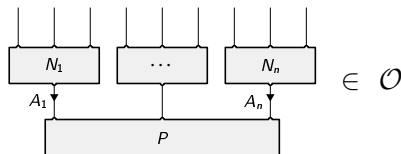
Lemma

Every valid net is reducible.

Normalization / WN of LL (proof by Girard)

Definition (Net reducibility)

A net P is *reducible* when for all $N_i \in \llbracket A_i \rrbracket$, we have:



Main lemma:

Lemma

Every valid net is reducible.

(Proof by induction on net sequentializations).

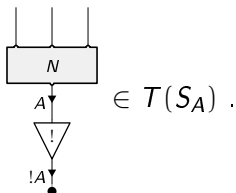
From LL to Full-DiLL...

reducibility candidates for exponential types must be built differently

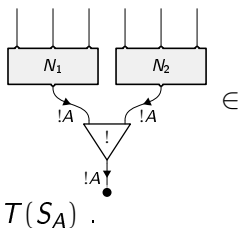
Normalization / WN of Full-DiLL

Given a set $\mathcal{S}_A$ of pointed nets, one can build $T(\mathcal{S}_A)$ such that:

- ▶ if $N \in \mathcal{S}_A$, then

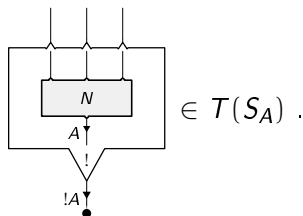


- ▶ if $N_1, N_2 \in T(\mathcal{S}_A)$, then



- ▶ $\in T(\mathcal{S}_A)$.

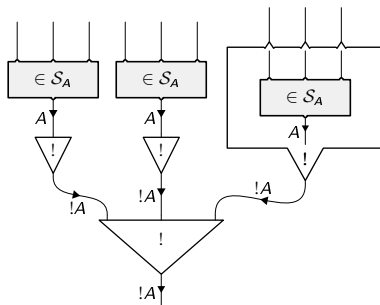
- ▶ if $N \in \mathcal{S}_A$ has adequate interface, then



- ▶ the axiom $\in T(\mathcal{S}_A)$.

Normalization / WN of Full-DiLL

Going from $\mathcal{S}_A$ to $T(\mathcal{S}_A)$ consists in applying an exponential macro-construction, restricted to a single exponential layer.



Now, we can define Full-DiLL exponential candidates as follows:

$$\llbracket !A \rrbracket := (T(\llbracket A \rrbracket))^{\star\star}$$

Normalization / WN of Full-DiLL

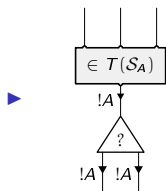
The rest of the proof is an induction and relies on:

- ▶ A property about reduction of exponential macro-constructions (weakening, dereliction, contraction cases)
- ▶ A technical lemma to deal with the chain rule (promotion case).

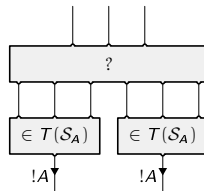
And we need to reuse a little trick to deal with the co-contraction case.

Normalization / WN of Full-DiLL

Property

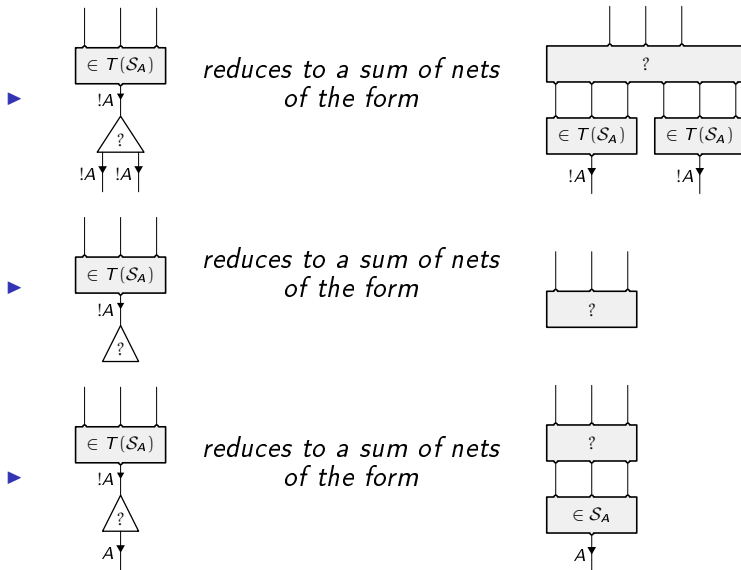


*reduces to a sum of nets
of the form*



Normalization / WN of Full-DiLL

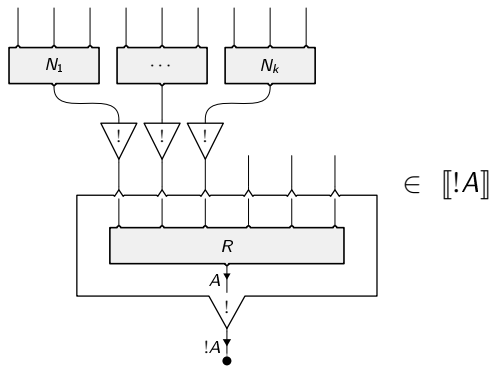
Property



Normalization / WN of Full-DiLL

Lemma

Take any reducible pointed net R with [adequate interface], and let $N_1 \in \llbracket V_1 \rrbracket, \dots, N_k \in \llbracket V_k \rrbracket$, we have:



Conclusion

Result:

- ▶ A modular normalization proof for WN of Full-DiLL.
- ▶ Due to modularity this proof extends trivially to WN of “Second Order Full-DiLL with Additives”.

Conclusion

Result:

- ▶ A modular normalization proof for WN of Full-DiLL.
- ▶ Due to modularity this proof extends trivially to WN of “Second Order Full-DiLL with Additives”.

Related work:

- ▶ Strong normalization property for second order linear logic [Pagani, Tortora de Falco, 2010]
- ▶ The Conservation Theorem for Differential Nets [Pagani, Tranquilli, to appear]

Pending work:

- ▶ Strong normalization of “Second Order Full-DiLL”.

Conclusion

Result:

- ▶ A modular normalization proof for WN of Full-DiLL.
- ▶ Due to modularity this proof extends trivially to WN of “Second Order Full-DiLL with Additives”.

Related work:

- ▶ Strong normalization property for second order linear logic [Pagani, Tortora de Falco, 2010]
- ▶ The Conservation Theorem for Differential Nets [Pagani, Tranquilli, to appear]

Pending work:

- ▶ Strong normalization of “Second Order Full-DiLL”.

Thank you!