Differential interaction nets and processes

Thomas Ehrhard  Olivier Laurent

Preuves, Programmes et Systèmes

April 16, 2007
Show that differential interaction nets are sufficiently expressive for encoding faithfully a significant fragment of the $\pi$-calculus.

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

1. Differential interaction nets
   - Cells and nets
   - Reduction rules
   - A labeled transition system of simple nets
   - A toolbox for process interpretation

2. A finitary polyadic $\pi$-calculus
   - The calculus
   - Environment machine

3. Translation of states to nets
   - Translation of processes
   - Translation of states

4. A bisimulation theorem

5. Examples
A typing system

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

We set $i = o \perp$, so that $i = !o \otimes i$ and $o = ?i \otimes i$.

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

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:

\[ \begin{array}{c}
\text{\(\text{?}\)} \\
\text{\(\text{!}\)} \\
\end{array} \]

\[ \begin{array}{c}
\text{\(\triangleleft\)} \\
\text{\(\triangleright\)} \\
\end{array} \]

Constants:

\[ \begin{array}{c}
\text{\(\bot\)} \\
\text{\(1\)} \\
\end{array} \]
Exponential fragment

Dereliction, weakening and contraction:

Codereliction, coweakening and cocontraction:
A **simple net** is an interaction net made of these cells (respecting types), and of the forthcoming **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 $\circ$, we introduce a cell $s_!^{\circ}$, 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 $\tau$.

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

If, in a simple net, two of these labels are equal, they must be equal to $\tau$. 
Multiplicative reduction

\[
\begin{align*}
\text{?} & \rightarrow o \\
\otimes & \rightarrow o \\
\rightarrow o & \sim_m o \\
\rightarrow o & \sim_m \perp \\
? & \rightarrow o \\
! & \rightarrow 1 \\
? & \rightarrow o \\
? & \rightarrow \perp \\
! & \rightarrow \perp \\
\end{align*}
\]
Exponentials: deterministic reductions

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

\[
\begin{array}{c}
? \\
l \\
\rightarrow
\end{array}
\begin{array}{c}
?_l \\
\rightarrow
\end{array}
\begin{array}{c}
! \\
m \\
\rightarrow
\end{array}
\begin{array}{c}
\sim_{c,R} \\
l \\
\rightarrow
\end{array}
\]
Exponentials: deterministic reductions (continued)

The next deterministic reduction rules are the structural ones:

\[
\begin{align*}
?_l & \rightarrow !_l & \sim_s & !_l \\
? & \rightarrow ! & \sim_s & ! \\
!_o & \rightarrow ?_o & \sim_s & ? \\
! & \rightarrow ? & \sim_s & ? \\
?_l & \rightarrow s! & \sim_s & s! \\
? & \rightarrow s! & \sim_s & s!
\end{align*}
\]
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:

\[
\begin{align*}
\langle ? \rightarrow l ?l \rightarrow ! \rangle & \rightsquigarrow_{\text{nd}, R} 0 \\
\langle ! \rightarrow l !o \rightarrow ? \rangle & \rightsquigarrow_{\text{nd}, R} 0
\end{align*}
\]
Exponentials: non-deterministic reductions (continued)
Exponentials: promotion reduction

\[ \langle l \rightarrow \mathbf{?} l \mathbf{,} \mathbf{s}! \rangle \sim_b \mathbf{s} \]
Δ: the set of all nets, \( N\langle\Delta\rangle \): the set of all nets.

If \( R \subseteq \Delta \times N\langle\Delta\rangle \) is a rewriting relation, \( R^* \subseteq N\langle\Delta\rangle \times N\langle\Delta\rangle \) is the transitive closure of its “extension to sums”.

**Theorem**

Let \( R, R' \subseteq L \). Let \( R \subseteq \Delta \times N\langle\Delta\rangle \) be the union of some of the reduction relations \( \rightsquigarrow_{c,R}, \rightsquigarrow_{nd,R'}, \rightsquigarrow_m, \rightsquigarrow_s \) and \( \rightsquigarrow_b \). The relation \( R^* \) is confluent on \( N\langle\Delta\rangle \).

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

Particular reduction: \( \rightsquigarrow_R = \rightsquigarrow_m \cup \rightsquigarrow_{c,\{\tau\}} \cup \rightsquigarrow_s \cup \rightsquigarrow_b \cup \rightsquigarrow_{nd,R} \).
We set \( \rightsquigarrow_d = \rightsquigarrow_\emptyset \).
A labeled transition system $\mathcal{D}_\mathcal{L}$:

- objects: simple nets
- transitions labeled by pairs of labels

$s \xrightarrow{\{l, m\}} t$ if $s \xrightarrow{*} \{l, m\} s_1 + s_2 + \cdots + s_n$ 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 $s_i \xrightarrow{*} \{l, m\} s'$, none of the summands of $s'$ has such a communication redex.
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 $\tau$).

Codereliction-par cell defined dually.
Prefix cells

\textit{n-ary input} and \textit{n-ary output} prefix cells are

\[ \begin{array}{c}
\begin{array}{c}
\vdots \\
! \\
\vdots \\
\end{array}
\end{array} =
\begin{array}{c}
\begin{array}{c}
\vdots \\
? \\
\vdots \\
\end{array}
\end{array} \]

\[ \begin{array}{c}
\begin{array}{c}
\vdots \\
? \\
\vdots \\
\end{array}
\end{array} =
\begin{array}{c}
\begin{array}{c}
\vdots
\end{array}
\end{array} \]

where \( n \) is the number of pairs of auxiliary ports.
If the two prefix cells have the same arity, then one has

\[
\vdash_c \{l,m\} \quad u \vdash_0^\ast
\]

otherwise, the lefthand configuration reduces to 0 (but we can avoid this situation).
Let $I$ be the following “identity” net

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

\[ I! \xrightarrow{?} \sim^* \emptyset \]
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:
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:

\[
p + q \sim^* s \quad p + q
\]
Identification structures

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

![Diagram](image_url)

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

![Diagram](image_url)
Applying communication areas aggregation, we have:

\[ \cdots f \quad \rightsquigarrow_{a^*} \quad g \circ f \quad \cdots \]
Interaction between prefixes and communication areas
Syntax

\[ \mathcal{N} = \{a, b, a_1, \ldots \} \] a set of names.

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

The labels of a process must be distinct from \( \tau \) and pairwise distinct.
States of the machine

- Environment: finite partial function $e : \mathcal{N} \rightarrow \mathcal{N}$.
- 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 $P_i$s start with input or output prefixes.
Canonical form of a state

The reduction

\[(\text{nil}, e)S, L \xrightarrow{\text{can}} (S, L)\]
\[(\nu a \cdot P, e)S, L \xrightarrow{\text{can}} ((P, e[a \mapsto a'])S, L \cup \{a'\}) \text{ fresh } a'\]
\[(P \mid Q, e)S, L \xrightarrow{\text{can}} ((P, e)(Q, e)S, L)\]

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

\text{Can}(S, L) \text{ the normal form of } (S, L) \text{ for this reduction.}
A labeled transition system of states

Objects: canonical states.

Transitions labeled by pairs of labels, defined by

\[
((\llbracket l \rrbracket a(b_1 \ldots b_n) \cdot P, e)(\llbracket m \rrbracket a'(b_1' \ldots b_n') \cdot P', e') S, L) \\
\xrightarrow{[m]} \text{Can}((P, e[b_1 \mapsto e'(b_1'), \ldots, b_n \mapsto e'(b_n')])(P', e') S, L)
\]

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

\[ \begin{array}{c}
  a_1 & \ldots & a_n \\
  \text{c} & t & \\
  \ldots & \\
 \end{array} \]

where $\text{c}$ is an additional controle port.
nil $\mathcal{I}_{b_1,\ldots,b_n}$ $t$ if $t$ is of the shape

\[
\begin{align*}
\text{nil } & \mathcal{I}_{b_1,\ldots,b_n} \quad \text{if } t \text{ is of the shape} \\
& \begin{array}{cc}
\text{c} & \?^* \\
& \begin{array}{c}
\bullet \\
\downarrow \\
(b_1) \\
& \cdots \\
& (b_n)
\end{array}
\end{array}
\end{align*}
\]
\[ \nu a \cdot P \lhd_{b_1, \ldots, b_n} t \text{ if there is } s \text{ such that } P \lhd_{a, b_1, \ldots, b_n} s \text{ and } t \text{ is of the shape} \]

\[
\begin{array}{c}
\nu a \cdot P \lhd_{b_1, \ldots, b_n} \begin{array}{ccc}
  & & \cdots \\
  \text{a} & \text{b}_1 & \text{b}_n \\
\end{array} \\
\text{s} \\
\text{c}
\end{array}
\]
Parallel composition

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

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

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

\[ \gamma \]

with \( P \mathcal{I}_{a,b_1,\ldots,b_n,c_1,\ldots,c_p} s \). Remember that \( a \) and the \( b_i \)'s are pairwise distinct.
\[ \overline{[l]b_f(0)\langle b_f(1) \ldots b_f(q) \rangle} \cdot P \ I_{b_1,\ldots,b_n} \ t \text{ if } t \text{ is} \]

with \( P \ I_{b_1,\ldots,b_n} \ s \).
Translation of soups

\[(P_1, e_1) \ldots (P_N, e_N) I_{b_1, \ldots, b_n} t \text{ if } t \text{ is}

\[
\begin{array}{c}
\text{if } P_i I_{c_{n_i+1}, \ldots, c_{n_i+1}} s_i \text{ (with } c_1, \ldots, c_p \text{ a repetition free list containing all the free names of all } P_i\text{s) and } e \text{ such that } e_i(c_j) = b_{e(j)} \text{ for } n_i + 1 \leq j \leq n_i + 1. 
\end{array}
\]
Translation of states

\[(S, L) \mathcal{I}_{b_1, \ldots, b_n} t \text{ if } S \mathcal{I}_{c_1, \ldots, c_p, b_1, \ldots, b_n, c_1, \ldots, c_p} \text{ is a repetition-free enumeration of } L \text{ and } t \text{ is } s \text{ where communication areas of arity } -1 \text{ have been plugged on the pairs of ports corresponding to the } c_j \text{s.}\]
The main result

\[(S, L) \overset{\tilde{I}_{b_1,\ldots,b_n}}{\sim} s\] if there exists a simple net \(s_0\) such that
\[(S, L) I_{b_1,\ldots,b_n} s_0\] and \(s_0 \sim_d s\).

**Theorem**

*The relation \(\tilde{I}_{b_1,\ldots,b_n}\) 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 \((S, L) \leadsto b_1, \ldots, b_n s\) and let \(l, m \in L \setminus \{\tau\}\).

- If \((S, L) \xrightarrow{lm} (T, M)\) then there is a simple net \(t\) such that \(s \xrightarrow{lm} t\) and \((T, M) \leadsto b_1, \ldots, b_n t\).

- If \(s \xrightarrow{lm} t\) then there is a canonical state \((T, M)\) such that \((S, L) \xrightarrow{lm} (T, M)\) and \((T, M) \leadsto b_1, \ldots, b_n t\).
Concurrent communication

\[ P = \nu a \cdot \left( ([l]a() \cdot \text{nil} \mid \overline{[m]}a\langle \rangle \cdot \text{nil}) \mid \overline{[r]}a\langle \rangle \cdot \text{nil} \right) I s \text{ where } s \text{ is} \]
Applying aggregation of communication areas, we get
Applying the $\sim_d$ 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

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

Then $P \mathcal{I}_b s$ where $s$ reduces by aggregation to
Differential interaction nets
A finitary polyadic π-calculus
Translation of states to nets
A bisimulation theorem
Examples
which reduces by $\sim_d$ 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 $s_1$

\[
\begin{array}{c}
\text{and only now it will be possible to reduce $l'/m'$}.
\end{array}
\]
Name passing

\[ \nu z \cdot \left( [l]a\langle z \rangle \cdot P \mid [l']z(y) \cdot Q \right) \mid [m]a(x) \cdot [m']x\langle b \rangle \cdot R \text{ translates to} \]

\[ s \text{ which (up to some aggregations...)} \] is

![Diagram](image-url)
Then $s \xrightarrow{\overline{m}l} t \sim_d t'$ 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