Embedding the Finitary Pi-Calculus in Differential Interaction Nets

Thomas Ehrhard and Olivier Laurent

Preuves Programmes Systèmes
CNRS – Université Paris 7
[Thomas.Ehrhard|Olivier.Laurent]@pps.jussieu.fr

Abstract. We propose a translation of a finitary (that is, replication-free) version of the monadic localised pi-calculus into the purely exponential part of promotion-free differential interaction nets. This embedding is a simulation of reduction.

Since the introduction of Linear Logic by Girard in 1986, it was clear to many logicians and computer scientists that some deep connection between this new logical setting and concurrency should show up. This impression has been enforced by the introduction of *interaction nets* by Lafont [1], where reduction is given by a purely local and asynchronous interaction.

There is an apparent contradiction between non-determinism and the Curry-Howard approach to computation. Indeed, one of the main properties that one expects from a well-behaved proof system is not only that it possesses a cut-elimination procedure, but also that this procedure enjoys a confluence property similar to the Church-Rosser property of the λ-calculus. But confluence is a way of expressing determinism in a rewriting setting, so that being able to represent the real concurrency of the π-calculus [2] requires to be able to go beyond confluence.

However, if one renounces to the domain-theoretic viewpoint on semantics, or more precisely, to the fact that the domains interpreting the types should have some kind of built-in coherence, or compatibility notion, then there is no obstacle to define models of λ-calculi, or of linear logic, which admit non-determinism under the guise of the possibility of defining arbitrary joins (or unions, or sums) of points. Such models of linear logic have been designed by the first author, using vector spaces [3]. Operations, which were absent from linear logic such as *differentiation* become quite natural, and strongly use the possibility of adding vectors, that is, the non-determinism of the model. Proof-theoretic counterparts of these models are extensions of the rules of linear logic allowing to recover, at the exponential level, a symmetry that linear logic possesses for its multiplicative and additive connectives. Here we will focus on *differential interaction nets* introduced in a joint work by the first author and Laurent Regnier [4].

In a joint work with Kohei Honda [5], the second author proposed a translation of a version of the π-calculus in an extension of polarized proof-nets. This allows to recast in a linear logic setting a typing system for the π-calculus previously introduced by Berger, Honda and Yoshida in [6]. This translation behaves quite well, in the sense that π-calculus reduction is faithfully simulated by the reduction of linear logic proof-nets and therefore has to be considered as a convincing Curry-Howard interpretation of processes. However, this translation is not really modular, in the sense for instance that the interpretation of the parallel composition of two processes can hardly be described by connecting together the corresponding proof-nets through their conclusions. Other relations between some kinds of proof-nets (or interaction nets) and process calculi have recently been worked out (for example [7–9]) with the lack of a clear underlying semantic setting.

The purpose of the present paper is to continue this line of ideas, using more systematically the new structures introduced by differential interaction nets solving, we think in a satisfactory way, the above mentioned defects.

1 A finitary and monadic π-calculus

We consider a monadic, finitary (without replication nor recursion), localised π-calculus without sums and matching. Processes are defined as follows (inaction, parallel composition, restriction, input prefix and output prefix):

$$\pi ::= 0 \;\Big|\; \pi \mid \pi \;\Big|\; \nu a \cdot \pi \;\Big|\; a(b) \cdot \pi \;\Big|\; \overline{a}\langle b\rangle \cdot \pi$$

Moreover, we restrict this grammar to *localised* terms [2]: the object of a reception cannot be the subject of a reception (typically $a(b) \cdot b(c) \cdot \pi$ is rejected).

The set $\mathsf{FV}(\pi)$ of free names of a process π is defined in the usual way (the only binders are the restriction and the input prefix constructions).

Processes are considered up to a structural congruence $\sim$ which is the least one such that:

$$0 \mid \pi \sim \pi \qquad (\pi_1 \mid \pi_2) \mid \pi_3 \sim \pi_1 \mid (\pi_2 \mid \pi_3) \qquad \pi_1 \mid \pi_2 \sim \pi_2 \mid \pi_1$$

$$\nu a \cdot \nu b \cdot \pi \sim \nu b \cdot \nu a \cdot \pi \qquad \nu a \cdot (\pi_1 \mid \pi_2) \sim (\nu a \cdot \pi_1) \mid \pi_2 \quad \text{if } a \notin \mathsf{FV}(\pi_2)$$

We can now define the reduction relation on processes:

$$\frac{}{\overline{a}\langle b\rangle \cdot \pi_1 \mid a(c) \cdot \pi_2 \rightsquigarrow \pi_1 \mid \pi_2\,[b/c]} \qquad \frac{\pi_1 \sim \pi_1' \qquad \pi_1' \rightsquigarrow \pi_2' \qquad \pi_2' \sim \pi_2}{\pi_1 \rightsquigarrow \pi_2}$$

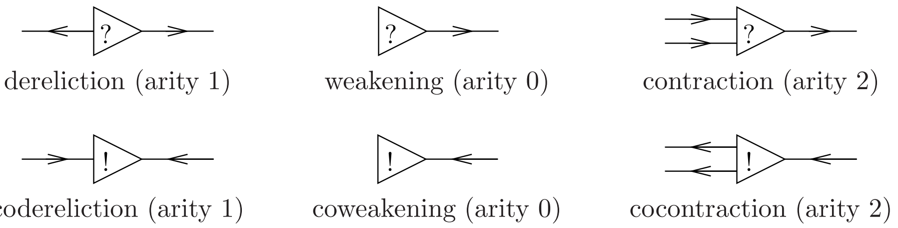
$$\frac{\pi_1 \rightsquigarrow \pi_2}{\pi_1 \mid \pi_0 \rightsquigarrow \pi_2 \mid \pi_0} \qquad \frac{\pi_1 \rightsquigarrow \pi_2}{\nu a \cdot \pi_1 \rightsquigarrow \nu a \cdot \pi_2}$$

2 Untyped differential interaction nets

Very much like terms are made of function symbols, interaction nets [1] are made of cells. Each cell has an arity n and has $n+1$ *ports*, among which one *principal port* and n *auxiliary ports*. Cells are pictured as triangles, and ports are drawn on the border of these triangles: the principal port is located on one of the angles of the triangle, and the other ports, on the opposite side of the triangle.

2.1 Cells and nets

In purely exponential finitary differential interaction nets that we consider here, there are six kinds of cells:



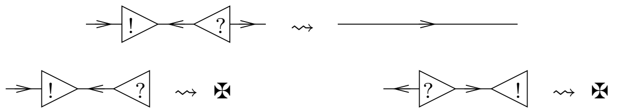
A *net* is a combination of cells, connected with each other by *oriented wires*, as described in [1, 4]. Non-connected ports are called *free ports*. The orientation has to respect the constraints given on cells. We also add a very particular net ✠ (which is not a combination of cells, but a formal object). It has to be understood as the fact that an error has occurred during reduction (see the end of the paper for more comments). For the present work, we do not ask for any kind of connectedness or correctness criterion for nets.

We call *unit net* the net which has no cells and no free ports.

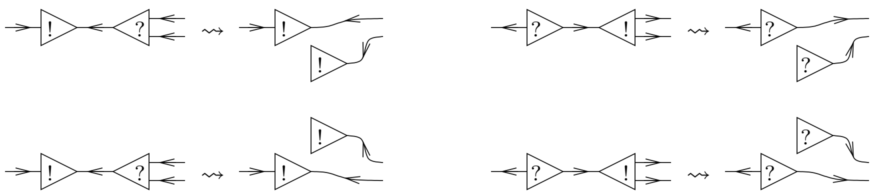
2.2 Reduction rules for nets

We introduce now the reduction rules for nets. We refer to [4] for a simple mathematical interpretation of these rules, in terms of elementary differential calculus. As usual in interaction nets, a redex is a sub-net consisting of two cells connected by their principal ports. Reducing a net consists in replacing this sub-net by a net which has the same free ports. The reduction rules on nets are divided into two groups.

Computational reduction rules. They correspond to the interaction between dereliction/codereliction and other cells (7 rules).

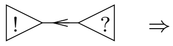


Since ✠ is a net but cannot appear as part of net, by $\rightsquigarrow$ ✠, we mean that the entire net reduces to the distinguished net ✠.

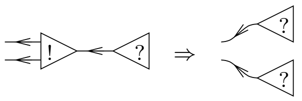


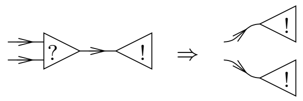
In these last four rules two redexes have two possible reductions. This is where non-determinism appears!

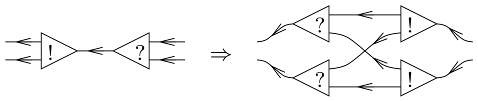
Structural reduction rules. They correspond to the interaction between the coweakening, cocontraction, weakening and contraction cells (4 rules).



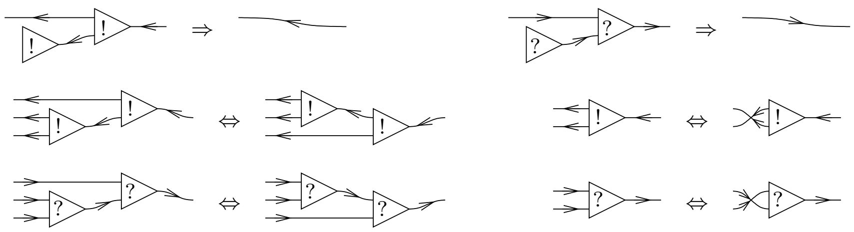
where the right-hand side of this reduction rule is the unit net. This means that if one encounters this coweakening/weakening redex in a net, one can simply erase it, without modifying the rest of the net.







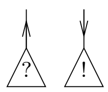
The structural reduction rules will also include the Rétoré reduction rules and equivalences. These are not exactly standard interaction nets reduction rules, but auxiliary conversion rules. The most important ones are those which express the neutrality of coweakening and weakening. The other rules express the associativity and commutativity of cocontraction and contraction and therefore cannot be oriented.

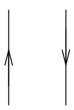


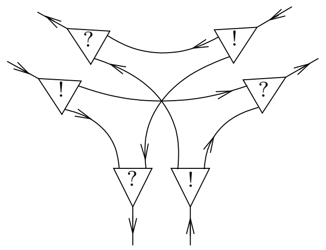
We will now consider nets up to the *structural* congruence $\equiv$ generated by structural rules ($\Leftrightarrow$). If we consider nets up to associativity and commutativity of contraction and cocontraction cells, we can define a notion of canonical form up to structural congruence (all the other rules have a natural orientation).

3 From processes to nets

Broadcast areas are the key objects allowing for a modular account of parallel composition. The simplest ones Br_{-1}, Br_0 and Br_1 are the following nets:





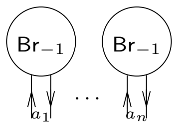


One defines more generally Br_n for each integer $n \geq -1$. This net is made of $n+2$ pairs of $(n+1)$-ary cocontraction and contraction cells with, for each $1 \leq i < j \leq n+2$, a wire from an auxiliary door of the i^{th} contraction (resp. cocontraction) to an auxiliary door of the j^{th} cocontraction (resp. contraction).

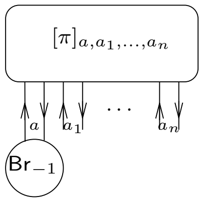
One can combine two broadcast areas Br_p and Br_q (as soon as $p+q \geq -1$) by connecting them through a chosen free port of each one. Such a combination of broadcast areas is structurally equivalent to the broadcast area Br_{p+q}.

Given a process π and a finite set A of names containing all the free names of π, we define a net $[\pi]_A$ with $2n$ free ports: to each element of A is associated a pair of free ports oriented in the opposite way (an input port oriented towards the net and an output port oriented towards the outside). The definition is by induction on processes.

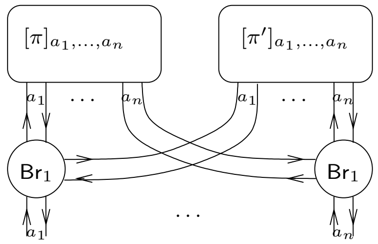
$[0]_{a_1,\ldots,a_n}$ is the net



$[\nu a \cdot \pi]_{a_1,\ldots,a_n}$ is the net

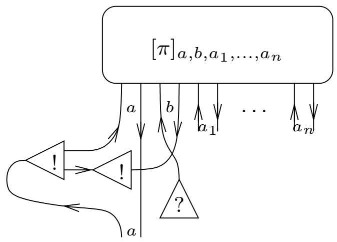


$[\pi \mid \pi']_{a_1,\ldots,a_n}$ is the net

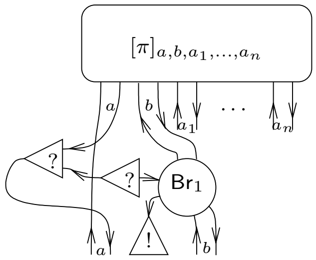


Observe that we introduce as many broadcast areas as there are free names under consideration.

$[a(b) \cdot \pi]_{a,a_1,\ldots,a_n}$ is the net



$[\overline{a}\langle b\rangle \cdot \pi]_{a,b,a_1,\ldots,a_n}$ is the net

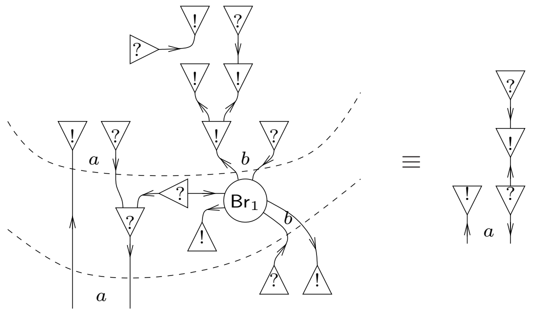


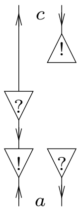
The encoding of the input prefix corresponds to the following idea: we kill any reception capability on b by plugging a weakening cell (this corresponds to locality), we create the capability of receiving b on a through the codereliction cell and finally we merge this capability with the other reception capabilities on

a through the cocontraction cell (the emission capabilities of a are unchanged, meaning that our encoding of prefixes is not as strong as in the π-calculus). The encoding of the output prefix is done in the same spirit but we introduce a broadcast area to allow b to still communicate with the outside.

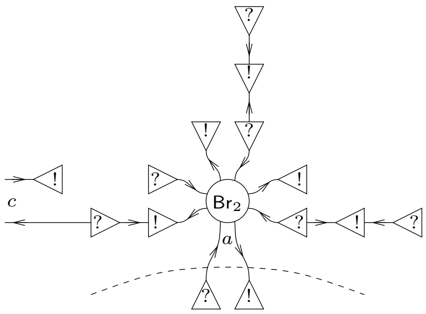
Here are two examples of translated processes (in the translation of π_1, the three levels respectively correspond to $[b(d) \cdot 0]_{a,b}$, $[\overline{a}\langle b\rangle \cdot b(d) \cdot 0]_{a,b}$ and $[\pi_1]_a$):

$\pi_1 = \nu b \cdot \overline{a}\langle b\rangle \cdot b(d) \cdot 0$ $\qquad\qquad$ $\pi_2 = a(x) \cdot \overline{x}\langle c\rangle \cdot 0$





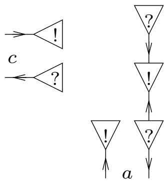
If we now want to translate the compound process $\pi_2 \mid \pi_1 \mid \pi_1$, we can treat the translations of π_1 and π_2 as black boxes and just connect them through broadcast areas. This is why we consider this translation as more modular than it is in [5], here we do not need to look inside the structure of $[\pi_1]_{a,c}$ and $[\pi_2]_{a,c}$ to describe $[\pi_2 \mid \pi_1 \mid \pi_1]_{a,c}$. The following picture gives both $[\pi_2 \mid \pi_1 \mid \pi_1]_{a,c}$ (above the dashed line) and $[\nu a \cdot (\pi_2 \mid \pi_1 \mid \pi_1)]_c$ (the whole picture):

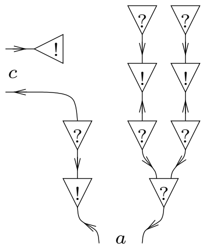


Proposition (Congruence). *Given a process π and a finite set A of names such that $\mathsf{FV}(\pi) \subset A$, if $\pi \sim \pi'$ then $[\pi]_A \equiv [\pi']_A$.*

Theorem (Simulation). *Given a process π and a finite set A of names such that $\mathsf{FV}(\pi) \subset A$, if π reduces to π' then $[\pi]_A$ computationally reduces to $[\pi']_A$ (that is, $[\pi]_A \rightsquigarrow^* [\pi']_A$).*

The process $\pi_2 \mid \pi_1 \mid \pi_1$ reduces to $\nu b \cdot (b(d) \cdot 0 \mid \overline{b}\langle c\rangle \cdot 0) \mid \pi_1$ (with a choice of the π_1 which has to interact with π_2) which itself reduces to π_1. On the differential interaction nets side, the role of the broadcast area in $[\pi_2 \mid \pi_1 \mid \pi_1]_{a,c}$ is to make possible any connection between an emission on a and a reception on a. By reduction, we obtain the following two nets:





The first one is the translation of π_1 and can be obtained in two different ways corresponding to the choice of the π_1 interacting with π_2 (this gives the simulation of the reduction in the π-calculus). The second one would correspond to the explicit choice that no internal communication will occur on a and thus only labelled transitions with the outside are possible (two possible emissions from the π_1s and one reception from π_2). The net $\maltese$ is the last possible normal form but can be neglected as soon as other normal forms exist.

There exist non-trivial processes π such that $[\pi]_A \rightsquigarrow \maltese$ (see for example $[\nu a \cdot (\pi_2 \mid \pi_1 \mid \pi_1)]_c$), this mainly corresponds to private names on which the numbers of emissions and receptions are not equal. However it is far from being the case of all processes so that our embedding is not completely degenerate.

Future directions. While there is no problem for dealing with the polyadic π-calculus by introducing back the multiplicative connectives of linear logic, dealing with sums requires to consider sets (or formal sums) of nets (as shown in [4], this is natural in the setting of differential linear logic and represents both non-determinism and sums of processes). Dealing with replication should involve the use of the promotion rule of linear logic (as done in [5]) but this has still to be properly understood.

References

1. Lafont, Y.: From proof nets to interaction nets. In Girard, J.Y., Lafont, Y., Regnier, L., eds.: Advances in Linear Logic. Cambridge University Press (1995) 225–247
2. Sangiorgi, D., Walker, D.: The pi-calculus: a Theory of Mobile Processes. Cambridge University Press (2001)
3. Ehrhard, T.: On Köthe sequence spaces and linear logic. Mathematical Structures in Computer Science **12** (2002) 579–623
4. Ehrhard, T., Regnier, L.: Differential interaction nets. In: Proceedings of WoLLIC'04. Volume 123 of ENTCS., Elsevier Science (2004) 35–74
5. Honda, K., Laurent, O.: Processes and polarized proof-nets. Unpublished (2006)
6. Berger, M., Honda, K., Yoshida, N.: Strong normalisability in the pi-calculus. Information and Computation (2003)
7. Beffara, E., Maurel, F.: Concurrent nets: a study of prefixing in process calculi. Theoretical Computer Science **356** (2005)
8. Mazza, D.: Multiport interaction nets and concurrency. In: Proceedings of CONCUR 2005. Number 3653 in LNCS, Springer-Verlag (2005) 21–35
9. Curien, P.L., Faggian, C.: An approach to innocent strategies as graphs. Technical report, Preuves, Programmes et Systèmes (2006) Submitted for publication.