

Proof-nets and their semantics expressed by local means (in english-puis-français)

Stéphane Gimenez

PPS - Université Paris Diderot

CHOCO – 7 Octobre 2010

Plan

Introduction

Localized Proof-nets

Localized Semantics

Conclusion

Introduction

- Interaction nets

- Boxes

- Interaction combinators

- Multiplexing

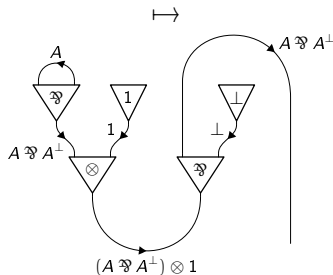
Localized Proof-nets

Localized Semantics

Conclusion

Born from multiplicative linear logic [Laf90]:

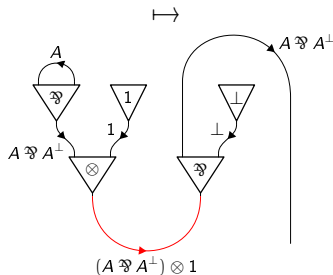
$$\begin{array}{c}
\frac{}{\vdash A, A^\perp} \text{ax} \\
\frac{}{\vdash A \wp A^\perp} \text{par} \quad \frac{}{\vdash 1} \text{one} \\
\hline
\vdash (A \wp A^\perp) \otimes 1 \quad \text{ten}
\end{array}
\quad
\begin{array}{c}
\frac{}{\vdash (A^\perp \otimes A), A \wp A^\perp} \text{ax} \\
\frac{}{\vdash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



Introduction / Interaction nets

Born from multiplicative linear logic [Laf90]:

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



Introduction / Interaction nets

Why interaction nets are interesting:

- ▶ a strong confluence (diamond property)
- ▶ local interaction (size of redexes is bounded)
- ▶ reduction can be done in parallel

Introduction / Interaction nets

Why interaction nets are interesting:

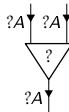
- ▶ a strong confluence (diamond property)
- ▶ local interaction (size of redexes is bounded)
- ▶ reduction can be done in parallel

We'd like to encode λ -calculus, PCF, ... into such a system. We need a way to encode:

- ▶ Multiplicatives
- ▶ Exponentials
- ▶ Additives
- ▶ Recursion

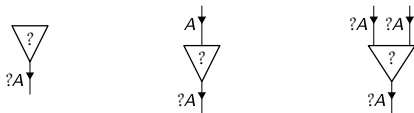
Introduction / Boxes

Exponentials:

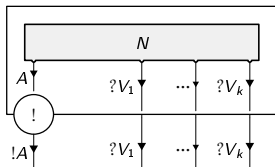


Introduction / Boxes

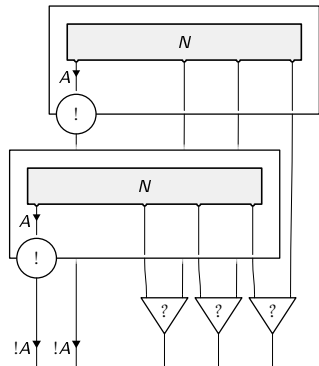
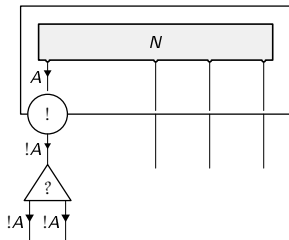
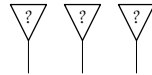
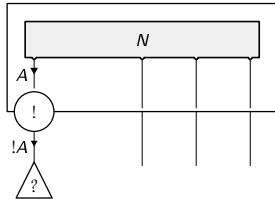
Exponentials:



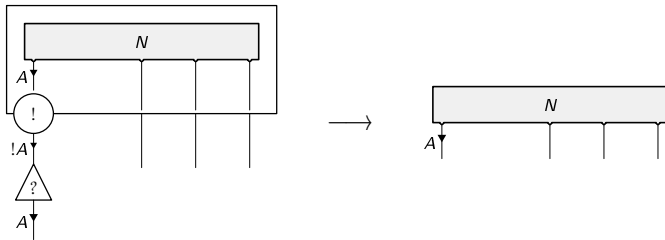
But we need boxes to represent promotion: $\frac{\vdash A, ?\Gamma}{\vdash !A, ?\Gamma} \text{ prom}$



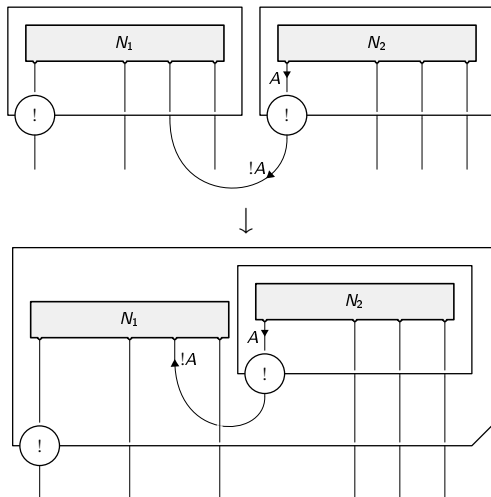
Introduction / Boxes



Introduction / Boxes



Introduction / Boxes



Introduction / Boxes

Boxes are unconventional cells with **several principal ports** which are **parametrized by a net** of arbitrary size.

Introduction / Boxes

Boxes are unconventional cells with **several principal ports** which are **parametrized by a net** of arbitrary size.

Issues:

- ▶ using boxes, the reduction is not automatically confluent anymore (not a big issue)
- ▶ reductions are not local
- ▶ duplications of top-level boxes prevent parallel computation.

Introduction / Boxes

Boxes are unconventional cells with **several principal ports** which are **parametrized by a net** of arbitrary size.

Issues:

- ▶ using boxes, the reduction is not automatically confluent anymore (not a big issue)
- ▶ reductions are not local
- ▶ duplications of top-level boxes prevent parallel computation.

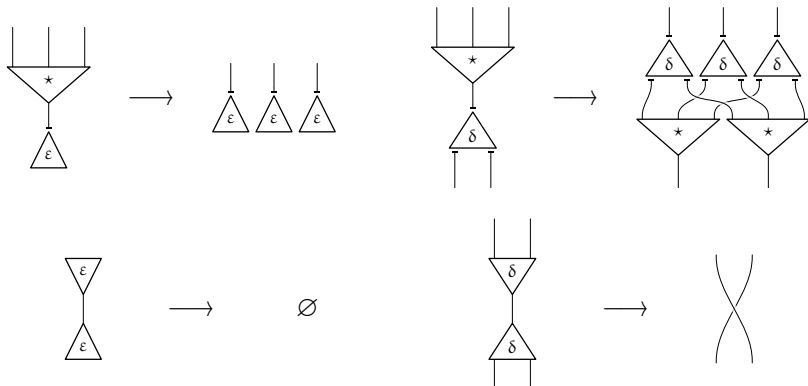
We'd like to use simple cells to perform locally duplication/erasing of boxes contents...

Introduction / Interaction combinators

Two first combinators:



Their interactions:



Introduction / Interaction combinators

How to use them ?

- ▶ adding a specific third cell γ , to obtain the original system of *interaction combinators* [Laf97]. (universality property).
- ▶ adding a second duplication operator ζ like in the *symmetric interaction combinators* [Maz07].
- ▶ using them together with other cells (e.g. multiplicatives)

Introduction / Interaction combinators

How to use them ?

- ▶ adding a specific third cell γ , to obtain the original system of *interaction combinators* [Laf97]. (universality property).
- ▶ adding a second duplication operator ζ like in the *symmetric interaction combinators* [Maz07].
- ▶ using them together with other cells (e.g. multiplicatives)

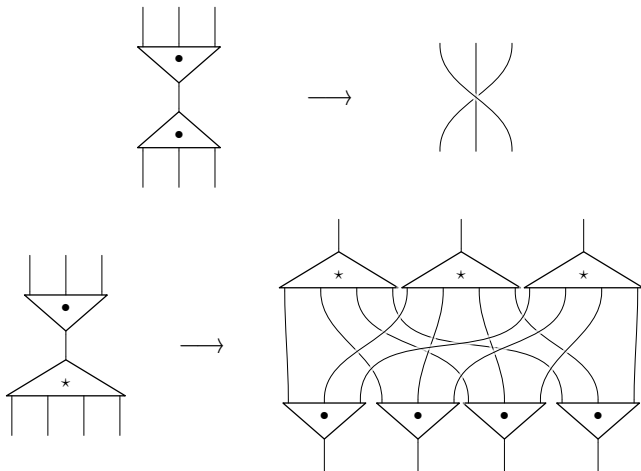
How will we use them ?

- ▶ using them together with logic cells
- ▶ **and** using many of them (indexed by an natural number)

Introduction / Multiplexing

Definition (multiplexing operator)

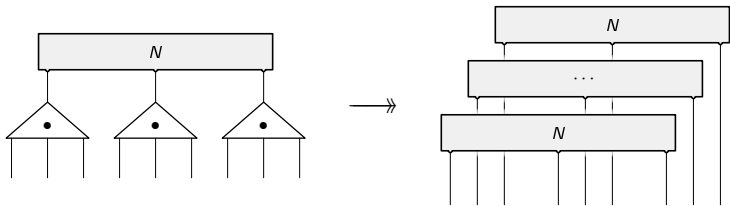
An operator $\bullet$ is called *multiplexing operator* over a set of operators $\star \in \mathcal{E}$ when its reductions are:



Introduction / Multiplexing

Property (replication by mutiplexing)

Given any operator $\bullet$ that is multiplexing over every operator in a net N supposed to be in *normal* form and deadlock free:



Introduction

Localized Proof-nets

- Active localization of Exponentials

- Passive localization of Exponentials

- Controlled localization of Exponentials

- Summary

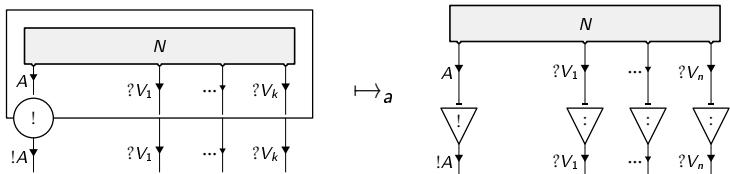
- Additives and Recursion

Localized Semantics

Conclusion

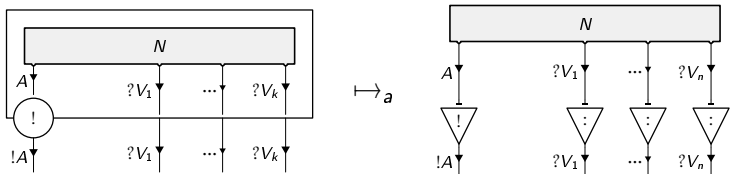
Localized Proof-nets / Active localization of Exponentials

L'idée: ne matérialiser que l'interface des boîtes

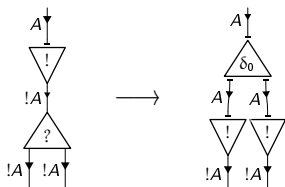


Localized Proof-nets / Active localization of Exponentials

L'idée: ne matérialiser que l'interface des boîtes

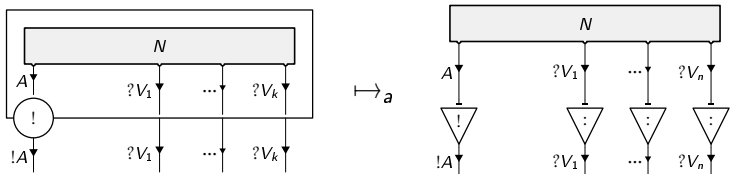


Le fonctionnement:

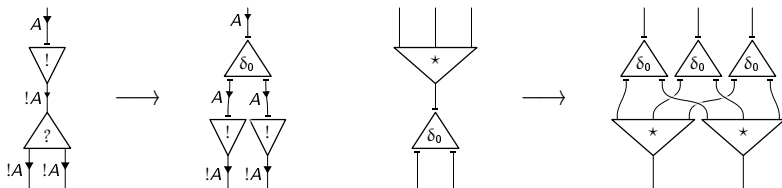


Localized Proof-nets / Active localization of Exponentials

L'idée: ne matérialiser que l'interface des boîtes

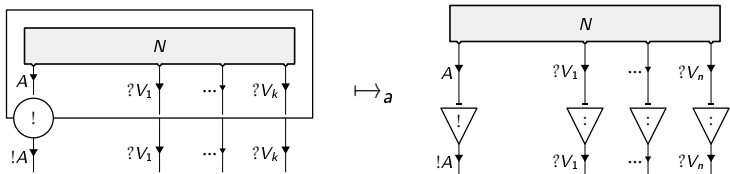


Le fonctionnement:

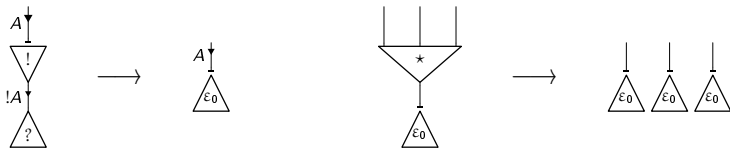


Localized Proof-nets / Active localization of Exponentials

L'idée: ne matérialiser que l'interface des boîtes

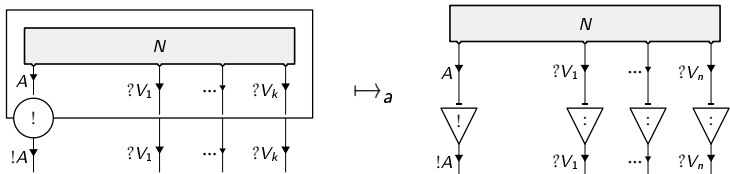


Le fonctionnement:

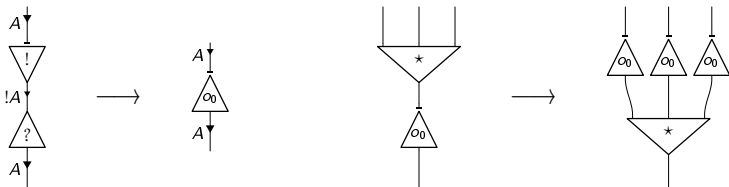


Localized Proof-nets / Active localization of Exponentials

L'idée: ne matérialiser que l'interface des boîtes

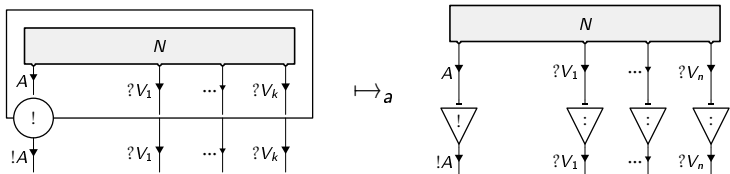


Le fonctionnement:

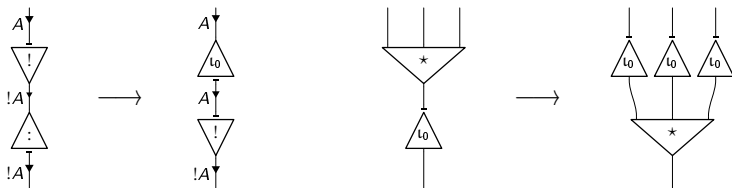


Localized Proof-nets / Active localization of Exponentials

L'idée: ne matérialiser que l'interface des boîtes

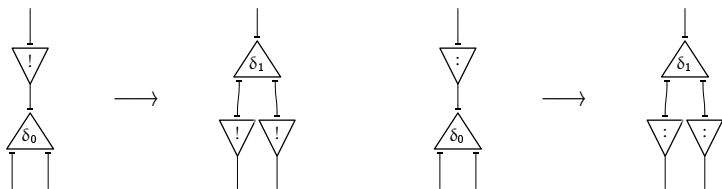


Le fonctionnement:



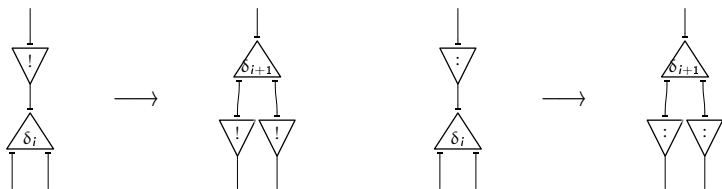
Localized Proof-nets / Active localization of Exponentials

Ce qu'il se passe à l'interface des boîtes :



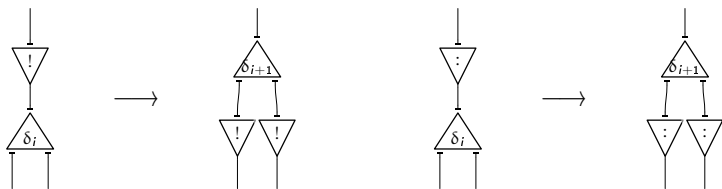
Localized Proof-nets / Active localization of Exponentials

Ce qu'il se passe à l'interface des boîtes :

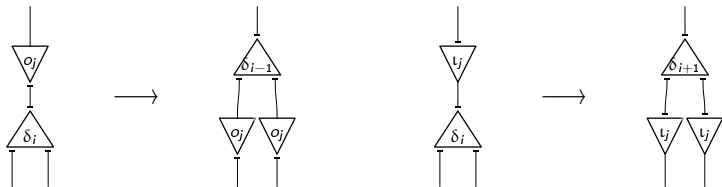


Localized Proof-nets / Active localization of Exponentials

Ce qu'il se passe à l'interface des boîtes :

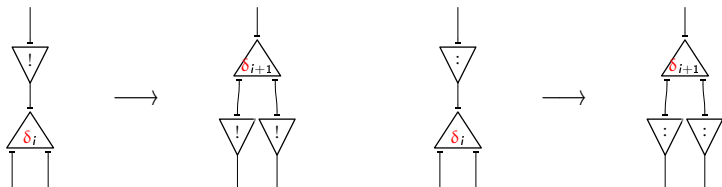


Aussi, lorsque $j < i$:

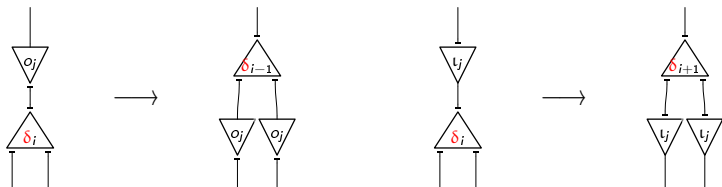


Localized Proof-nets / Active localization of Exponentials

Ce qu'il se passe à l'interface des boîtes (pareil pour ε, o, ι):



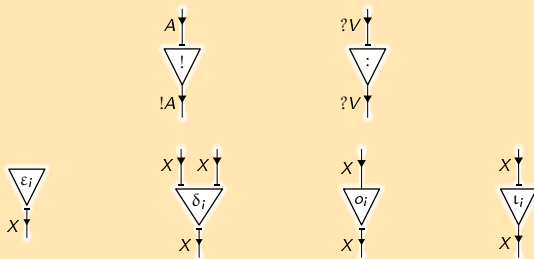
Aussi, lorsque $j < i$:



Localized Proof-nets / Active localization of Exponentials

On a réussi à remplacer les boîtes par différents nœuds simples.

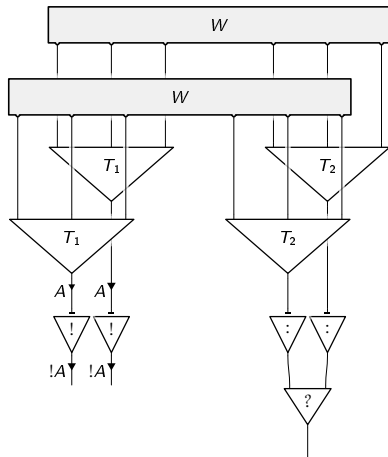
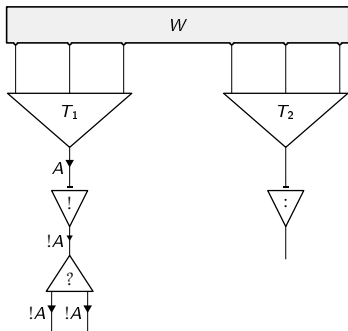
Le système exponentiel localisé **actif**



- Ce système est similaire au système des « sharing graphs ». (*J. Lamping* [Lam90], puis *G. Gonthier*, *M. Abadi*, *J.-J. Levy*)
- On conserve un peu plus de structure logique, mais pas tout.

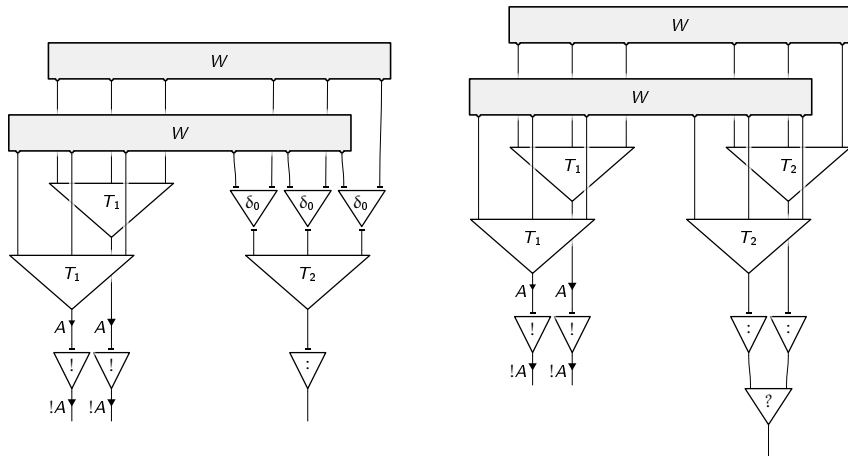
Localized Proof-nets / Active localization of Exponentials

Un inconvénient pour ce système actif:



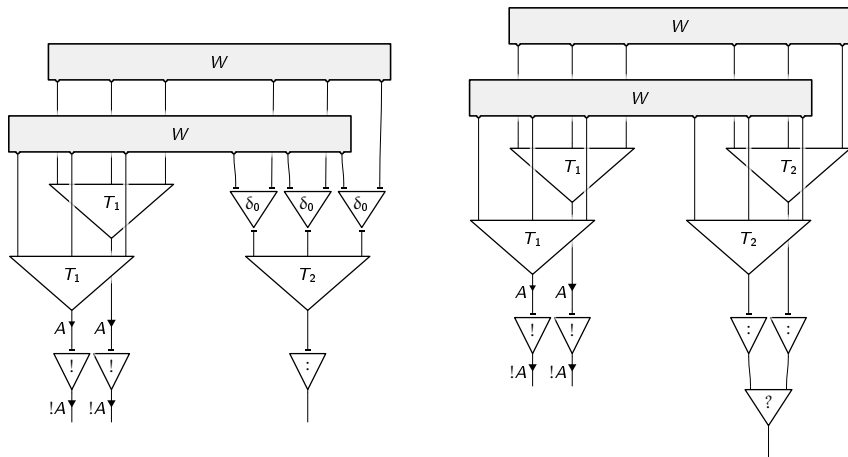
Localized Proof-nets / Active localization of Exponentials

Un inconvénient pour ce système actif:



Localized Proof-nets / Active localization of Exponentials

Un inconvénient pour ce système actif:

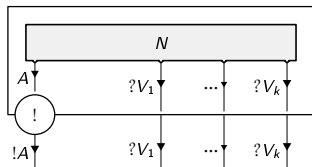


Le problème de la « relecture »

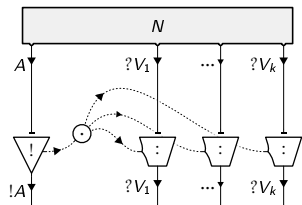
(S. Guerrini, S. Martini, A. Masini [GMM03])

Localized Proof-nets / Passive localization of Exponentials

L'idée: un canal de contrôle, et des messages

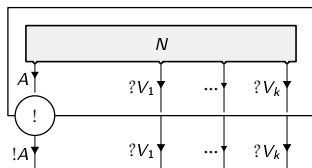


$\mapsto_p$

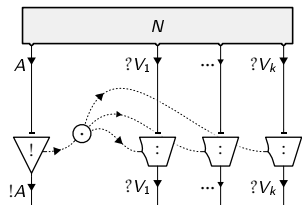


Localized Proof-nets / Passive localization of Exponentials

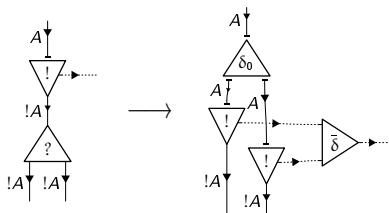
L'idée: un canal de contrôle, et des messages



$\mapsto p$

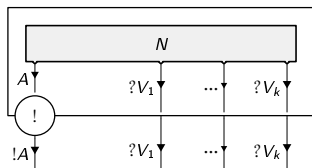


Le fonctionnement:

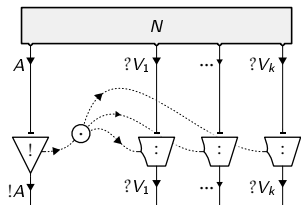


Localized Proof-nets / Passive localization of Exponentials

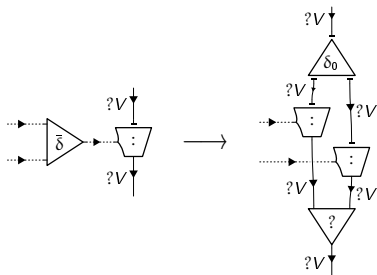
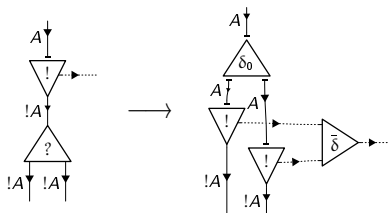
L'idée: un canal de contrôle, et des messages



$\mapsto_p$

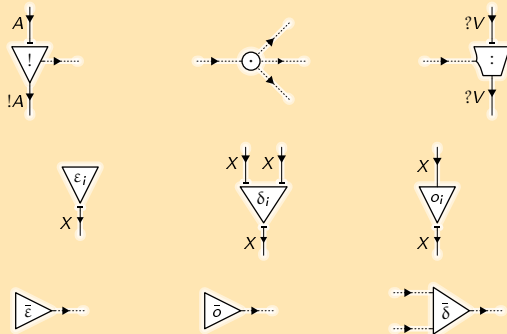


Le fonctionnement:



Localized Proof-nets / Passive localization of Exponentials

Le système exponentiel localisé **passif**



- Les systèmes **actif** et **passif** s'inscrivent dans le cadre des réseaux d'interaction, une **confluence forte** de la réduction est automatiquement assurée.

Localized Proof-nets / Passive localization of Exponentials

Theorem

*Le codage dans le système exponentiel localisé **passif** simule faiblement la **réduction interne** et **sans commutations exponentielles** des preuves de la logique linéaire.*

Corollary

*Ce codage simule la normalisation (pour la **réduction sans commutations exponentielles**) d'une preuve de la logique linéaire.*

Localized Proof-nets / Passive localization of Exponentials

Theorem

*Le codage dans le système exponentiel localisé **passif** simule faiblement la **réduction interne** et **sans commutations exponentielles** des preuves de la logique linéaire.*

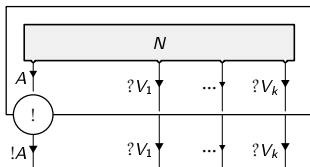
Corollary

*Ce codage simule la normalisation (pour la **réduction sans commutations exponentielles**) d'une preuve de la logique linéaire.*

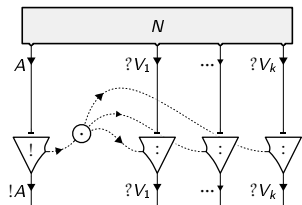
Comment retrouver les commutations exponentielles?

Localized Proof-nets / Controlled localization of Exponentials

L'idée: des nœuds avec plusieurs ports principaux.

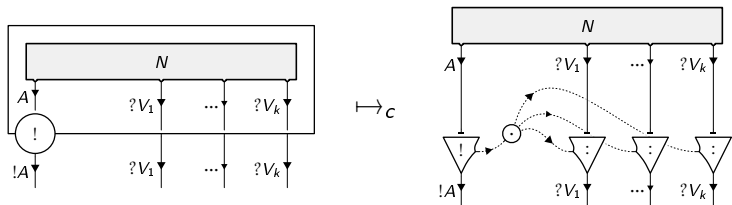


$\mapsto_C$

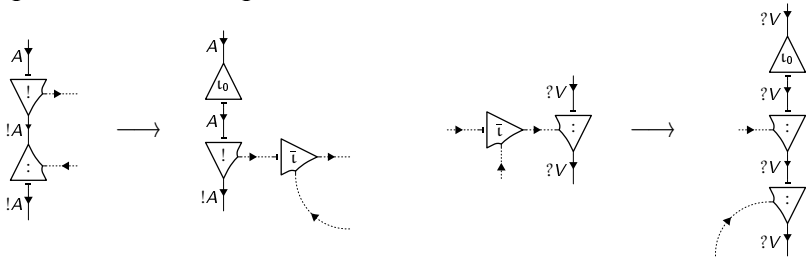


Localized Proof-nets / Controled localization of Exponentials

L'idée: des nœuds avec plusieurs ports principaux.

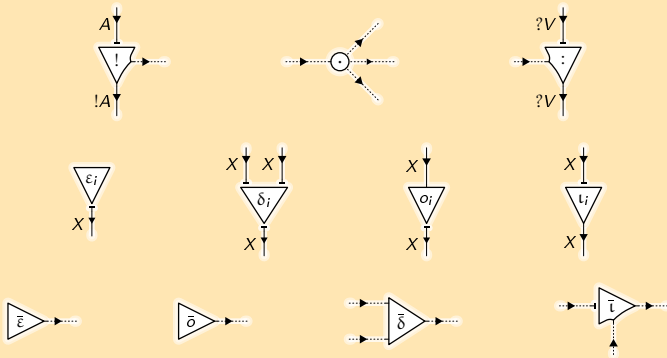


Les opérateurs d'*ingestion* retrouvent leur utilité, et il nous faut également un message associé:



Localized Proof-nets / Controlled localization of Exponentials

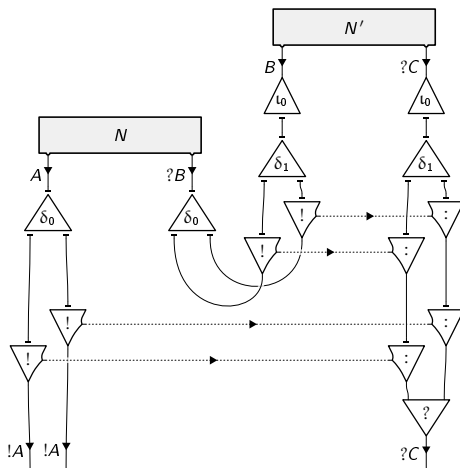
Le système exponentiel localisé **contrôlé**



- On a utilisé des multi-nœuds, la confluence n'est plus automatique.

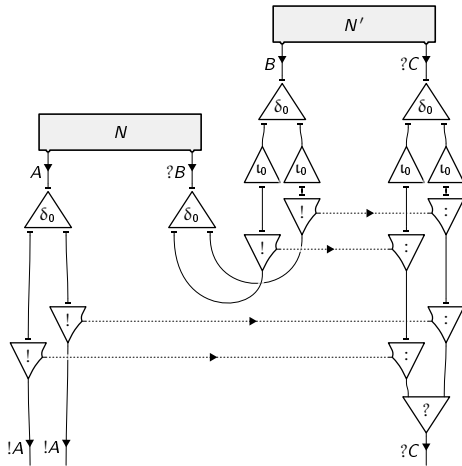
Localized Proof-nets / Controlled localization of Exponentials

On rate la confluence locale de peu:



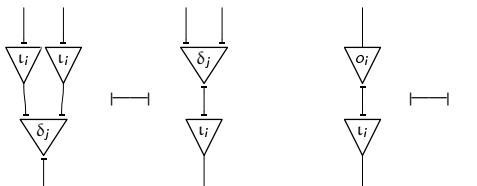
Localized Proof-nets / Controlled localization of Exponentials

On rate la confluence locale de peu:



Localized Proof-nets / Controlled localization of Exponentials

Pour la confluence du système contrôlé, il faut d'abord remarquer quelques équivalences:



Theorem

La réduction du système exponentiel localisé *contrôlé* est *localement confluente modulo* commutation des opérateurs de boîtes [Hue80].

Localized Proof-nets / Controled localization of Exponentials

Theorem

*Le codage dans le système exponentiel localisé **contrôlé** simule faiblement la **réduction interne** des preuves de la logique linéaire.*

Corollary

Ce codage simule la normalisation (usuelle) d'une preuve de la logique linéaire.

Localized Proof-nets / Summary

système	cardinalité	com. exp.	relecture	nœuds	optimalité?
actif	infini	oui	non	normaux	séquentielle
passif	fini	non	oui	normaux	-
contrôlé	infini	oui	oui	multiples	parallèle

Théorèmes:

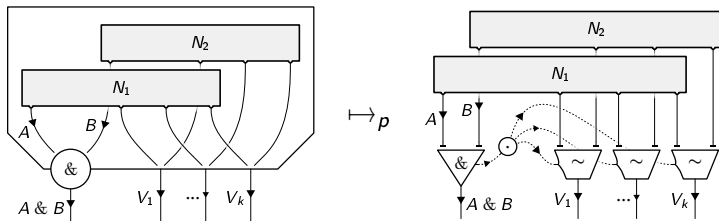
- ▶ actif: confluence forte, problème de la relecture
- ▶ passif: confluence forte, simulation précise
- ▶ contrôlé: confluence modulo, simulation précise

Localized Proof-nets / Additives and Recursion

On peut exécuter le λ -calcul (typé) grâce aux codages présentés.

On parvient également à exécuter le langage *PCF* car la méthode de localisation s'adapte:

- ▶ aux **boîtes additives**:



- ▶ à des **boîtes de récursion** introduites spécifiquement pour coder la construction récursive présente dans *PCF*.

Introduction

Localized Proof-nets

Localized Semantics

- Interpretation of types

- Standard relational semantics

- Exponential Trees

- Tree closures

- Without control channels

- With control channels

Conclusion

Localized Semantics / Interpretation of types

Soit $|\cdot|$ une fonction d'interprétation qui associe un ensemble à tout type de base. On peut l'étendre à tous les types de la façon suivante:

$$|A \wp B| = |A \otimes B| = |A| \times |B|$$

$$|1| = |\perp| = \mathcal{U}$$

$$|!A| = |?A| = \mathcal{M} |A|$$

$$|A \& B| = |A \oplus B| = |A| + |B|$$

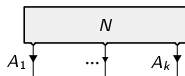
$$|\top| = |0| = \emptyset$$

Une preuve Π de $\vdash A_1, \dots, A_n$ dans la logique linéaire est traditionnellement interprétée par une relation

$$[\Pi] \subseteq |A_1| \times \dots \times |A_n|.$$

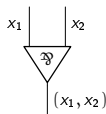
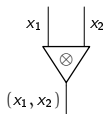
Localized Semantics / Standard relational semantics

Un réseau:



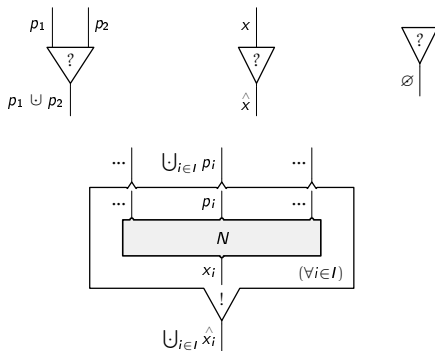
sera aussi interprété dans $|A_1| \times \dots \times |A_n|$.

Multiplicatifs:



Localized Semantics / Standard relational semantics

Exponentiels:



Localized Semantics / Exponential Trees

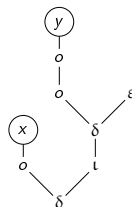
Definition

Soit Δ un ensemble, on appelle arborescences exponentielles les éléments de l'ensemble Δ^e défini par la syntaxe suivante:

$$\Delta^e ::= \varepsilon \mid \delta \Delta^e \Delta^e \mid o \Delta \mid \iota \Delta^{ee}$$

Par exemple, si $x, y \in \Delta$ alors l'expression suivante désigne une arborescence exponentielle de Δ^e :

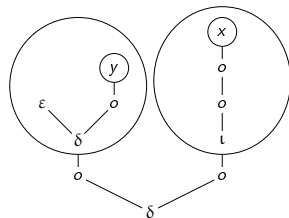
$$\delta(o x)(\iota(\delta(o(o y))\varepsilon))$$



Localized Semantics / Exponential Trees

On peut aussi construire des arborescences d'arborescences (c'est-à-dire des éléments de Δ^{ee}), par exemple:

$$\delta(o(\delta \varepsilon(o y))) (o(\iota(o(o x))))$$



Notamment, le constructeur ι transforme ces arborescences d'arborescences en de simples arborescences.

Localized Semantics / Exponential Trees

Definition (Contextes exponentiels)

Les arborescences de profondeur i d'un ensemble Δ sont les éléments de $\Delta^{(i)}$, qui est défini par:

$$\Delta^{(0)} = \Delta \qquad \Delta^{(i+1)} = \Delta^{(i)e}$$

On note Δ^* l'ensemble des arborescences de Δ de profondeurs arbitraires:

$$\Delta^* = \bigcup_{i \in \mathbb{N}} \Delta^{(i)}$$

Localized Semantics / Tree closures

To define a local semantics we will change the interpretation space:

- ▶ A localized net N with interface $\vdash A_1, \dots, A_n$ will be interpreted by a relation $[N] \subseteq |A_1|^* \times \dots \times |A_n|^*$.

Localized Semantics / Tree closures

To define a local semantics we will change the interpretation space:

- ▶ A localized net N with interface $\vdash A_1, \dots, A_n$ will be interpreted by a relation $[N] \subseteq |A_1|^* \times \dots \times |A_n|^*$.

The idea being that when we got some points:

- ▶ $(x_1, \dots, x_n) \in |A_1| \times \dots \times |A_n|$
- ▶ $(y_1, \dots, y_n) \in |A_1| \times \dots \times |A_n|$

in the semantics of some net, we want also:

- ▶ $(o x_1, \dots, o x_n) \in |A_1|^e \times \dots \times |A_n|^e$
- ▶ $(o y_1, \dots, o y_n) \in |A_1|^e \times \dots \times |A_n|^e$
- ▶ $(\delta(o x_1)(o y_1), \dots, \delta(o x_n)(o y_n)) \in |A_1|^e \times \dots \times |A_n|^e$

and so on...

Localized Semantics / Tree closures

To define a local semantics we will change the interpretation space:

- ▶ A localized net N with interface $\vdash A_1, \dots, A_n$ will be interpreted by a relation $[N] \subseteq |A_1|^* \times \dots \times |A_n|^*$.

The idea being that when we got some points:

- ▶ $(x_1, \dots, x_n) \in |A_1| \times \dots \times |A_n|$
- ▶ $(y_1, \dots, y_n) \in |A_1| \times \dots \times |A_n|$

in the semantics of some net, we want also:

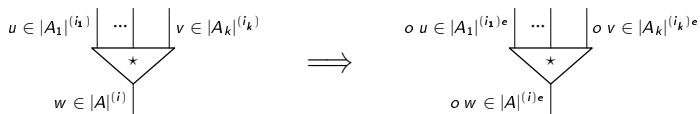
- ▶ $(o x_1, \dots, o x_n) \in |A_1|^e \times \dots \times |A_n|^e$
- ▶ $(o y_1, \dots, o y_n) \in |A_1|^e \times \dots \times |A_n|^e$
- ▶ $(\delta(o x_1)(o y_1), \dots, \delta(o x_n)(o y_n)) \in |A_1|^e \times \dots \times |A_n|^e$

and so on...

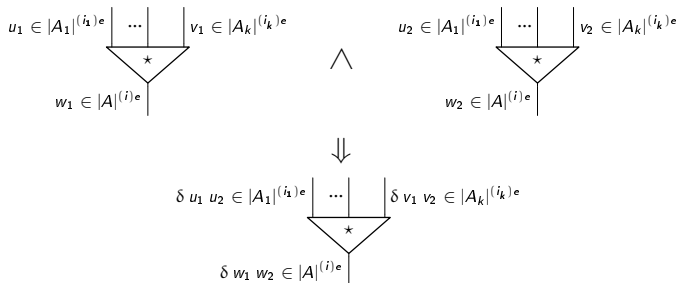
Localized Semantics / Tree closures

This global closure can be enforced locally on the semantics of each operator.

► o -closure:

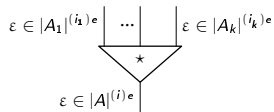


► δ -closure:



Localized Semantics / Tree closures

► ε -closure:



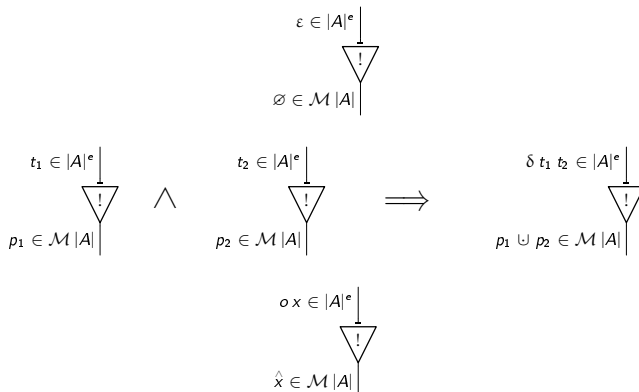
► ι -closure:



Localized Semantics / Without control channels

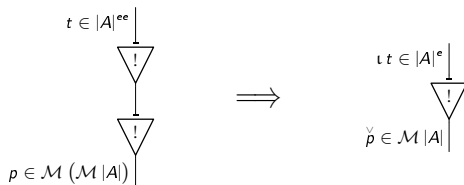
The relational semantics of each operator can now be defined. We add additional rules to generate their points.

Here's what we can do for the **active** system:

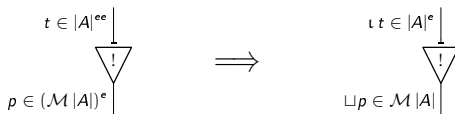


Localized Semantics / Without control channels

Le cas des contextes d'ingestion pourrait lui aussi être traité directement par une déduction de motif:



$\sqcup: (\mathcal{M} \Delta)^e \longrightarrow \mathcal{M} \Delta$: écrase par unions une arborescence en un simple multi-ensemble.



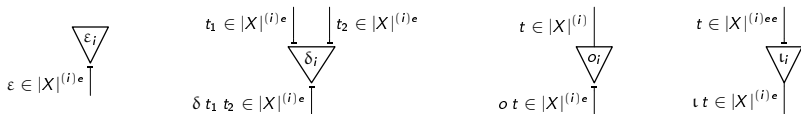
Localized Semantics / Without control channels

D'ailleurs, la sémantique des entrées de *promotion* correspond à cette pure opération:

$$\begin{array}{c}
 t \in (\mathcal{M} \mid V)^e \\
 \downarrow \\
 \begin{array}{c} \triangle \\ \vdots \end{array} \\
 \downarrow \\
 \sqcup t \in \mathcal{M} \mid V
 \end{array}$$

Localized Semantics / Without control channels

Pour les opérateurs de boîtes, on utilise les motifs suivants (ils dépendent de l'altitude de l'opérateur):



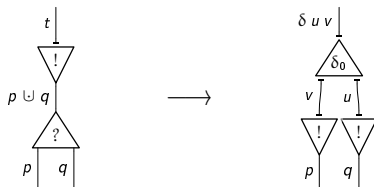
Localized Semantics / Without control channels

Lemma (Dynamique de la sémantique localisée)

La sémantique localisée d'un morceau de réseau (pas nécessairement valide) est affinée par réduction:

$$P \longrightarrow P' \implies [P']_a \subseteq [P]_a$$

Example:



Localized Semantics / Without control channels

Theorem

La sémantique localisée d'un réseau valide contient la clôture contextuelle de la sémantique standard de la preuve qui lui est associée.

$$\Pi \mapsto_a N \implies \overline{[\Pi]} \subseteq [N]_a$$

Localized Semantics / Without control channels

Theorem

La sémantique localisée d'un réseau valide contient la clôture contextuelle de la sémantique standard de la preuve qui lui est associée.

$$\Pi \mapsto_a N \implies \overline{[\Pi]} \subseteq [N]_a$$

On aimerait obtenir des égalités à la place des inclusions...

Localized Semantics / Without control channels

Theorem

La sémantique localisée d'un réseau valide contient la clôture contextuelle de la sémantique standard de la preuve qui lui est associée.

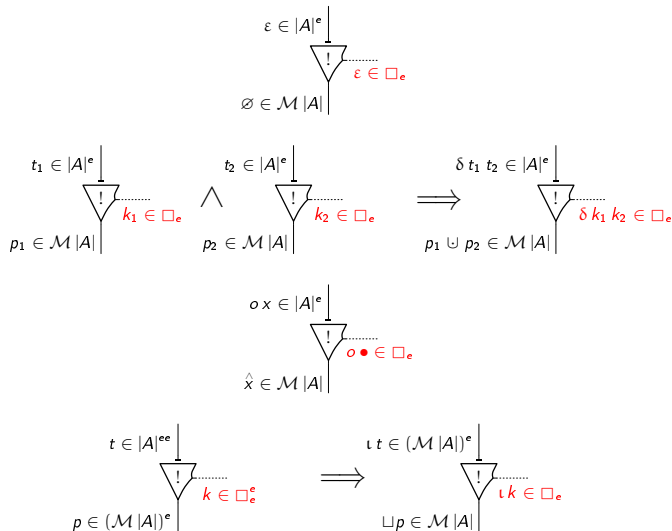
$$\Pi \mapsto_a N \implies \overline{[\Pi]} \subseteq [N]_a$$

On aimerait obtenir des égalités à la place des inclusions...

Regardons ce qu'il se passe lorsqu'on utilise des canaux de contrôle!
(Pas de différence entre le système passif et le système contrôlé)

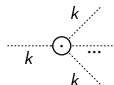
Localized Semantics / With control channels

On étend l'interprétation des types au type associé aux canaux de contrôle exponentiels: $\Box_e = \{\bullet\}^e$

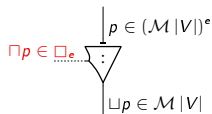


Localized Semantics / With control channels

Et pour les opérateurs de diffusion c'est simple:



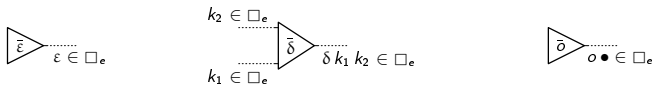
Pour les entrées ça marche de la même manière qu'avant:



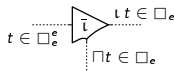
$\sqcap: \Delta^e \longrightarrow \{\bullet\}^e$: ne garde que la surface de l'arborescence donnée en argument.

Localized Semantics / With control channels

Les interprétations des opérateurs de boîte sont inchangés.
 Les interprétations des messages obéissent aux motifs suivants:



Pour les messages d'ingestion on utilisera:



Localized Semantics / With control channels

Lemma (Dynamique de la sémantique localisée)

La sémantique localisée d'un morceau de réseau (pas nécessairement valide) est affinée par réduction:

$$P \longrightarrow P' \implies [P']_c \subseteq [P]_c$$

Localized Semantics / With control channels

Theorem

La sémantique localisée d'un réseau valide du système contrôlé est égale à la clôture contextuelle de la sémantique standard de la preuve qui lui est associée.

$$\Pi \mapsto_c N \implies \overline{[\Pi]} = [N]_c$$

Corollary (Dynamique de la sémantique localisée d'un réseau valide)

La sémantique localisée d'un réseau valide (du système passif ou contrôlé) est préservée par réduction:

$$N \longrightarrow N' \implies [N']_c = [N]_c$$

Introduction

Localized Proof-nets

Localized Semantics

Conclusion

What do we have?

What next?

Conclusion / What do we have?

Quelques systèmes localisés (actifs, passif, contrôlé):

- ▶ pour le fragment exponentiel
- ▶ pour le fragment additif
- ▶ pour un fragment récursif

Avantages: exécution simple, stratégies de calculs intéressantes.

Une sémantique adaptée à ces systèmes:

- ▶ pour le fragment exponentiel
- ▶ pour le fragment additif
- ▶ pour un fragment récursif?

Intérêt: sémantique localisée

Contre-partie: manipulation de gros objets (arborescences exponentielles)

Conclusion / What next?

Travail en cours sur la localisation:

- ▶ Un moyen de “séquentialiser” les réseaux localisés grâce au calcul des structures.
- ▶ Appliquer la méthode de localisation à la *super-promotion* dans le but d'améliorer sa dynamique.

D'autres choses à faire:

- ▶ Préciser le sens de l'optimalité parallèle.
- ▶ Vérifier que la sémantique localisée garde ses propriétés en présence de récursion.

Et peut-être:

- ▶ Améliorer l'implémentation des réseaux d'interaction (Nœuds multi-ports, Utilisation de GPUs).
- ▶ Faire un compilateur *Mini-ML* $\mapsto$ *Réseaux*.



Stefano Guerrini, Simone Martini, and Andrea Masini.

Coherence for sharing proof-nets.

Theoretical Computer Science, 294(3):379–409, 2003.



G rard P. Huet.

Confluent reductions: Abstract properties and applications to term rewriting systems.

J. ACM, 27(4):797–821, 1980.



Yves Lafont.

Interaction nets.

Principles of Programming Languages (POPL '90), pages 95–108, 1990.



Yves Lafont.

Interaction combinators.

Information and Computation, 137(1):69–101, 1997.



John Lamping.

An algorithm for optimal lambda calculus reduction.

Principles of Programming Languages (POPL '90), pages 16–30, 1990.



Damiano Mazza.

A denotational semantics for the symmetric interaction combinators.

Mathematical Structures in Computer Science, 17(3), 2007.