The Geometry of Interaction of Differential Interaction Nets

choco

Marc de Falco

Institut de Mathématiques de Luminy

Logic in Computer Science 08

Outline

We study

- differential interaction nets (din) : extension of linear logic [Ehrhard and Regnier, 2005], presented as formal sums of graph-like structures and rewriting, encoding resource λ-calculus and a finitary π-calculus
- geometry of interaction (GoI) : a special kind of semantics accounting for reduction, akin to game semantics, defined on fragments of linear logic [Girard, 1989],[Girard, 1995]

We extend the path based version of GoI [Danos and Regnier, 1995], i.e. we

- define a proper notion of paths
- define a proper equational theory encoding reduction in a local and asynchronous way
- prove that the theory is coherent by giving a realisation
- prove that our encoding of path reduction is sound

Outline

We study

- differential interaction nets (din) : extension of linear logic [Ehrhard and Regnier, 2005], presented as formal sums of graph-like structures and rewriting, encoding resource λ-calculus and a finitary π-calculus
- geometry of interaction (GoI) : a special kind of semantics accounting for reduction, akin to game semantics, defined on fragments of linear logic [Girard, 1989],[Girard, 1995]

We extend the path based version of GoI [Danos and Regnier, 1995], i.e. we

- define a proper notion of paths
- define a proper equational theory encoding reduction in a local and asynchronous way
- prove that the theory is coherent by giving a realisation
- prove that our encoding of path reduction is sound

Outline

We study

- differential interaction nets (din) : extension of linear logic [Ehrhard and Regnier, 2005], presented as formal sums of graph-like structures and rewriting, encoding resource λ-calculus and a finitary π-calculus
- geometry of interaction (GoI) : a special kind of semantics accounting for reduction, akin to game semantics, defined on fragments of linear logic [Girard, 1989],[Girard, 1995]

We extend the path based version of GoI [Danos and Regnier, 1995], i.e. we

- define a proper notion of paths
- define a proper equational theory encoding reduction in a local and asynchronous way
- prove that the theory is coherent by giving a realisation
- prove that our encoding of path reduction is sound

Outline

We study

- differential interaction nets (din) : extension of linear logic [Ehrhard and Regnier, 2005], presented as formal sums of graph-like structures and rewriting, encoding resource λ-calculus and a finitary π-calculus
- geometry of interaction (GoI) : a special kind of semantics accounting for reduction, akin to game semantics, defined on fragments of linear logic [Girard, 1989],[Girard, 1995]

We extend the path based version of GoI [Danos and Regnier, 1995], i.e. we

- define a proper notion of paths
- define a proper equational theory encoding reduction in a local and asynchronous way
- prove that the theory is coherent by giving a realisation
- prove that our encoding of path reduction is sound

Outline

We study

- differential interaction nets (din) : extension of linear logic [Ehrhard and Regnier, 2005], presented as formal sums of graph-like structures and rewriting, encoding resource λ-calculus and a finitary π-calculus
- geometry of interaction (GoI) : a special kind of semantics accounting for reduction, akin to game semantics, defined on fragments of linear logic [Girard, 1989],[Girard, 1995]

We extend the path based version of GoI [Danos and Regnier, 1995], i.e. we

- define a proper notion of paths
- define a proper equational theory encoding reduction in a local and asynchronous way
- prove that the theory is coherent by giving a realisation
- prove that our encoding of path reduction is sound

Outline

We study

- differential interaction nets (din) : extension of linear logic [Ehrhard and Regnier, 2005], presented as formal sums of graph-like structures and rewriting, encoding resource λ-calculus and a finitary π-calculus
- geometry of interaction (GoI) : a special kind of semantics accounting for reduction, akin to game semantics, defined on fragments of linear logic [Girard, 1989],[Girard, 1995]

We extend the path based version of GoI [Danos and Regnier, 1995], i.e. we

- define a proper notion of paths
- define a proper equational theory encoding reduction in a local and asynchronous way
- prove that the theory is coherent by giving a realisation
- prove that our encoding of path reduction is sound

Outline

We study

- differential interaction nets (din) : extension of linear logic [Ehrhard and Regnier, 2005], presented as formal sums of graph-like structures and rewriting, encoding resource λ-calculus and a finitary π-calculus
- geometry of interaction (GoI) : a special kind of semantics accounting for reduction, akin to game semantics, defined on fragments of linear logic [Girard, 1989],[Girard, 1995]

We extend the path based version of GoI [Danos and Regnier, 1995], i.e. we

- define a proper notion of paths
- define a proper equational theory encoding reduction in a local and asynchronous way
- prove that the theory is coherent by giving a realisation
- prove that our encoding of path reduction is sound

Linear Logic can be seen as an explicit substitution system for λ-calculus

data is split between

offers : *arguments of application*
provided as a factory producing exact copies of the same object

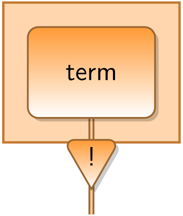
demands : *occurrences of variables*
organized as a tree of demands

Mass production issues: *non personalized offer, not fault-tolerant, . . .*
Can we replace it with craftsmanship?

Linear Logic can be seen as an explicit substitution system for λ-calculus

data is split between

offers : *arguments of application* provided as a factory producing exact copies of the same object



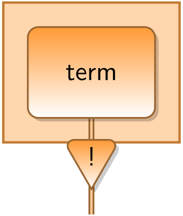
demands : *occurrences of variables* organized as a tree of demands

Mass production issues: *non personalized offer, not fault-tolerant, . . .*
Can we replace it with craftsmanship?

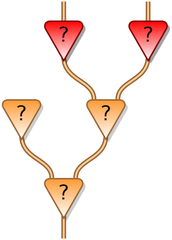
Linear Logic can be seen as an explicit substitution system for λ-calculus

data is split between

offers : *arguments of application*
provided as a factory producing exact copies of the same object



demands : *occurrences of variables*
organized as a tree of demands



Mass production issues: *non personalized offer, not fault-tolerant, . . .*
Can we replace it with craftsmanship?

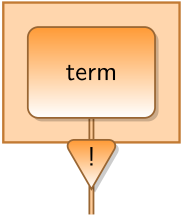
Linear Logic

Linear Logic from a calculus point of view

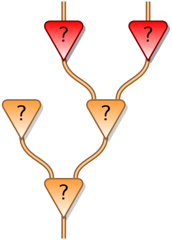
Linear Logic can be seen as an explicit substitution system for λ-calculus

data is split between

offers : *arguments of application* provided as a factory producing exact copies of the same object



demands : *occurrences of variables* organized as a tree of demands



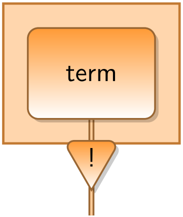
Mass production issues: *non personalized offer, not fault-tolerant, . . .*

Can we replace it with craftsmanship?

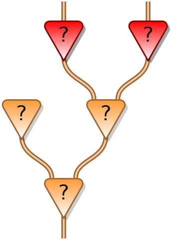
Linear Logic can be seen as an explicit substitution system for λ-calculus

data is split between

offers : *arguments of application*
provided as a factory producing exact copies of the same object



demands : *occurrences of variables*
organized as a tree of demands

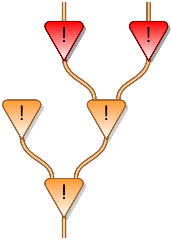


Mass production issues: *non personalized offer, not fault-tolerant, . . .*
Can we replace it with craftsmanship?

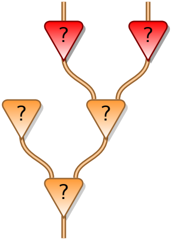
Differential Linear Logic can be seen as an explicit substitution system for **resource** λ-calculus

data is split between

offers : *arguments of application*
organized as a tree of offers



demands : *occurrences of variables*
organized as a tree of demands



Differential Interaction Nets

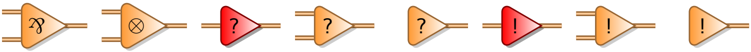
- the natural presentation of differential linear logic akin to proof-net of linear logic
- a special kind of interaction nets using the cells

- with formal sums

and 0

Differential Interaction Nets

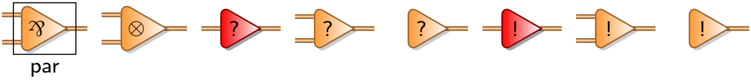
- the natural presentation of differential linear logic akin to proof-net of linear logic
- a special kind of interaction nets using the cells



- with formal sums

Differential Interaction Nets

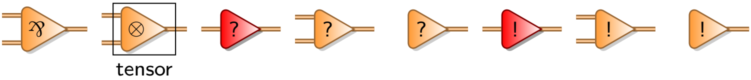
- the natural presentation of differential linear logic akin to proof-net of linear logic
- a special kind of interaction nets using the cells



- with formal sums

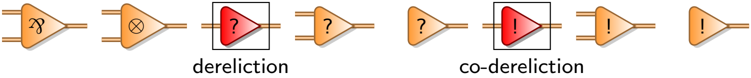
Differential Interaction Nets

- the natural presentation of differential linear logic akin to proof-net of linear logic
- a special kind of interaction nets using the cells



- with formal sums

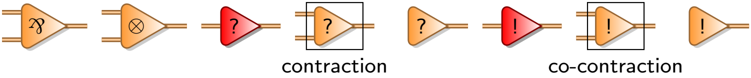
- the natural presentation of differential linear logic akin to proof-net of linear logic
- a special kind of interaction nets using the cells



- with formal sums

Differential Interaction Nets

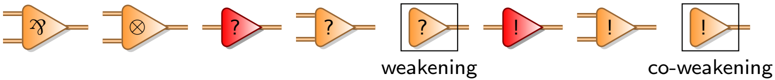
- the natural presentation of differential linear logic akin to proof-net of linear logic
- a special kind of interaction nets using the cells



- with formal sums

Differential Interaction Nets

- the natural presentation of differential linear logic akin to proof-net of linear logic
- a special kind of interaction nets using the cells

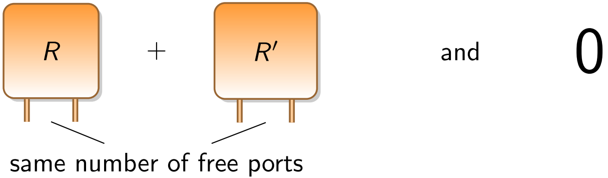


- with formal sums

Differential Interaction Nets

- the natural presentation of differential linear logic akin to proof-net of linear logic
- a special kind of interaction nets using the cells

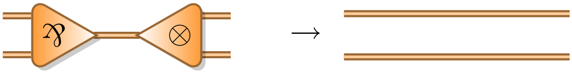
- with formal sums



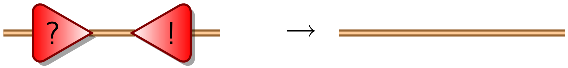
- Dynamics over dins expressed by means of interaction rewriting rules
- presented in the *economy* settings but possible in both network setting (hence π-calculus) and mathematical setting (hence *differential*)
- $⅋/\otimes$: *synchronisation of two trades*

- Dynamics over dins expressed by means of interaction rewriting rules
- presented in the *economy* settings but possible in both network setting (hence π-calculus) and mathematical setting (hence *differential*)
- $\parr/\otimes$: *synchronisation of two trades*

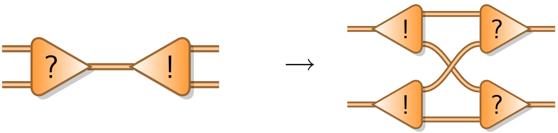
- Dynamics over dins expressed by means of interaction rewriting rules
- presented in the *economy* settings but possible in both network setting (hence π-calculus) and mathematical setting (hence *differential*)
- $\parr/\otimes$: *synchronisation of two trades*



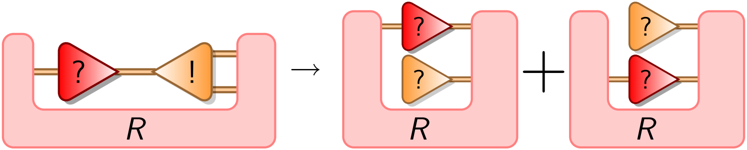
- ?/! rules: we only present half of them, others obtained by duality
- dereliction/co-dereliction : *offer meets demand*



- contraction/co-contraction : *independent routing*



dereliction/co-contraction : *one demand for a binary offer*

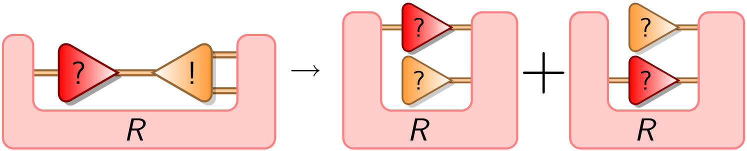


duplication of R: *global reduction rule*

Goal of a GoI

Can we replace this global reduction with a lot of local sum propagations?

dereliction/co-contraction : *one demand for a binary offer*

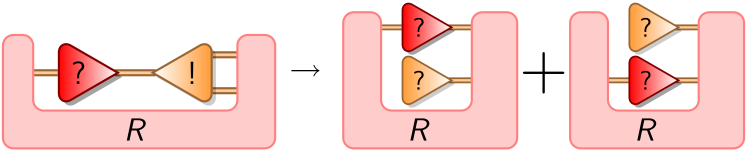


duplication of R: *global reduction rule*

Goal of a GoI

Can we replace this global reduction with a lot of local sum propagations?

dereliction/co-contraction : *one demand for a binary offer*



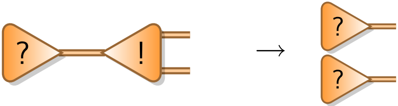
duplication of R: *global reduction rule*

Goal of a GoI

Can we replace this global reduction with a lot of local sum propagations?

dereliction/co-weakening : *a demand filled by a void offer crisis*

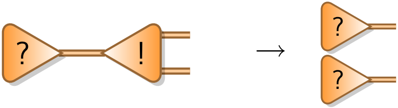
weakening/co-contraction : *routing a void demand*



weakening/co-weakening : *void offer meets void demand*

We do not consider these rules here. We work on weak-reduction. (usual restriction used by GoI) dereliction/co-weakening : *a demand filled by a void offer crisis*

weakening/co-contraction : *routing a void demand*

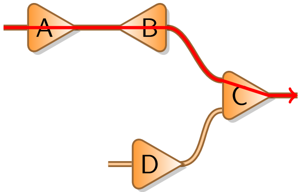


weakening/co-weakening : *void offer meets void demand*

We do not consider these rules here. We work on weak-reduction. (usual restriction used by GoI)

Paths in interaction nets

- standard notion of path coming from the graph-like structure



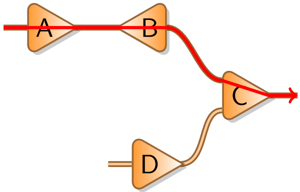
- a reduction $R \longrightarrow R'$ extends to a reduction from $\mathcal{P}(R) \longrightarrow \mathcal{P}(R')^\star$
- a path can be destroyed

- *persistent path*: a path not destroyed by any chain of reduction
- *GoI goal*: find a structure S and a morphism $w : \mathcal{P}(R) \longrightarrow S$ such that

$$\varphi \text{ persistent} \iff w(\varphi) \neq 0$$

Paths in interaction nets

- standard notion of path coming from the graph-like structure



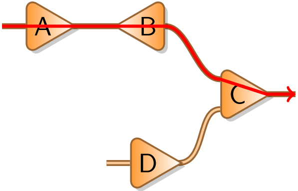
- a reduction $R \longrightarrow R'$ extends to a reduction from $\mathcal{P}(R) \longrightarrow \mathcal{P}(R')^{\star}$
- a path can be destroyed

- *persistent path*: a path not destroyed by any chain of reduction
- *GoI goal*: find a structure S and a morphism $w : \mathcal{P}(R) \longrightarrow S$ such that

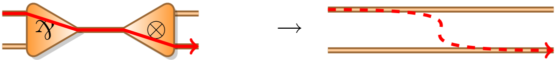
$$\varphi \text{ persistent} \iff w(\varphi) \neq 0$$

Paths in interaction nets

- standard notion of path coming from the graph-like structure



- a reduction $R \to R'$ extends to a reduction from $\mathcal{P}(R) \to \mathcal{P}(R')^\star$
- a path can be destroyed

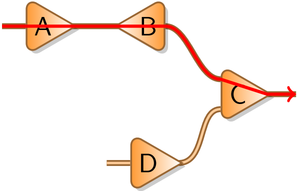


- *persistent path*: a path not destroyed by any chain of reduction
- *GoI goal*: find a structure S and a morphism $w : \mathcal{P}(R) \to S$ such that

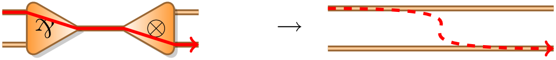
$$\varphi \text{ persistent} \iff w(\varphi) \neq 0$$

Paths in interaction nets

- standard notion of path coming from the graph-like structure



- a reduction $R \to R'$ extends to a reduction from $\mathcal{P}(R) \to \mathcal{P}(R')^\star$
- a path can be destroyed

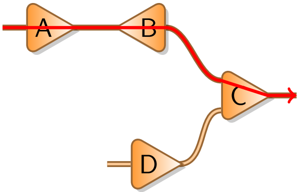


- *persistent path*: a path not destroyed by any chain of reduction
- *GoI goal*: find a structure S and a morphism $w : \mathcal{P}(R) \to S$ such that

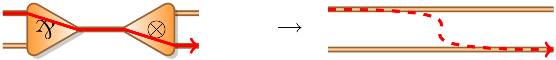
$$\varphi \text{ persistent} \iff w(\varphi) \neq 0$$

Paths in interaction nets

- standard notion of path coming from the graph-like structure



- a reduction $R \to R'$ extends to a reduction from $\mathcal{P}(R) \to \mathcal{P}(R')^\star$
- a path can be destroyed



- *persistent path*: a path not destroyed by any chain of reduction
- *GoI goal*: find a structure S and a morphism $w : \mathcal{P}(R) \to S$ such that

$$\varphi \text{ persistent} \iff w(\varphi) \neq 0$$

Paths in sums

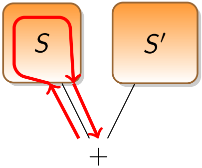
- What is a path in $R + R'$? How can we distinguish a path in one the Rs of $R + R$?
- We need to fix an orientation: we consider sums are purely syntactical, i.e. as *trees*
- a path in a simple net is prefixed and suffixed by the branch of the tree to get a path in a net

Paths in sums

- What is a path in $R + R'$? How can we distinguish a path in one the Rs of $R + R$?
- We need to fix an orientation: we consider sums are purely syntactical, i.e. as *trees*
- a path in a simple net is prefixed and suffixed by the branch of the tree to get a path in a net

Paths in sums

- What is a path in $R + R'$? How can we distinguish a path in one the Rs of $R + R$?
- We need to fix an orientation: we consider sums are purely syntactical, i.e. as *trees*
- a path in a simple net is prefixed and suffixed by the branch of the tree to get a path in a net



Losing confluency

Everything can now be defined as intended

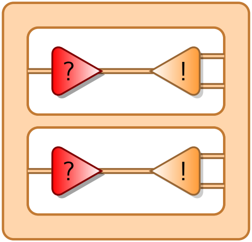
the reduction is just no longer confluent. . .

Losing confluency

Everything can now be defined as intended
the reduction is just no longer confluent. . .

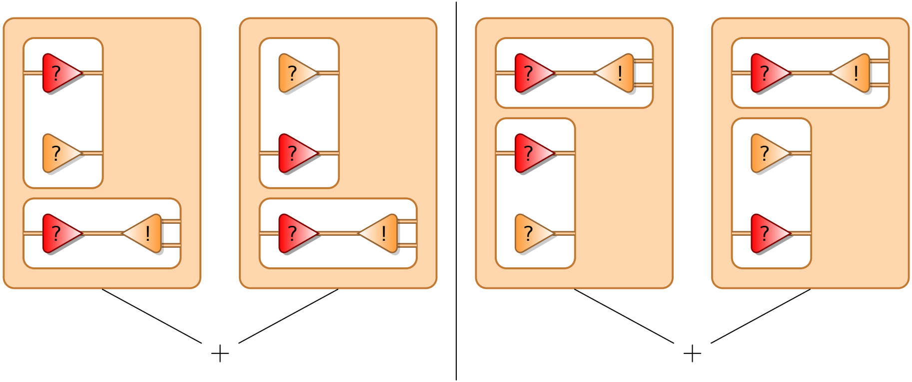
Losing confluency

Everything can now be defined as intended
the reduction is just no longer confluent...



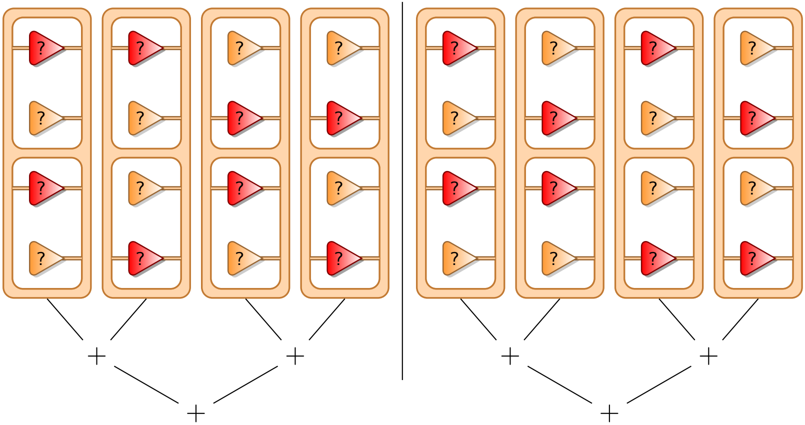
Losing confluency

Everything can now be defined as intended
the reduction is just no longer confluent. . .



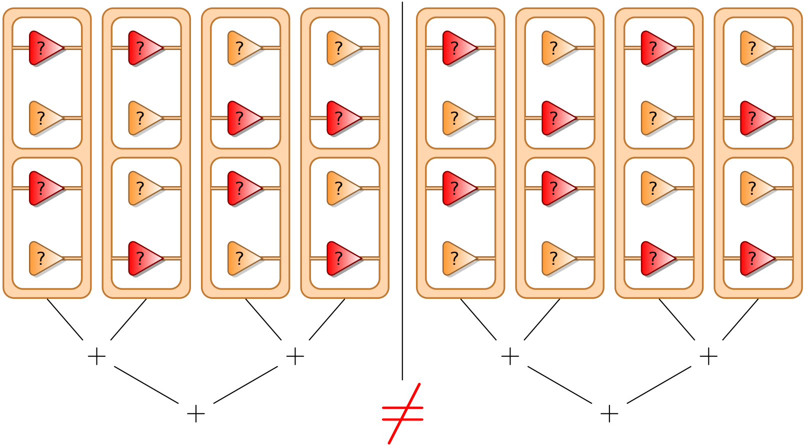
Losing confluency

Everything can now be defined as intended
the reduction is just no longer confluent. . .



Losing confluency

Everything can now be defined as intended
the reduction is just no longer confluent. . .



Names

We add names to track down the context of sum production

- we give a unique name to each (co)dereliction in a simple net, and we label the nodes of the tree
- we replace the sum producing rules by

- we add middle-four interchange law between $+_\alpha$ and $+_\beta$ for $\alpha \neq \beta$:

$$(R_1 +_\alpha R_2) +_\beta (R_3 +_\alpha R_4) \equiv (R_1 +_\beta R_3) +_\alpha (R_2 +_\beta R_4)$$

Names

We add names to track down the context of sum production

- we give a unique name to each (co)dereliction in a simple net, and we label the nodes of the tree
- we replace the sum producing rules by

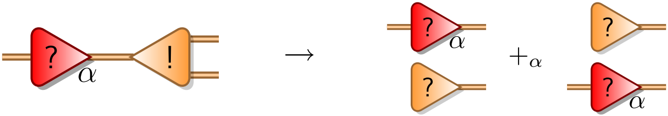
- we add middle-four interchange law between $+_\alpha$ and $+_\beta$ for $\alpha \neq \beta$:

$$(R_1 +_\alpha R_2) +_\beta (R_3 +_\alpha R_4) \equiv (R_1 +_\beta R_3) +_\alpha (R_2 +_\beta R_4)$$

Names

We add names to track down the context of sum production

- we give a unique name to each (co)dereliction in a simple net, and we label the nodes of the tree
- we replace the sum producing rules by



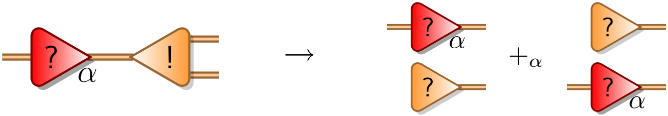
- we add middle-four interchange law between $+_\alpha$ and $+_\beta$ for $\alpha \neq \beta$:

$$(R_1 +_\alpha R_2) +_\beta (R_3 +_\alpha R_4) \equiv (R_1 +_\beta R_3) +_\alpha (R_2 +_\beta R_4)$$

Names

We add names to track down the context of sum production

- we give a unique name to each (co)dereliction in a simple net, and we label the nodes of the tree
- we replace the sum producing rules by



- we add middle-four interchange law between $+_\alpha$ and $+_\beta$ for $\alpha \neq \beta$:

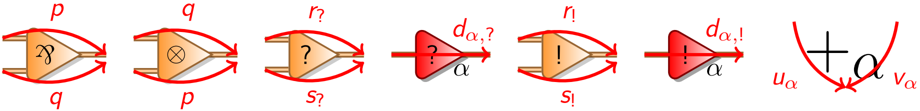
$$(R_1 +_\alpha R_2) +_\beta (R_3 +_\alpha R_4) \equiv (R_1 +_\beta R_3) +_\alpha (R_2 +_\beta R_4)$$

we construct an inverse monoid with zero $\partial L^\star$ and a weighting of path w with

$$w(\varphi\varphi') = w(\varphi')w(\varphi) \quad w(\rightarrow) = w(\leftarrow)^\star$$

$\partial L^\star$

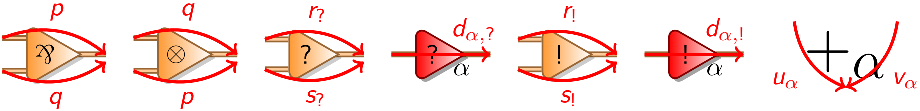
we weight cell traversals with generators:



and relations :

- (p, q) and (u_α, v_α) like MLL : $p^\star p = 1, q^\star p = 0, \ldots$
- $r_!, r_?, s_!, s_?$ have bigebras relations : $r_!^\star s_? = s_? r_!^\star, \ldots$
- $d^\star_{\alpha,!} d_{\beta,?}$
- $u_\alpha, v_\alpha, e_\alpha$ commutes with everything non-α
- $d^\star_{\alpha,t} s_{t'} = u_\alpha d^\star_{\alpha,t} u^\star_\alpha$, $d^\star_{\alpha,t} r_{t'} = v_\alpha d^\star_{\alpha,t} v^\star_\alpha$ where $t \neq t'$, $\alpha \neq \beta$

we weight cell traversals with generators:

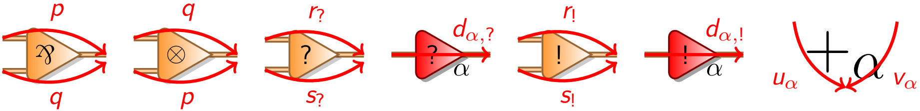


and relations :

- (p, q) and (u_α, v_α) like MLL : $p^\star p = 1, q^\star p = 0, \ldots$
- $r_!, r_?, s_!, s_?$ have bigebras relations : $r_!^\star s_? = s_? r_!^\star, \ldots$
- $d_{\alpha,!}^\star d_{\beta,?}$
- $u_\alpha, v_\alpha, e_\alpha$ commutes with everything non-α
- $d_{\alpha,t}^\star s_{t'} = u_\alpha d_{\alpha,t}^\star u_\alpha^\star$, $d_{\alpha,t}^\star r_{t'} = v_\alpha d_{\alpha,t}^\star v_\alpha^\star$ where $t \neq t'$, $\alpha \neq \beta$

$\partial L^\star$

we weight cell traversals with generators:

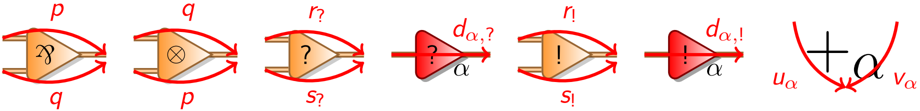


and relations :

- (p, q) and (u_α, v_α) like MLL : $p^\star p = 1, q^\star p = 0, \dots$
- $r_!, r_?, s_!, s_?$ have bigebras relations : $r_!^\star s_? = s_? r_!^\star, \dots$
- $d^\star_{\alpha,!} d_{\beta,?}$
- $u_\alpha, v_\alpha, e_\alpha$ commutes with everything non-α
- $d^\star_{\alpha,t} s_{t'} = u_\alpha d^\star_{\alpha,t} u^\star_\alpha$, $d^\star_{\alpha,t} r_{t'} = v_\alpha d^\star_{\alpha,t} v^\star_\alpha$ where $t \neq t'$, $\alpha \neq \beta$

$\partial L^\star$

we weight cell traversals with generators:

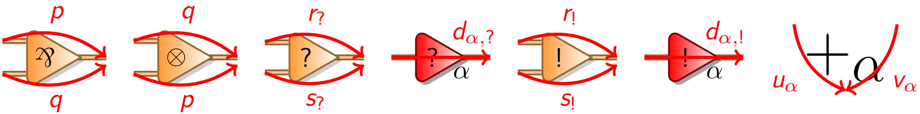


and relations :

- (p, q) and (u_α, v_α) like MLL : $p^\star p = 1, q^\star p = 0, \ldots$
- $r_!, r_?, s_!, s_?$ have bigebras relations : $r_!^\star s_? = s_? r_!^\star, \ldots$
- $d^\star_{\alpha,!} d_{\beta,?} = 1$
- $u_\alpha, v_\alpha, e_\alpha$ commutes with everything non-α
- $d^\star_{\alpha,t} s_{t'} = u_\alpha d^\star_{\alpha,t} u^\star_\alpha$, $d^\star_{\alpha,t} r_{t'} = v_\alpha d^\star_{\alpha,t} v^\star_\alpha$ where $t \neq t'$, $\alpha \neq \beta$

$\partial L^\star$

we weight cell traversals with generators:

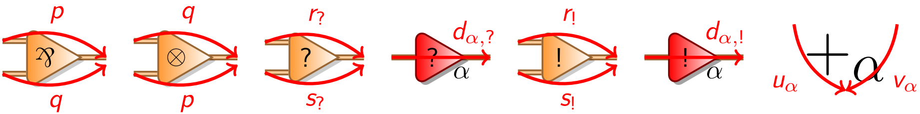


and relations :

- (p, q) and (u_α, v_α) like MLL : $p^\star p = 1, q^\star p = 0, \ldots$
- $r_!, r_?, s_!, s_?$ have bigebras relations : $r_!^\star s_? = s_? r_!^\star, \ldots$
- $d^\star_{\alpha,!} d_{\beta,?} = 1$ *everything collapses*
- $u_\alpha, v_\alpha, e_\alpha$ commutes with everything non-α
- $d^\star_{\alpha,t} s_{t'} = u_\alpha d^\star_{\alpha,t} u^\star_\alpha$, $d^\star_{\alpha,t} r_{t'} = v_\alpha d^\star_{\alpha,t} v^\star_\alpha$ where $t \neq t'$, $\alpha \neq \beta$

$\partial L^\star$

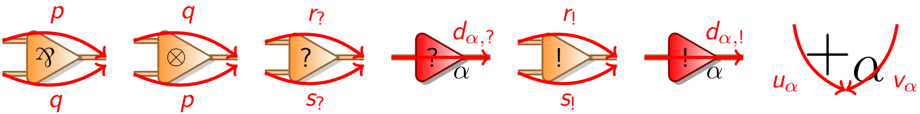
we weight cell traversals with generators:



and relations :

- (p, q) and (u_α, v_α) like MLL : $p^\star p = 1, q^\star p = 0, \ldots$
- $r_!, r_?, s_!, s_?$ have bigebras relations : $r_!^\star s_? = s_? r_!^\star, \ldots$
- $d^\star_{\alpha,!} d_{\beta,?} \quad = e_\alpha e_\beta$ special generators: e_α *means no alpha*
- $u_\alpha, v_\alpha, e_\alpha$ commutes with everything non-α
- $d^\star_{\alpha,t} s_{t'} = u_\alpha d^\star_{\alpha,t} u^\star_\alpha$, $d^\star_{\alpha,t} r_{t'} = v_\alpha d^\star_{\alpha,t} v^\star_\alpha$ where $t \neq t'$, $\alpha \neq \beta$

we weight cell traversals with generators:

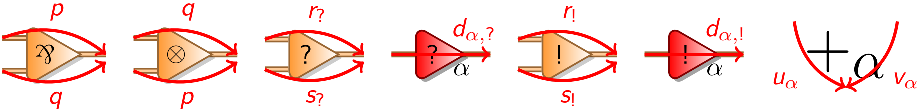


and relations :

- (p, q) and (u_α, v_α) like MLL : $p^\star p = 1, q^\star p = 0, \ldots$
- $r_!, r_?, s_!, s_?$ have bigebras relations : $r_!^\star s_? = s_? r_!^\star, \ldots$
- $d_{\alpha,!}^\star d_{\beta,?} \quad = e_\alpha e_\beta$ special generators: e_α *means no alpha*
- $u_\alpha, v_\alpha, e_\alpha$ commutes with everything non-α
- $d_{\alpha,t}^\star s_{t'} = u_\alpha d_{\alpha,t}^\star u_\alpha^\star$, $d_{\alpha,t}^\star r_{t'} = v_\alpha d_{\alpha,t}^\star v_\alpha^\star$ where $t \neq t'$, $\alpha \neq \beta$

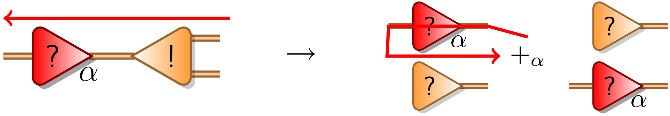
$\partial L^\star$

we weight cell traversals with generators:



and relations :

- (p, q) and (u_α, v_α) like MLL : $p^\star p = 1, q^\star p = 0, \ldots$
- $r_!, r_?, s_!, s_?$ have bigebras relations : $r_!^\star s_? = s_? r_!^\star, \ldots$
- $d^\star_{\alpha,!} d_{\beta,?} \quad = e_\alpha e_\beta$ special generators: e_α *means no alpha*
- $u_\alpha, v_\alpha, e_\alpha$ commutes with everything non-α
- $d^\star_{\alpha,t} s_{t'} = u_\alpha d^\star_{\alpha,t} u^\star_\alpha$, $d^\star_{\alpha,t} r_{t'} = v_\alpha d^\star_{\alpha,t} v^\star_\alpha$ where $t \neq t'$, $\alpha \neq \beta$



Fact

$\partial L^\star$ is non-trivial: $0 \neq 1$

Proof.

by constructing a non-trivial realization as a operations on concrete objects: *tokens* made of stacks,. . . □

Theorem

φ is weakly-persistent $\iff$ $w(\varphi) \neq 0$

normalizing factors:

$$n_\varphi(\varphi') = \prod_{\alpha \in \varphi, \alpha \notin \varphi'} e_\alpha$$

we have equality along weak-reduction up to normalizing factors:

Lemma (fundamental lemma)

$R \to R'$ weakly, $\varphi \in \mathcal{P}(R)$ deformed by reduction

- *either $\varphi \to \varphi'$ and $w(\varphi) = n_\varphi(\varphi')w(\varphi')$*
- *or φ destroyed and $w(\varphi) = 0$*

Weights structure

We recover the Danos-Regnier $ab^\star$ theorem of GoI:

Theorem ($\alpha ab^\star \alpha^\star$)

$w(\varphi) = 0$ *or* $\exists \alpha \in \partial {L^\star}^+_a, a, b \in \partial {L^\star}^+_{me}$ *and* $w(\varphi) = \alpha ab^\star \alpha^\star$

where we distinguish two monoids:

- $\partial {L^\star}^+_{me}$ generated by $\{p, q, r, s, d\}$
- $\partial {L^\star}^+_a$ generated by $\{u, v, e\}$

Normalized Execution

We consider formal sums of weights and set

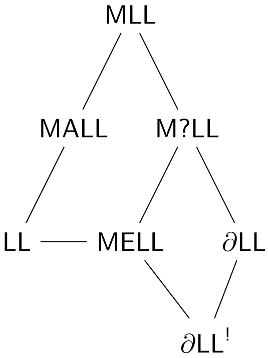
$$\mathrm{NEX}_{R_0}(R) = \sum_{\varphi \in R} n_{R_0}(\varphi) w(\varphi)$$

Theorem

NEX_{R_0} is an invariant of weak reductions starting from R_0

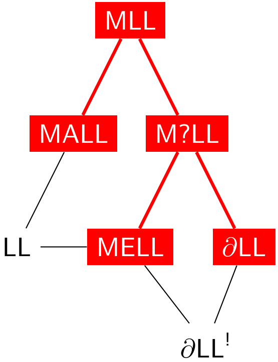
Expressive power of the equational theory

$\partial L^\star$ is rich enough to encode more than ∂LL



Expressive power of the equational theory

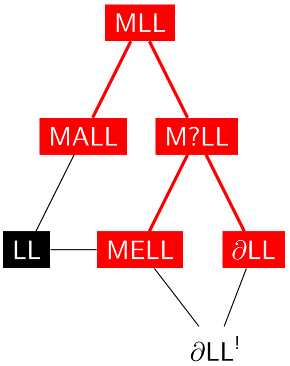
$\partial L^\star$ is rich enough to encode more than ∂LL



for MELL we have to extend a bit $\partial L^\star$, but the extension is also non-trivial

Expressive power of the equational theory

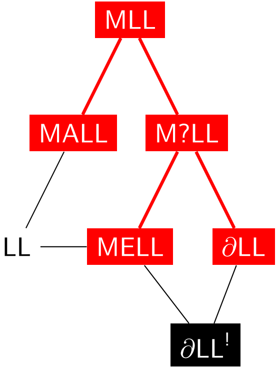
$\partial L^{\star}$ is rich enough to encode more than ∂LL



out of reach because of the interactions between exponentials and additives

Expressive power of the equational theory

$\partial L^\star$ is rich enough to encode more than ∂LL



out of reach because of the strong use of commutativity of links [Tranquilli, 2007]

Future works

- Non-deterministic computation
  - differential interaction nets can encode a finitary π-calculus, thus giving a notion of paths and GoI for this calculus
  - the notion of sub-tree can be made compatible with the sum interchange law: *slices*
  - the weight of slices is a lattice, thus, we can express properties of non-determinism as computations of lower and upper bounds
- sharing graphs and readback process for dins
- AJM-style game semantics extracted from the GoI
- try to extend the GoI to $\partial \mathrm{LL}^!$, maybe by means of *variants* to cope with commutativity issues

Future works

- Non-deterministic computation
  - differential interaction nets can encode a finitary π-calculus, thus giving a notion of paths and GoI for this calculus
  - the notion of sub-tree can be made compatible with the sum interchange law: *slices*
  - the weight of slices is a lattice, thus, we can express properties of non-determinism as computations of lower and upper bounds
- sharing graphs and readback process for dins
- AJM-style game semantics extracted from the GoI
- try to extend the GoI to $\partial LL^!$, maybe by means of *variants* to cope with commutativity issues

Future works

- Non-deterministic computation
  - differential interaction nets can encode a finitary π-calculus, thus giving a notion of paths and GoI for this calculus
  - the notion of sub-tree can be made compatible with the sum interchange law: *slices*
  - the weight of slices is a lattice, thus, we can express properties of non-determinism as computations of lower and upper bounds
- sharing graphs and readback process for dins
- AJM-style game semantics extracted from the GoI
- try to extend the GoI to $\partial \mathrm{LL}^!$, maybe by means of *variants* to cope with commutativity issues

Future works

- Non-deterministic computation
  - differential interaction nets can encode a finitary π-calculus, thus giving a notion of paths and GoI for this calculus
  - the notion of sub-tree can be made compatible with the sum interchange law: *slices*
  - the weight of slices is a lattice, thus, we can express properties of non-determinism as computations of lower and upper bounds
- sharing graphs and readback process for dins
- AJM-style game semantics extracted from the GoI
- try to extend the GoI to $\partial \mathrm{LL}^!$, maybe by means of *variants* to cope with commutativity issues

Questions

beamericonarticle Danos, V. and Regnier, L. (1995).
Proof-nets and the Hilbert space.
In [Girard et al., 1995].

beamericonarticle Ehrhard, T. and Regnier, L. (2005).
Uniformity and the Taylor expansion of ordinary lambda-terms.
Accepté à Theoretical Computer Science, en cours de révision.

beamericonarticle Girard, J.-Y. (1989).
Geometry of interaction I: an interpretation of system F.
In Ferro, Bonotto, V. and Zanardo, editors, *Proceedings of the Logic Colloquium 88*, pages 221–260, Padova. North-Holland.

beamericonarticle Girard, J.-Y. (1995).
Geometry of interaction III: acommodating the additives.
In [Girard et al., 1995].

beamericonarticle Girard, J.-Y., Lafont, Y., and Regnier, L., editors (1995).
Advances in Linear Logic, volume 222 of *London Mathematical Society Lecture Note Series*. Cambridge University Press.