Differential interaction nets and processes

Thomas Ehrhard Olivier Laurent

Preuves, Programmes et Systèmes

April 16, 2007

Objective

Show that differential interaction nets are sufficiently expressive for encoding faithfully a sgnificant fragment of the π-calculus.

The fragment: no sums (additives?), no recursion, no replication (promotion?).

Outline

A typing system

Single type symbol o (outputs), subject to the following recursive equation $o = ?o^{\perp} \parr o$.

We set $\iota = o^{\perp}$, so that $\iota = !o \otimes \iota$ and $o = ?\iota \parr o$.

Types are MELL formulae based on o and ι (up to these equations). Here, we use only o, ι, $!o$ and $?\iota$.

Typing a net consists in associating a type A to each oriented wire w. If w' is w reversed, the type of w' must be $A^{\perp}$.

Typing rules associated with cells must be respected.

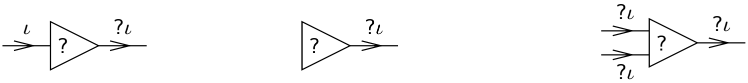
Multiplicative fragment

Binary cells:

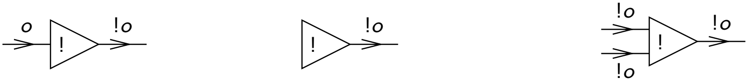
Constants:

Exponential fragment

Dereliction, weakening and contraction:



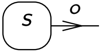
Codereliction, coweakening and cocontraction:

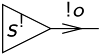


Closed promotion cell

A simple net is an interaction net made of these cells (respecting types), and of the fothcoming closed promotion cell.

A net is a finite formal sum of simple net with the same interface.

Given a (non necessarily simple) net s with only one free port 

we introduce a cell 

, called closed promotion.

Labels

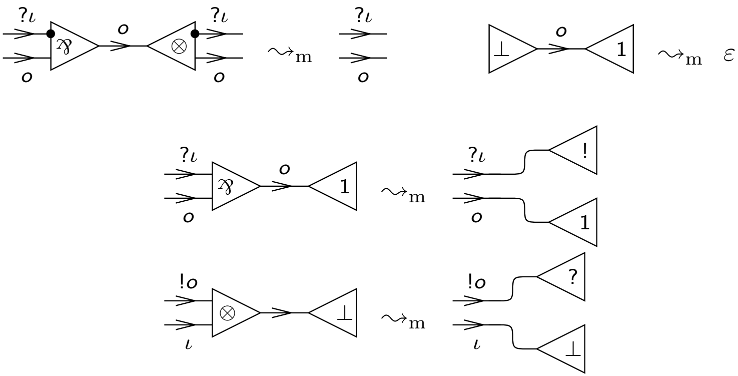
We use a set $\mathcal{L}$ of labels. They will determine what is observable from our reduction and used for defining labeled transitions systems of nets and of processes.

$\mathcal{L}$ is countable and has a dummy element τ.

The simple nets are labeled: each dereliction and each codereliction cell is equiped with a label from $\mathcal{L}$.

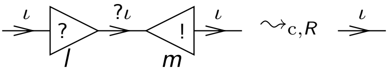
If, in a simple net, two of these labels are equal, they must be equal to τ.

Multiplicative reduction



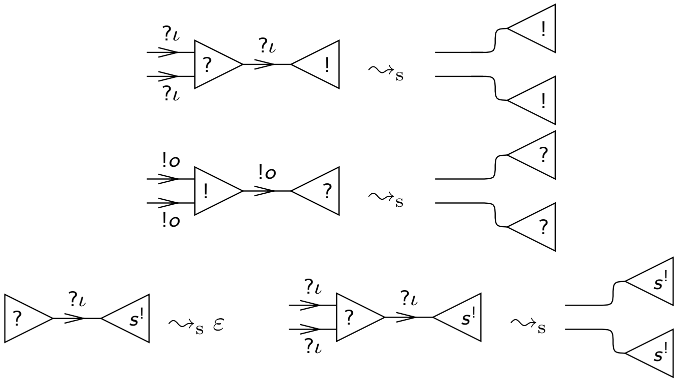
Exponentials: deterministic reductions

Let R be a set of labels, if $l, m \in R$, then we have the communication reduction:



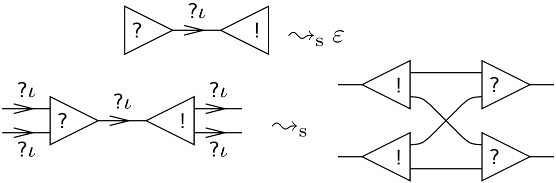
Exponentials: deterministic reductions (continued)

The next deterministic reduction rules are the structural ones:



Exponentials: deterministic reductions (continued)

Structural reductions (continued):



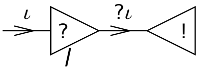
Semantically, contraction is associative, weakening is neutral for contraction etc. But there is no need to require corresponding reductions or equivalences on nets.

Exponentials: non-deterministic reductions

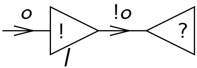
It is here that sums of nets appear. To be understood as non-deterministic superposition.

All net constructions distribute over sums of nets. If a subnet of a simple nets reduces to 0, the whole simple net reduces to 0.

If $R \subseteq \mathcal{L}$ and $l, r \in R$, we have the reductions:

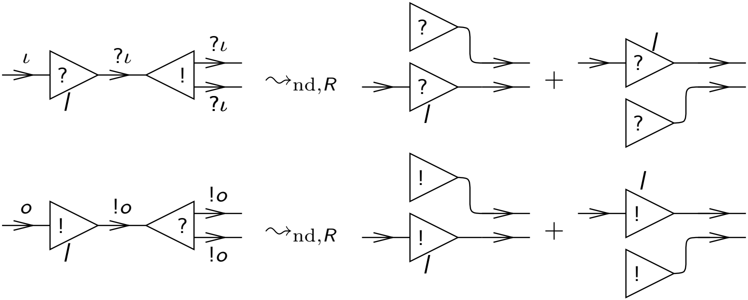


$\rightsquigarrow_{\mathrm{nd},R} 0$

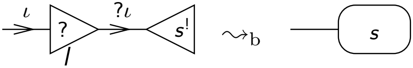


$\rightsquigarrow_{\mathrm{nd},R} 0$

Exponentials: non-deterministic reductions (continued)



Exponentials: promotion reduction



Confluence

Δ: the set of all nets, $\mathbb{N}\langle\Delta\rangle$: the set of all nets.

If $\mathcal{R} \subseteq \Delta \times \mathbb{N}\langle\Delta\rangle$ is a rewriting relation, $\mathcal{R}^* \subseteq \mathbb{N}\langle\Delta\rangle \times \mathbb{N}\langle\Delta\rangle$ is the transitive closure of its "extension to sums".

Theorem

Let $R, R' \subseteq \mathcal{L}$. Let $\mathcal{R} \subseteq \Delta \times \mathbb{N}\langle\Delta\rangle$ be the union of some of the reduction relations $\rightsquigarrow_{\mathrm{c},R}$, $\rightsquigarrow_{\mathrm{nd},R'}$, $\rightsquigarrow_{\mathrm{m}}$, $\rightsquigarrow_{\mathrm{s}}$ and $\rightsquigarrow_{\mathrm{b}}$. The relation $\mathcal{R}^*$ is confluent on $\mathbb{N}\langle\Delta\rangle$.

The proof is straightforward (reduction is local, no critical pairs).

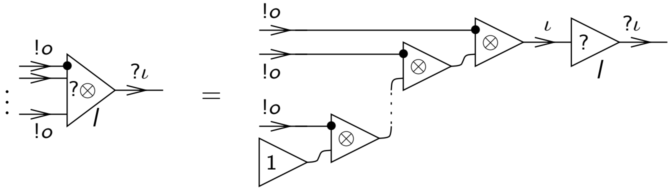
Particular reduction: $\rightsquigarrow_R = \rightsquigarrow_{\mathrm{m}} \cup \rightsquigarrow_{\mathrm{c},\{\tau\}} \cup \rightsquigarrow_{\mathrm{s}} \cup \rightsquigarrow_{\mathrm{b}} \cup \rightsquigarrow_{\mathrm{nd},R}$.
We set $\rightsquigarrow_{\mathrm{d}} = \rightsquigarrow_{\emptyset}$.

A labeled transition system $\mathbb{D}_{\mathcal{L}}$:

- objects: simple nets
- transitions labeled by pairs of labels
- $s \xrightarrow{l\overline{m}} t$ if $s \leadsto^{*}_{\{l,m\}} s_1 + s_2 + \cdots + s_n$ where
  - s_1 is a simple net which contains a communication redex with dereliction labeled by m and codereliction labeled by l, and becomes t when one reduces this redex
  - and for $i > 1$, whenever $s_i \leadsto^{*}_{\{l,m\}} s'$, none of the summands of s' has such a communication redex.

Dereliction-tensor and codereliction-par cells

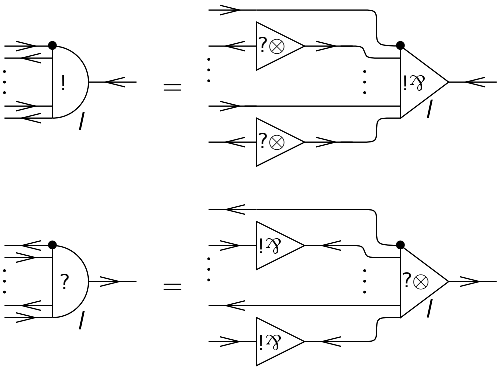
Let $n \in \mathbb{N}$ be a non-negative integer. We define an n-ary cell as follows. It will be decorated by the label of its dereliction cell (if different from τ).



Codereliction-par cell defined dually.

Prefix cells

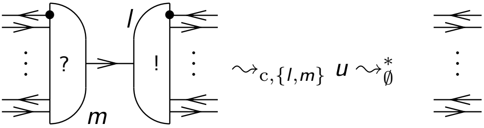
n-ary input and *n*-ary output prefix cells are



where n is the number of pairs of auxiliary ports.

Reduction of prefixes

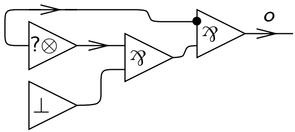
If the two prefix cells have the same arity, then one has

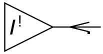


otherwise, the lefthand configuration reduces to 0 (but we can avoid this situation).

Boxed identity

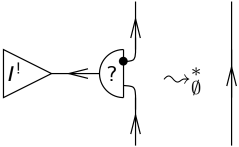
Let I be the following "identity" net



Then we shall use the closed promotion cell $I^!$: 

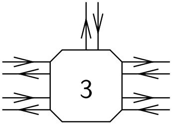
Transistor triggering

We use the unlabeled unary output prefix cell as a kind of transistor, triggered by the boxed identity cell, since indeed we have the reduction

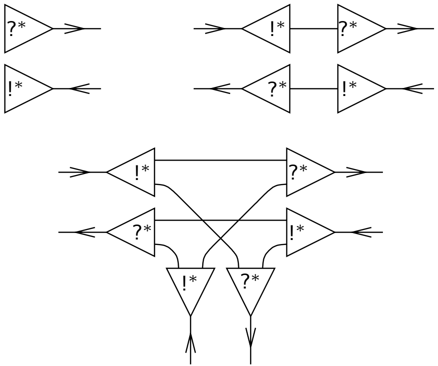


Communication areas

Let $n \geq -2$. We define a family of nets with $2(n+2)$ free ports, called communication areas of order n. Here is how we picture a communication area of order 3:

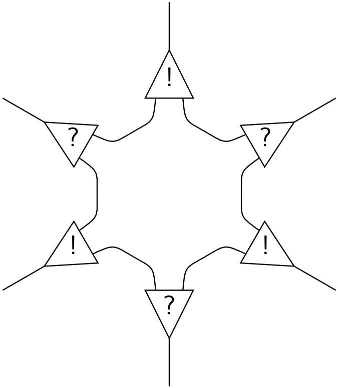


Communication areas of order -1, 0 and 1



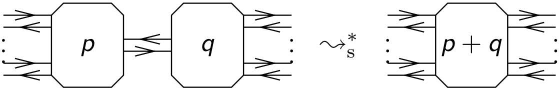
where the ?*-cells are “contraction trees” (containing possibly weakening cells) and similarly for the !*-cells.

Other representation of a communication area of order 1



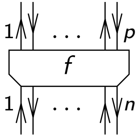
Aggregation of communication areas

When connecting two distinct communication areas through a pair of wires, one obtains a new one, applying only structural reductions:

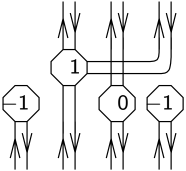


Identification structures

Given a function $f : \{1, \dots, p\} \to \{1, \dots, n\}$, one defines a structure, using only communication areas:

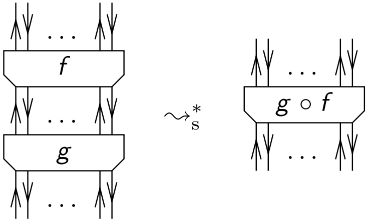


For instance, if $n = 4$, $p = 3$, $f(1) = 2$, $f(2) = 3$ and $f(3) = 2$, it is

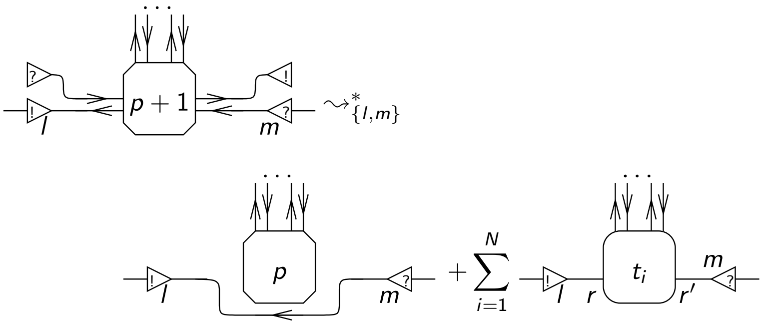


Identification structures composition

Applying communication areas aggregation, we have:



Interaction between prefixes and communication areas



Syntax

$\mathcal{N} = \{a, b, a_1, \dots\}$ a set of names.

- Empty process: nil
- Parallel composition: $P_1 \mid P_2$
- Name restriction: $\nu a \cdot P$
- Input prefix: $Q = [l]a(b_1 \dots b_n) \cdot P$ where the names $a, b_1, \dots, b_n$ are pairwise distinct. The b_is are bound. $l \in \mathcal{L}$.
- Output prefix: $\overline{[l]a}\langle b_1 \dots b_n \rangle \cdot P$, no restriction on the names $a, b_1, \dots, b_n$, they are all free in the process. $l \in \mathcal{L}$.

The labels of a process must be distinct from τ and pairwise distinct.

States of the machine

- Environment: finite partial function $e : \mathcal{N} \to \mathcal{N}$.
- Closure: (P, e) with all free names of P in the domain of e.
- Soup: multiset $S = (P_1, e_1) \cdots (P_N, e_N)$ with all labels pairwise distinct.
- State: (S, L) with $L \subseteq \mathcal{N}$ finite (the private names of the state).

The state is canonical if all the P_is start with input or output prefixes.

Canonical form of a state

The reduction

$$
\begin{array}{rcll}
((\mathrm{nil}, e)S, L) & \rightsquigarrow_{\mathrm{can}} & (S, L) & \\
((\nu a \cdot P, e)S, L) & \rightsquigarrow_{\mathrm{can}} & ((P, e[a \mapsto a'])S, L \cup \{a'\}) & \text{fresh } a' \\
((P \mid Q, e)S, L) & \rightsquigarrow_{\mathrm{can}} & ((P, e)(Q, e)S, L) &
\end{array}
$$

is confluent on states (up to α-conversion). The normal forms are canonical states.

$\mathrm{Can}(S, L)$ the normal from of (S, L) for this reduction.

A labeled transition system of states

Objects: canonical states.

Transitions labeled by pairs of labels, defined by

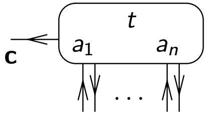
$$(([l]a(b_1 \dots b_n) \cdot P, e)(\overline{[m]a'}\langle b'_1 \dots b'_n \rangle \cdot P', e')S, L)$$
$$\xrightarrow{l\overline{m}} \mathrm{Can}((P, e[b_1 \mapsto e'(b'_1), \dots, b_n \mapsto e'(b'_n)])(P', e')S, L)$$

if $e(a) = e'(a')$.

General principle

The translation is not a function but a relation because we do not work up to associativity, commutativity. . . of (co)contraction: there are many different (co)contraction trees of the same arity.

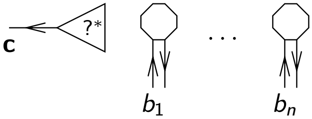
Given a repetition-free list $a_1, \ldots, a_n$ of names, $\mathcal{I}_{a_1,\ldots,a_n}$ is a relation from processes whose free names are in that list and simple nets of the shape



where **c** is an additional controle port.

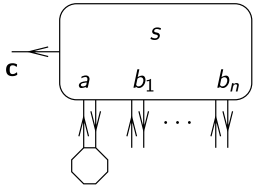
Empty process

nil $\mathcal{I}_{b_1,\ldots,b_n}$ t if t is of the shape



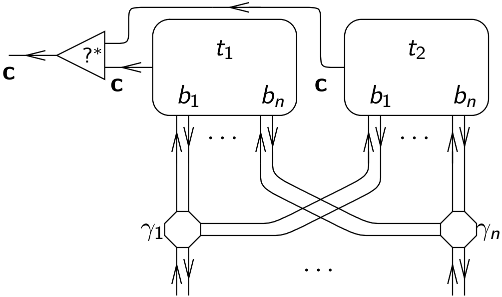
Name restriction

$\nu a \cdot P\, \mathcal{I}_{b_1,\ldots,b_n}\, t$ if there is s such that $P\, \mathcal{I}_{a,b_1,\ldots,b_n}\, s$ and t is of the shape



Parallel composition

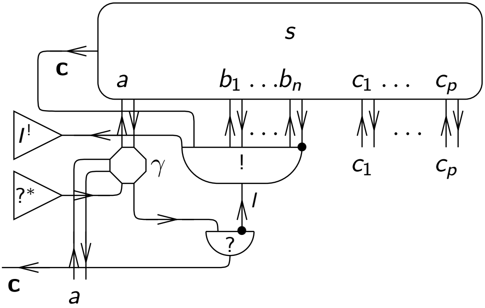
$P_1 \mid P_2 \, \mathcal{I}_{b_1,\ldots,b_n} \, t$ if t is



with $P_1 \, \mathcal{I}_{b_1,\ldots,b_n} \, t_1$, $P_2 \, \mathcal{I}_{b_1,\ldots,b_n} \, t_2$ and $\gamma_1, \ldots, \gamma_n$ are communication areas of order 1.

Input prefix

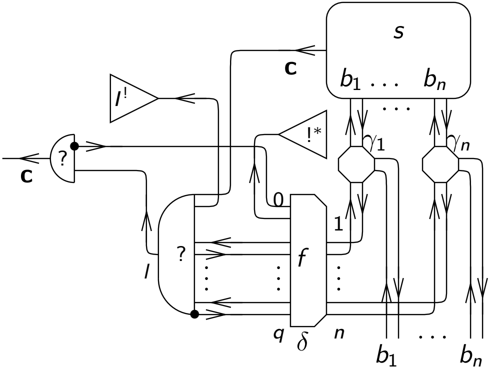
$[l]a(b_1 \dots b_n) \cdot P \; \mathcal{I}_{a,c_1,\dots,c_p} \; t$ if t is



with $P \; \mathcal{I}_{a,b_1,\dots,b_n,c_1,\dots,c_p} \; s$. Remember that a and the b_is are pairwise distinct.

Output prefix

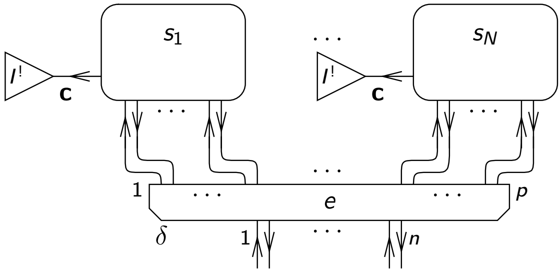
$\overline{[l]b_{f(0)}}\langle b_{f(1)} \dots b_{f(q)} \rangle \cdot P \, \mathcal{I}_{b_1,\dots,b_n} \; t$ if t is



with $P \, \mathcal{I}_{b_1,\dots,b_n} \; s$.

Translation of soups

$(P_1, e_1) \dots (P_N, e_N)\, \mathcal{I}_{b_1,\dots,b_n}\ t$ if t is



if $P_i\, \mathcal{I}_{c_{n_i+1},\dots,c_{n_{i+1}}}\ s_i$ (with $c_1, \dots, c_p$ a repetition free list containing all the free names of all P_is) and e such that $e_i(c_j) = b_{e(j)}$ for $n_i + 1 \leq j \leq n_{i+1}$.

Translation of states

$(S, L)\,\mathcal{I}_{b_1,\dots,b_n}\,t$ if $S\,\mathcal{I}_{c_1,\dots,c_p,b_1,\dots,b_n}$, $c_1,\dots,c_p$ is a repetition-free enumeration of L and t is s where communication areas of arity -1 have been plugged on the pairs of ports corresponding to the c_js.

The main result

$(S, L)\ \widetilde{\mathcal{I}}_{b_1,\dots,b_n}\ s$ if there exists a simple net s_0 such that $(S, L)\ \mathcal{I}_{b_1,\dots,b_n}\ s_0$ and $s_0 \sim_{\mathrm{d}} s$.

Theorem

The relation $\widetilde{\mathcal{I}}_{b_1,\dots,b_n}$ is a bisimulation from the labeled transition system of canonical states to the labeled transition system of simple nets.

Uses crucially the confluence of the reduction.

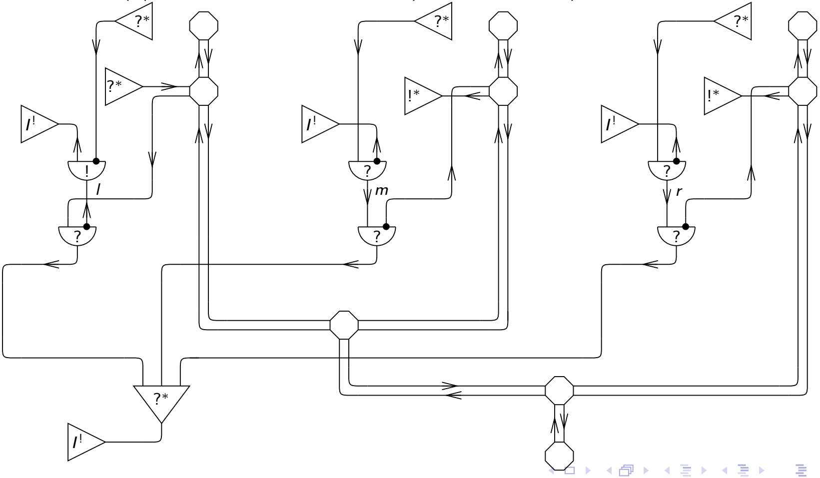
What this means

Assume that $(S, L)\ \widetilde{\mathcal{I}}_{b_1,\ldots,b_n}\ s$ and let $l, m \in \mathcal{L} \setminus \{\tau\}$.

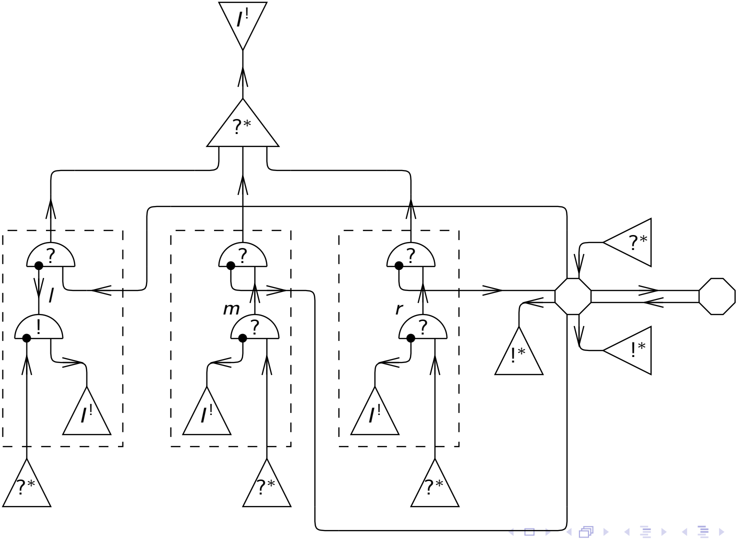
- If $(S, L) \xrightarrow{l\overline{m}} (T, M)$ then there is a simple net t such that $s \xrightarrow{l\overline{m}} t$ and $(T, M)\ \widetilde{\mathcal{I}}_{b_1,\ldots,b_n}\ t$.
- If $s \xrightarrow{l\overline{m}} t$ then there is a canonical state (T, M) such that $(S, L) \xrightarrow{l\overline{m}} (T, M)$ and $(T, M)\ \widetilde{\mathcal{I}}_{b_1,\ldots,b_n}\ t$.

Concurrent communication

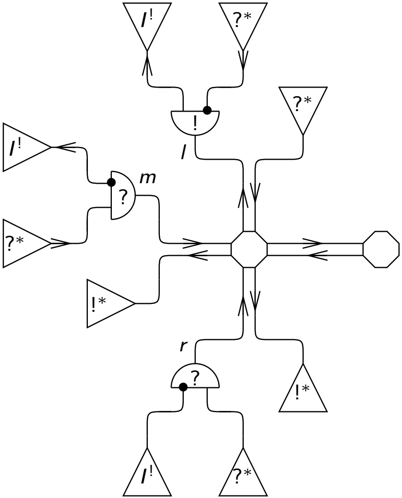
$$P = \nu a \cdot \Big(\big([l]a() \cdot \mathsf{nil} \mid \overline{[m]a}\langle\rangle \cdot \mathsf{nil} \big) \mid \overline{[r]a}\langle\rangle \cdot \mathsf{nil} \Big) \ \mathcal{I}\ s \text{ where } s \text{ is}$$



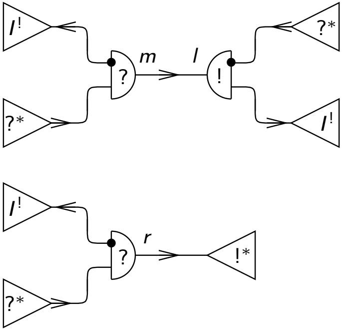
Applying aggregation of communication areas, we get



Applying the $\leadsto_{\mathrm{d}}$ reduction, we get



And this nets reduces to a sum of two nets, by the prefix/communication area interaction. One of these is

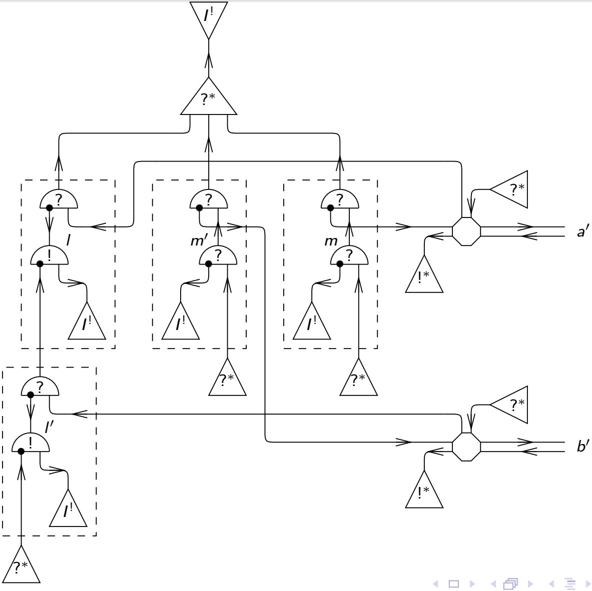


Sequentiality

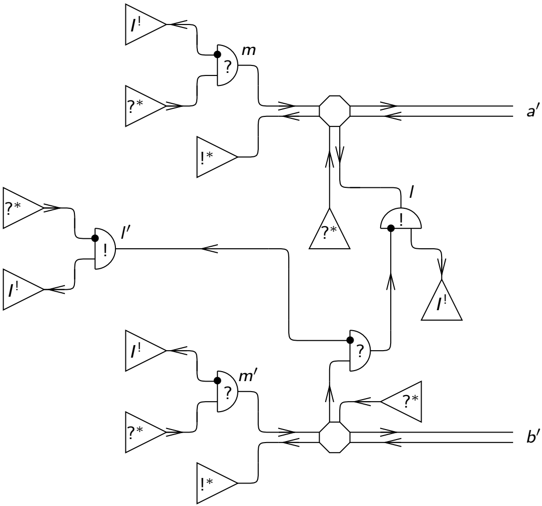
Let P be the process

$$[l]a'() \cdot [l']b'() \cdot \mathsf{nil} \mid \overline{[m']b'}\langle\rangle \cdot \mathsf{nil} \mid \overline{[m]a'}\langle\rangle \cdot \mathsf{nil}$$

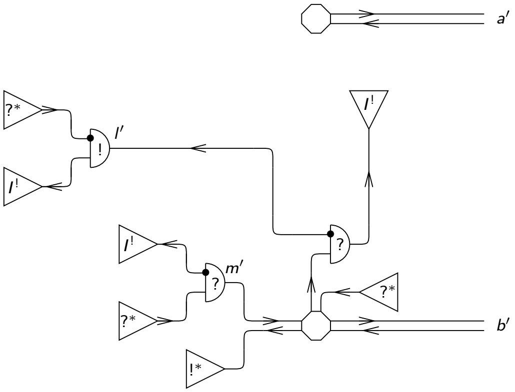
Then $P \, \mathcal{I}_b \, s$ where s reduces by aggregation to


$I^!$
$?^*$
?
!
l
m'
m
a'
b'
l'
$!^*$

which reduces by $\leadsto_{\mathrm{d}}$ to



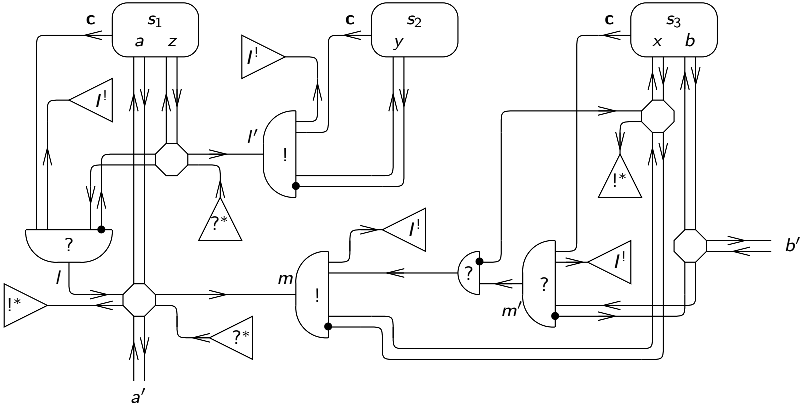
Which reduces to a sum $s_1 + \cdots$ where s_1 (and only s_1) contains a communication redex on l and m, and by reducing this redex, we get from s_1



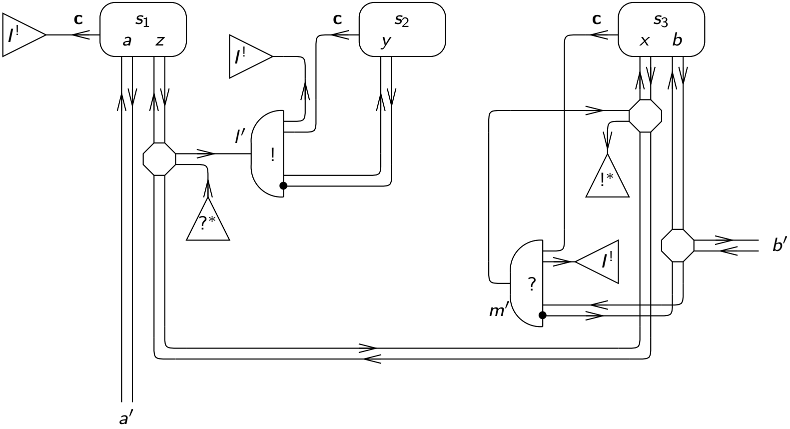
and only now it will be possible to reduce l'/m'.

Name passing

$\nu z \cdot \left(\overline{[l]a}\langle z\rangle \cdot P \mid [l']z(y) \cdot Q\right) \mid [m]a(x) \cdot \overline{[m']x}\langle b\rangle \cdot R$ translates to s which (up to some aggregations. . .) is



Then $s \xrightarrow{m\bar{l}} t \sim_{\mathrm{d}} t'$ where t' is



in which the names x and z are now identified (the corresponding communication areas are connected).

Finally $t' \xrightarrow{l'\overline{m'}} t''$ where t'' is