CiE 2006

Embedding the π-calculus in differential interaction nets

Thomas Ehrhard
PPS
Université Denis Diderot and CNRS

Joint work with Olivier Laurent, PPS

02-07-2006

A finitary and monadic π-calculus. Names $a, b, a_1, \ldots$

- *Empty process:* $*$
- *Restriction:* $\nu a \cdot \pi$ (a is bound)
- *Parallel composition:* $\pi_1 \mid \pi_2$
- *Reception:* $a(b) \cdot \pi$ (b is bound)
- *Emission:* $\overline{a}\langle b \rangle \cdot \pi$ (b is *not* bound)

Operational semantics: a machine.

- *Closure:* $c = (\pi, e)$ where e is a finite function from the free names of π to names;

- *Soup:* multiset $S = c_1 \ldots c_n$ of closures;

- *State:* $(S, \mathcal{P})$ where $\mathcal{P}$ is a finite set of names, the private names of the state [the names in $\mathcal{P}$ must be considered as bound].

A soup is *canonical* if all its processes start with an input or output prefix.

A machine: reduction rules

- $((*, e)S, \mathcal{P}) \rightsquigarrow_{\mathsf{can}} (S, \mathcal{P})$

- $((\pi_1 \mid \pi_2, e)S, \mathcal{P}) \rightsquigarrow_{\mathsf{can}} ((\pi_1, e)(\pi_2, e)S, \mathcal{P})$

- $((\nu a \cdot \pi, e)S, \mathcal{P}) \rightsquigarrow_{\mathsf{can}} ((\pi, e[a \mapsto \alpha])S, \mathcal{P} \cup \{\alpha\})$ with α a fresh name

- $((a_1(b){\cdot}\pi_1, e_1)(\overline{a_2}\langle c\rangle{\cdot}\pi_2, e_2)S, \mathcal{P}) \rightsquigarrow ((\pi_1, e_1[b \mapsto e_2(c)])(\pi_2, e_2)S, \mathcal{P})$ if $e_1(a_1) = e_2(a_2)$.

It is a simple way of presenting the usual reduction rules and congruence of the π-calculus (cf. the abstract machine of Amadio and Curien's book).

Pure polarized exponential differential logic.

- *Positive formulae:* ι and $!N$ where N is negative
- *Negative formulae:* $?P$ where P is positive
- *Equation on formulae:* $\iota = !(\iota^{\perp})$ [cf. $D = D \Rightarrow D$ in pure lambda-calculus]

Set $o = ?\iota$, so that $o = \iota^{\perp}$.

Up to equality of formulae, there are only two formulae in this logic:

ι (or "+") and o (or "−").

Pure polarized exponential differential logic: a sequent calculus.

Identity rules:

$$\frac{}{\vdash o, \iota} \qquad \frac{\vdash \Gamma, o \quad \vdash \iota, \Delta}{\vdash \Gamma, \Delta}$$

Negative rules (structural rules and dereliction):

$$\frac{\vdash \Gamma}{\vdash \Gamma, o} \qquad \frac{\vdash \Gamma, o, o}{\vdash \Gamma, o} \qquad \frac{\vdash \Gamma, \iota}{\vdash \Gamma, o}$$

Pure polarized exponential differential logic: a sequent calculus (cont.)

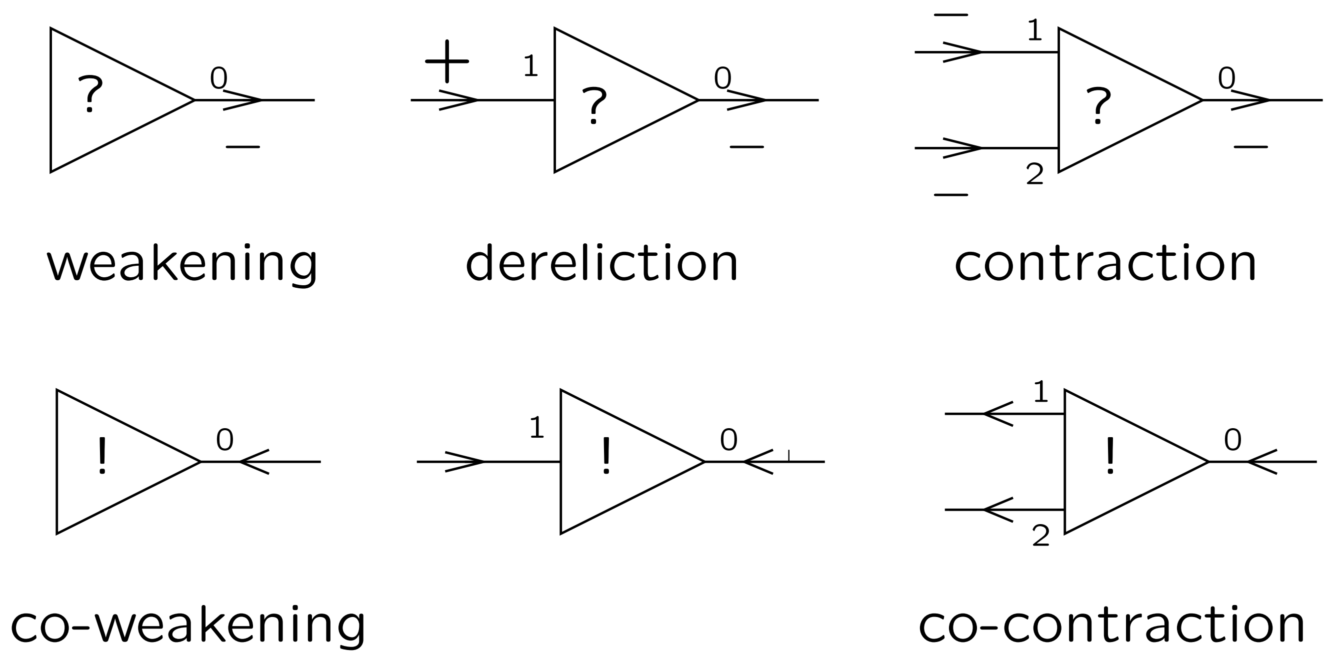
Positive rules (costructural rules and codereliction):

$$\frac{}{\vdash \iota} \qquad \frac{\vdash \Gamma, \iota \quad \vdash \Delta, \iota}{\vdash \Gamma, \Delta, \iota} \qquad \frac{\vdash \Gamma, o}{\vdash \Gamma, \iota}$$

Mix rule:

$$\frac{}{\vdash} \qquad \frac{\vdash \Gamma \quad \vdash \Delta}{\vdash \Gamma, \Delta}$$

Differential interaction nets: interaction nets for this logic.



Arrow convention: the arrows we put on wires correspond to the typing of all wires by the o ("$-$") type.

Differential interaction nets: desquentialization.

Any proof of the sequent calculus above can be "desequentialized" into a unique interaction net structure with these cells, just like in MLL.

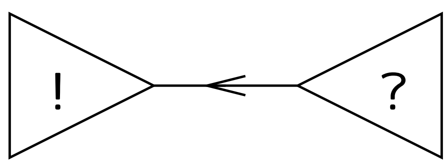
Differential interaction nets: correctness criterion.

Adaptation of the Girard or Danos-Regnier MLL criterion, but only for acyclicity. Connexity is not required.

Cocontraction is handled like a *tensor* link, *contraction* is handled like a *par* link.

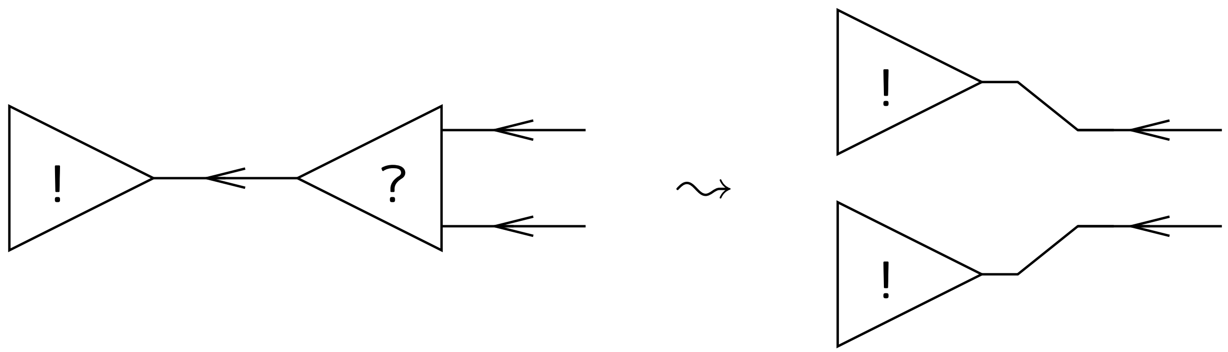
Fact: a net structure which satisfies the criterion is the de-sequentialization of a proof of the sequent calculus of the ι/o logic.

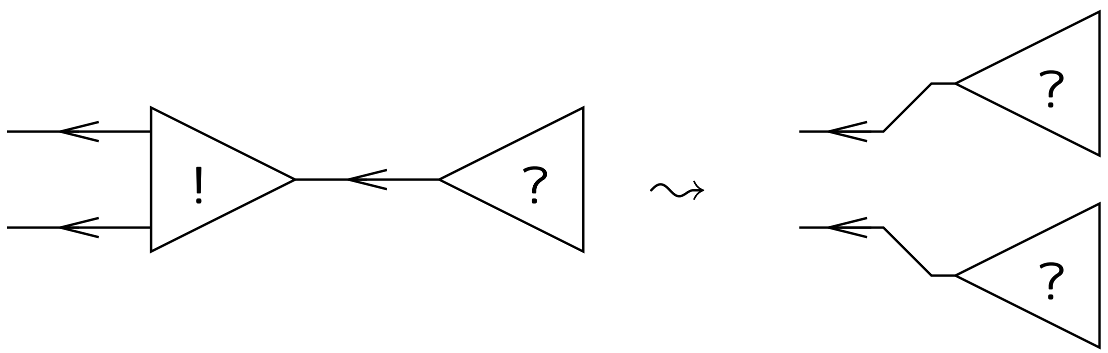
Differential interaction nets: structural rewriting rules. Correspond to the fact that each type $!A$ has the structure of a *bialgebra*.



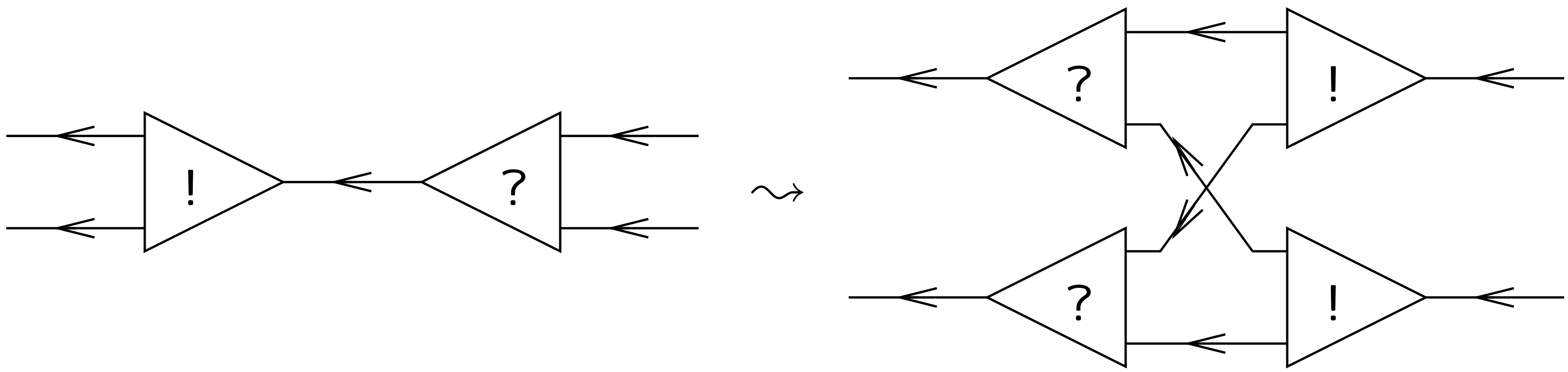
$\rightsquigarrow$ (multiplicative nothing)

Differential interaction nets: structural rewriting rules (cont.)





Differential interaction nets: structural rewriting rules (cont.)



Differential interaction nets: structural rewriting rules (cont.)

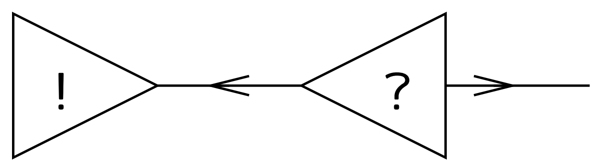
Structural rules include the Rétoré rules which express the neutrality of weakening wrt. contraction, and of coweakening wrt. cocontraction.

We must also work up to associativity of contraction and cocontraction.

Strangely enough, neither commutativity of cocontraction nor cocommutativity of contraction seem to be necessary for translating replication-free processes.

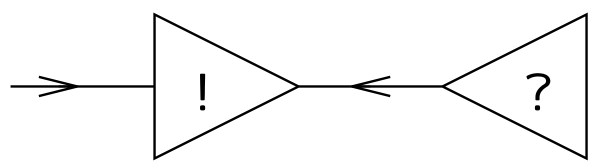
Differential interaction nets: non-deterministic rewriting rules.

Describe the behaviour of linearity and differentiation — which have some kind of duality — wrt. costructural and structural cells.



$\rightsquigarrow \quad 0$

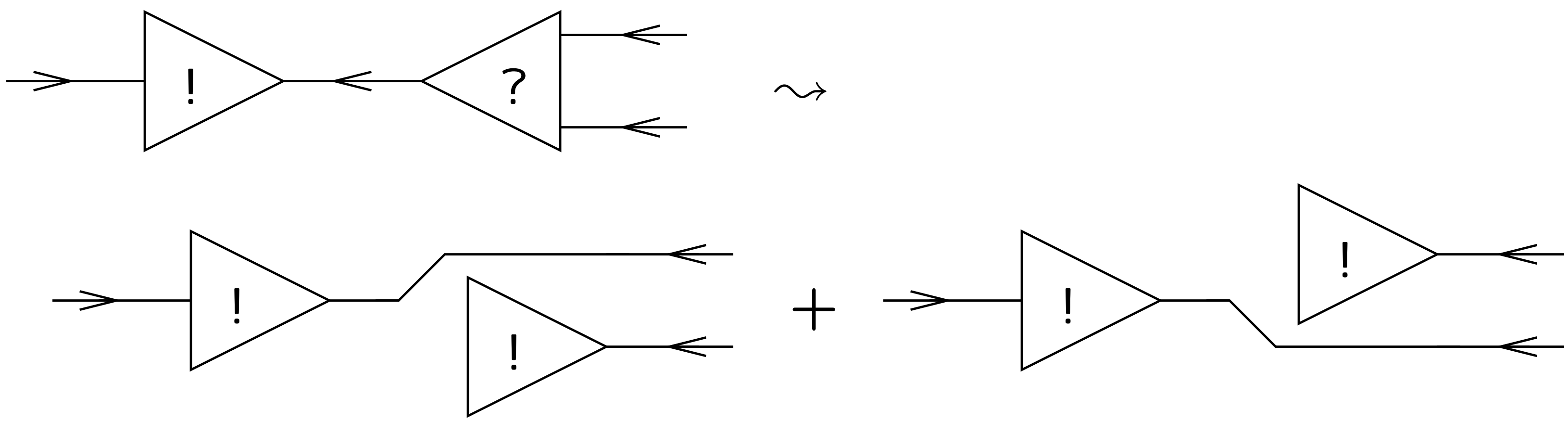
"Applying a linear function to 0 yields 0."



$\rightsquigarrow \quad 0$

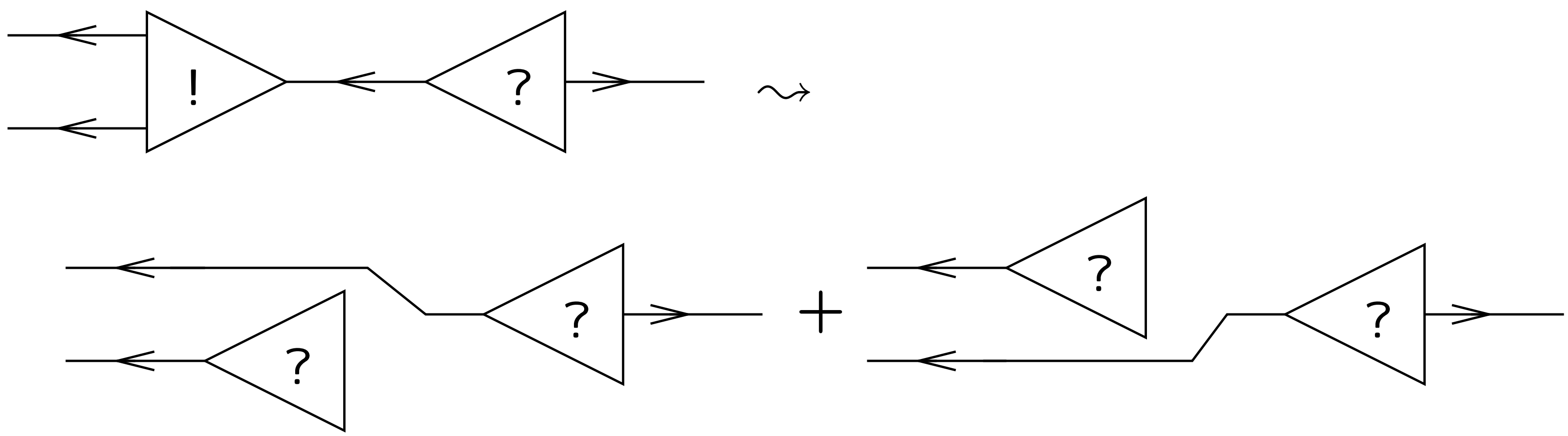
"Derivating a constant function yields 0."

Differential interaction nets: non-deterministic rewriting rules (cont.)



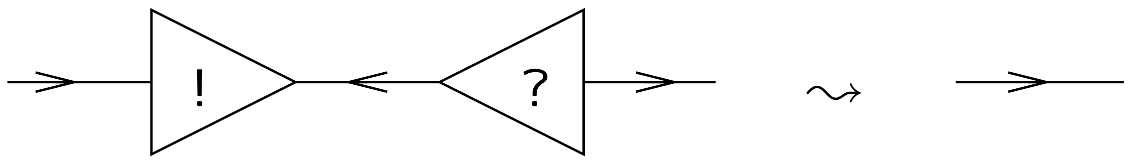
"Derivating $f(x,x)$ wrt. x yields a sum of two partial derivatives."

Differential interaction nets: non-deterministic rewriting rules (cont.)



"Applying a linear function to a sum yields a sum."

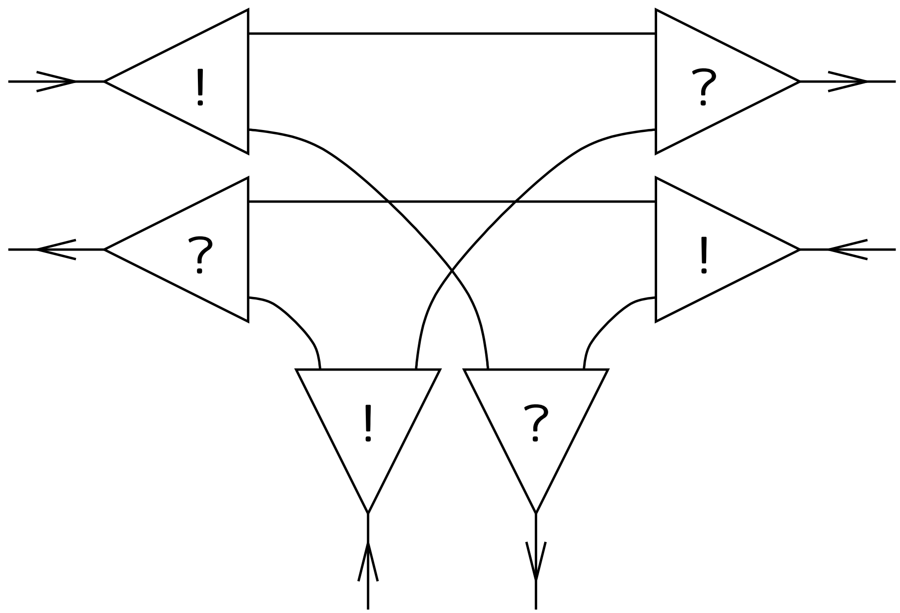
Differential interaction nets: communication rewriting rule (cont.)



"The value of the derivative of a linear function at any point is that function."

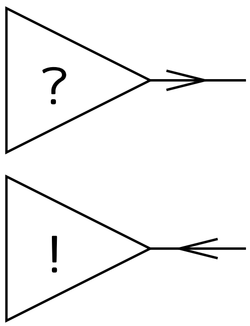
Broadcast areas.

A family of nets Br_n for $n = -1, 0, 1, 2, \ldots$, with $2n+4$ free ports. The most interesting is Br_1:

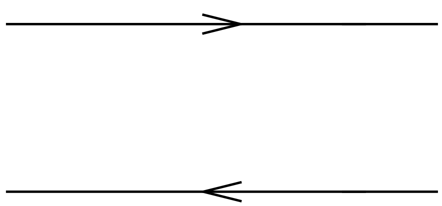


Broadcast areas (cont.)

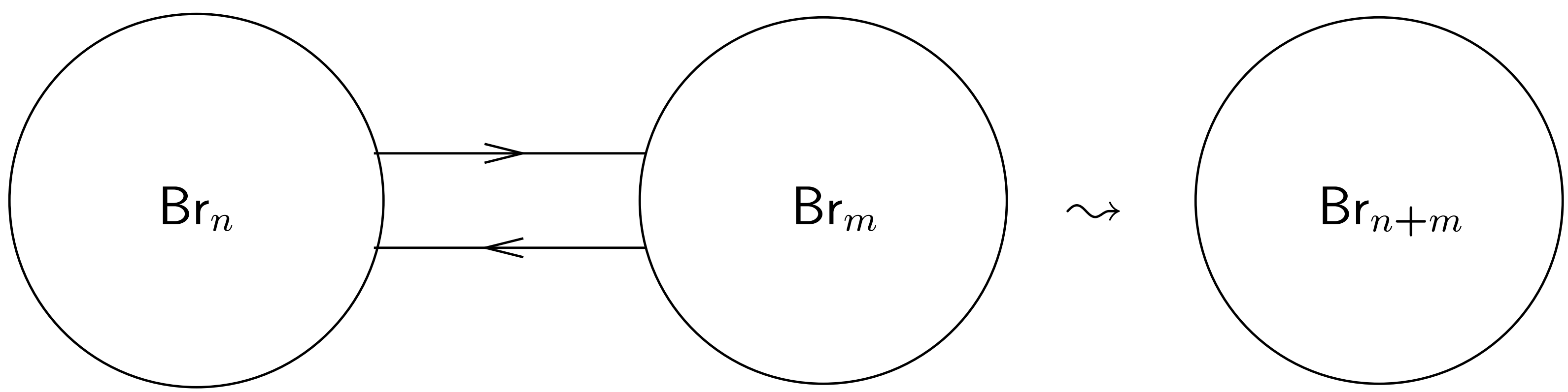
Br_{-1} is



and Br_0 is



Broadcast areas: associativity.

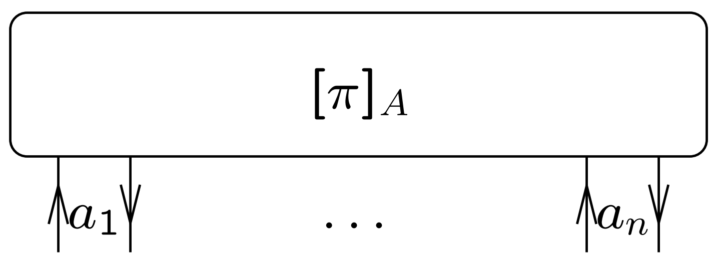


using only structural reduction rules.

Interpreting processes: principle.

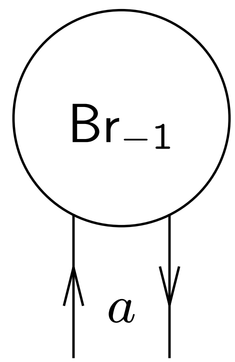
Let π be a process, A a set of names containing all the free names of π.

Define $[\pi]_A$, a net with two free ports a^+ and a^- for each $a \in A$; if $A = \{a_1, \ldots, a_n\}$:



Interpreting processes: empty process.

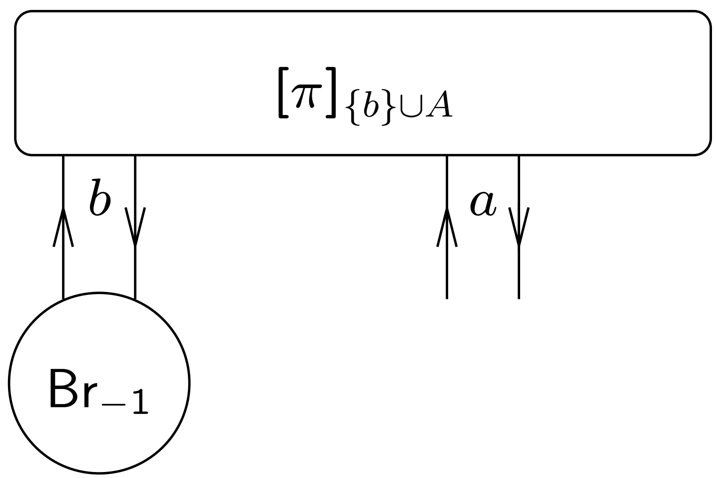
$[*]_A$ is the juxtaposition of nets



one for each $a \in A$.

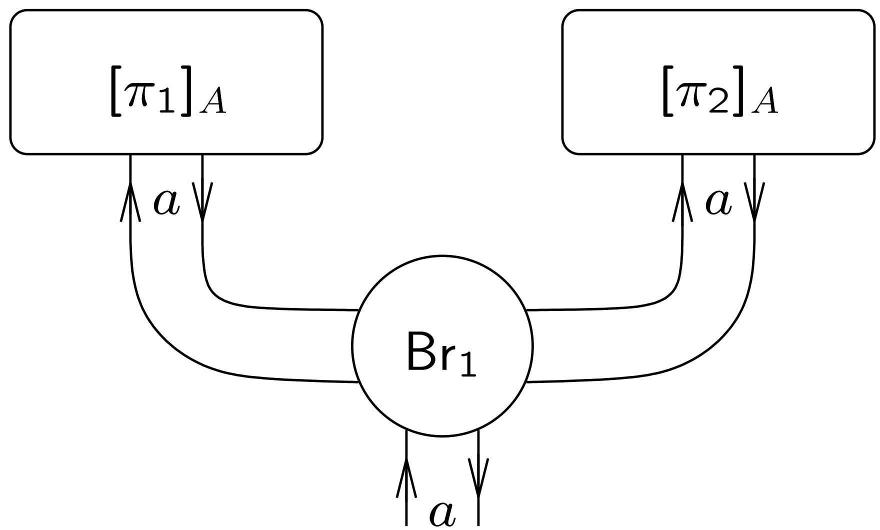
Interpreting processes: name restriction.

a: a generic element of A, b: a name with $b \notin A$. Then $[\nu b \cdot \pi]_A$ is



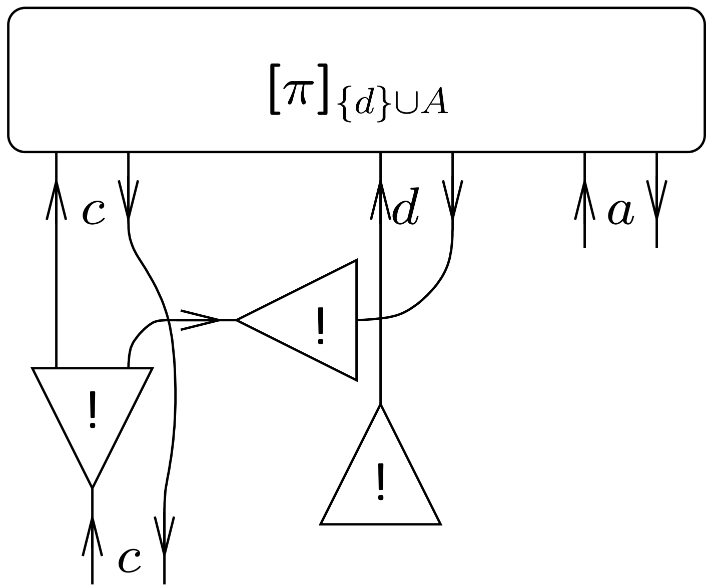
Interpreting processes: parallel composition.

a: a generic element of A, then $[\pi_1 \mid \pi_2]_A$ is



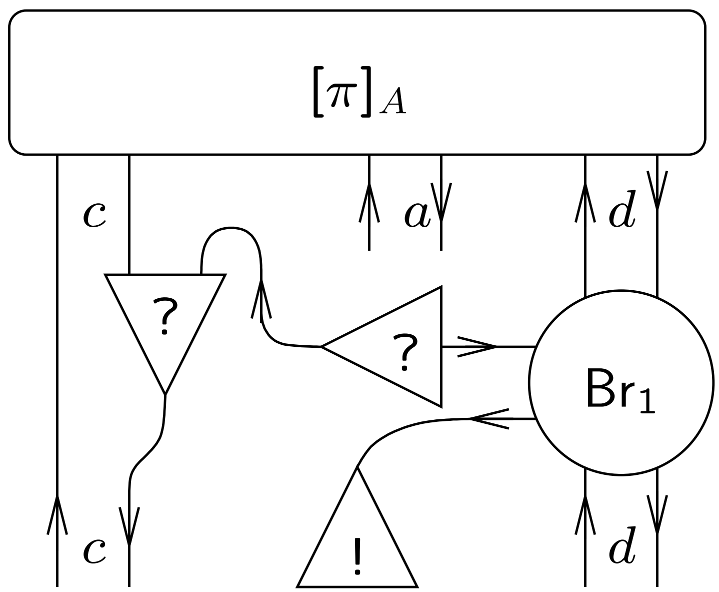
Interpreting processes: reception.

a: a generic element of A, c and d two distinct names, $c \in A$ and $d \notin A$. Then $[c(d) \cdot \pi]_A$ is



Interpreting processes: emission.

a: a generic element of A, c and d two distinct names, $c \in A$ and $d \notin A$. Then $[\overline{c}\langle d\rangle \cdot \pi]_A$ is



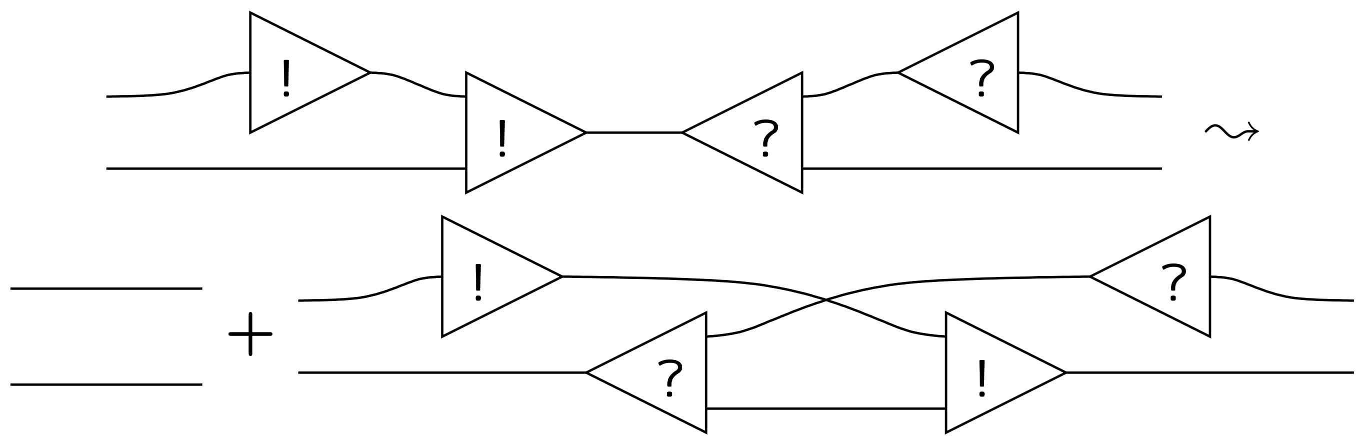
Interpreting states.

This translation extends easily to states.

Fact: the interpretations of processes and of states are "weakly correct" nets (that is: they can contain switching cycles, but all these cycles pass through dereliction or codereliction cells, or: none of these cycles is oriented, with our arrow convention).

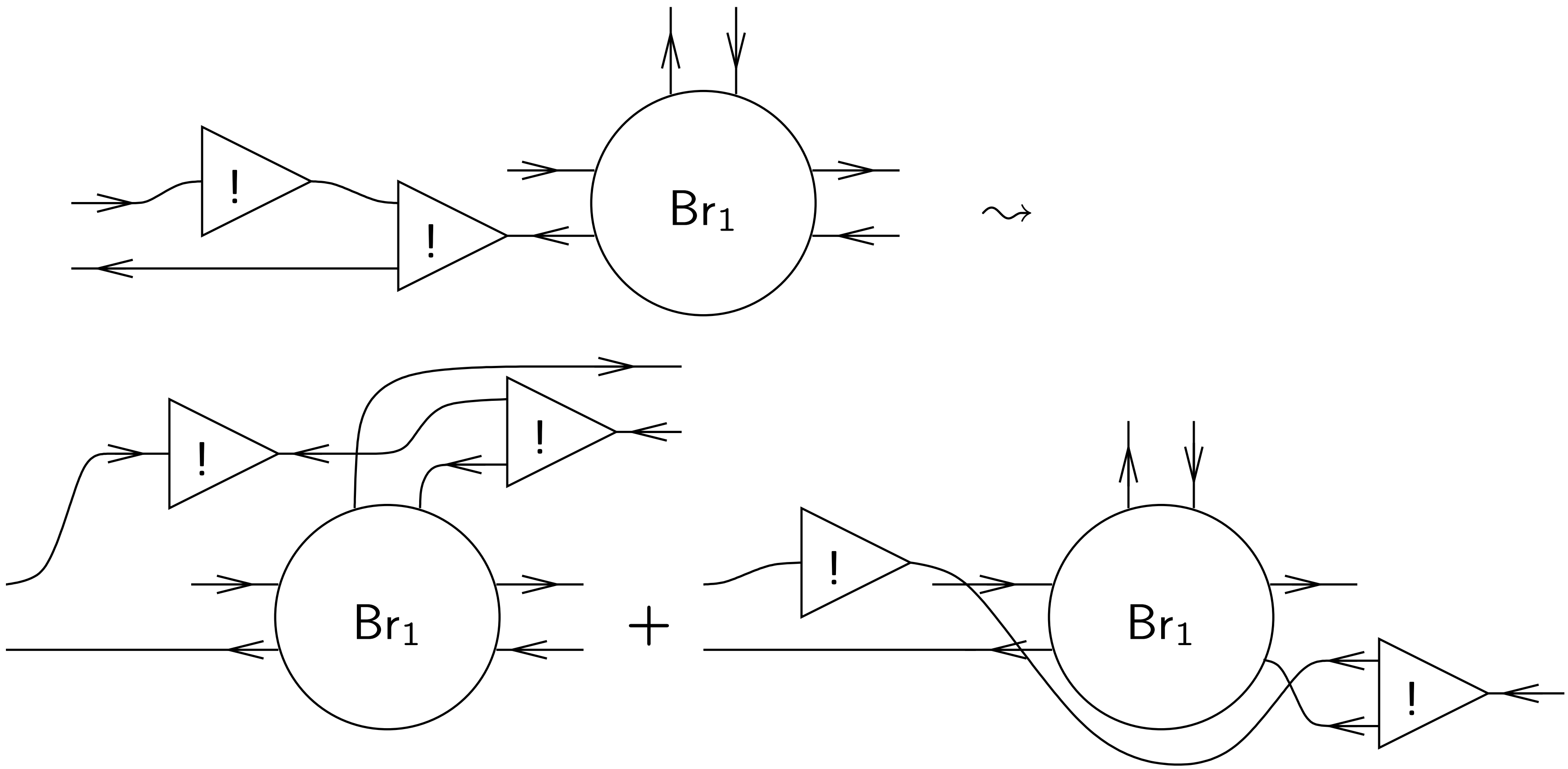
Key reductions in process interpretations: prefix/prefix.

The following reductions hold in nets:



Key reduction in process interpretations: prefix/broadcast.

The following reduction (as well as its dual) holds in nets:



Simulation theorem.

If we interpret a reduction of nets $t \rightsquigarrow t_1 + \cdots + t_n$ (where t and the t_i's are not sums) as the fact that $t \rightsquigarrow t_1, \ldots,$ $t \rightsquigarrow t_n$ non-deterministically, then:

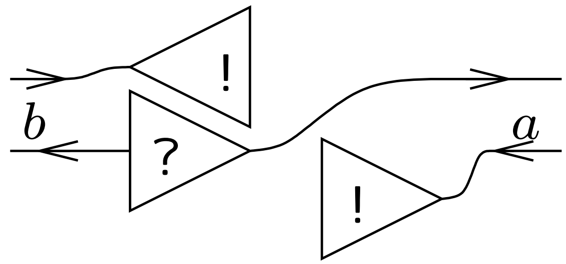
$$(S, \mathcal{P}) \rightsquigarrow (S', \mathcal{P}') \Rightarrow [(S, \mathcal{P})] \rightsquigarrow [(S, \mathcal{P}')]$$

where $[(S, \mathcal{P})]$ is the net associated to the state $(S, \mathcal{P})$ of our machine. This holds in the *localized* π-calculus.

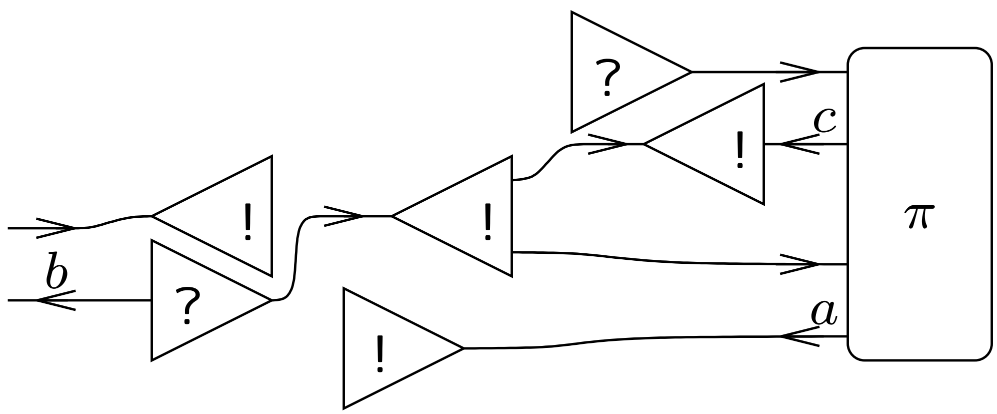
BUT the converse is false!

Example.

The interpretation of $\overline{a}\langle b\rangle \cdot *$ reduces to



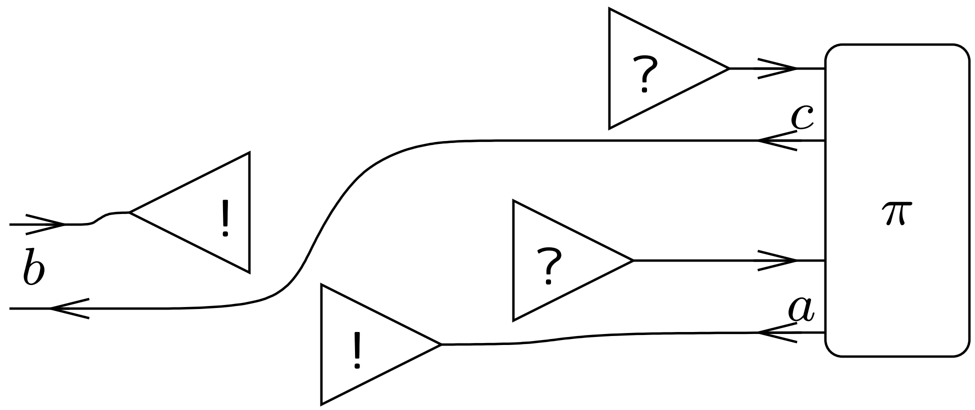
So the interpretation of $\nu a \cdot (\overline{a}\langle b\rangle \cdot * \mid a(c) \cdot \pi)$ reduces to



assuming that b is not free in π.

Example (cont.)

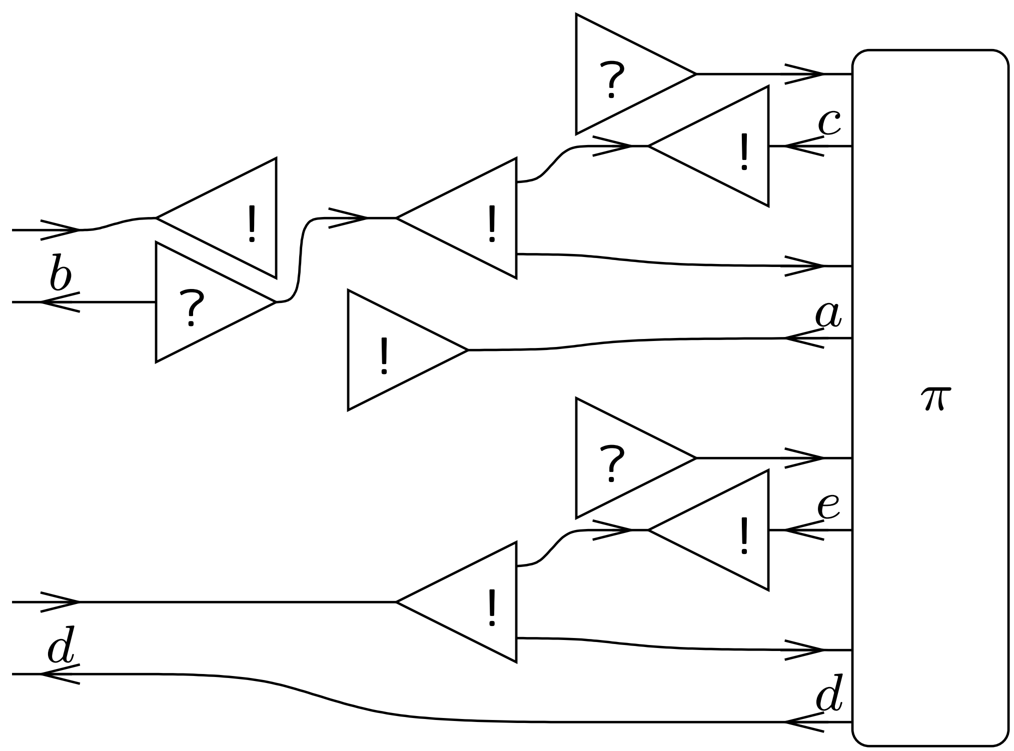
Which reduces to



and this net corresponds to $\nu a \cdot \pi[b/c]$ if we admit that π *has no reception capabilities on chanel* c, which is the case if we are in the *localized* π-calculus.

Example (cont.)

Consider now the interpretation of $\nu a \cdot (\overline{a}\langle b \rangle \cdot * \mid d(e) \cdot a(c) \cdot \pi')$; this net reduces to



Taking into account the sequentiality of prefixes of processes.

- Impose some sequentiality to differential interaction nets by forbidding that certain communication reductions occur before others (like Girard's *jumps* in MALL proof nets). Drawback: no clean denotational semantics known yet, whereas standard differential interaction nets have a lot of nice models.

- Instead of the π-calculus, consider a completely asynchronous calculus, translate this calculus to differential interaction nets, and encode the π-calculus in this asynchronous calculus using process calculi techniques.

Related works.

- Berger, Honda, Laurent, Yoshida
- Beffara and Maurel
- Mazza