On Context Semantics and Interaction Nets *

Matthieu Perrinel

ENS Lyon
Université de Lyon
LIP (UMR 5668 CNRS ENS Lyon UCBL INRIA)
matthieu.perrinel@ens-lyon.fr

Abstract

Context semantics is a tool inspired by Girard' s geometry of interaction. It has had many applications from study of optimal reduction to proofs of complexity bounds. Yet, context semantics have been defined only on λ-calculus and linear logic.

In order to study other languages, in particular languages with more primitives (built-in arithmetic, pattern matching,...) we define a context semantics for a broader framework: interaction nets. These are a well-behaved class of graph rewriting systems.

Here, two applications are explored. First, we define a notion of weight, based on context semantics paths, which bounds the length of reduction of nets. Then, we define a denotational semantics for a large class of interaction net systems.

Categories and Subject Descriptors F [*3*]: 2—Denotational semantics

Keywords interaction nets, geometry of interaction, context semantics, denotational semantics

1. Introduction

Context semantics (CS) is a tool related to geometry of interaction (GoI) [6, 11]. CS is a mean of studying the evaluation of a program (a λ-term or a proof-net of linear logic) by means of paths in the program. Those paths are defined by a token travelling across the program according to some rules. It has first been used to study optimal reduction in λ-calculus [11] and linear logic [12]. It has also been used for the design of interpreters for λ-calculus [16]. Finally, it has been used to prove complexity bounds on subsystems of System T [4] and linear logic [2, 5, 20]. For this latter application, an advantage of context semantics compared to the syntactic study of reduction is its genericity: some common results can be proved for different variants of linear logic, which allows to factor out proofs of complexity results for these various systems.

Since CS had many interesting developments in λ-calculus and linear logic, we would like to have a similar tool for programming languages. For instance, we want pattern-matching, inductive datatypes (as opposed to Church encoding) and built-in arithmetic operation. As the set of features needed is not precisely defined, a general framework of systems would be preferred to a single system. This way, we would need to define the CS and prove the general theorems only once, and they will stand for any system of the framework. The framework we chose is interaction nets [13].

Interaction nets are a model of asynchronous deterministic computation. They are based on rewriting rules on graphs and were inspired by the proof-nets of linear logic [10]. Interaction nets can, in particular, encode proof-nets [17] and λ-calculus [15]. Moreover, interaction nets are general enough to encode functional programming languages containing pattern-matching and built-in recursion [9]. A non-deterministic extension is powerful enough to encode the full π-calculus [18].

A net is a graph-like structure whose nodes are called *cells*. Each cell is labelled by a symbol. A library defines the set of symbols and the rewriting rules for the symbols. Thus, a library corresponds to a programming language. Interaction nets as a whole, correspond to a set of programming languages.

Contributions

In this paper, we define CS for any library and we show that the CS paths are stable along reduction. We present two applications of this CS:

- For any net N, we define a weight $W_N \in \mathbb{N}\cup\{\infty\}$ based on CS paths. We prove that if M reduces to N, then $W_N = W_M - 1$. Thus, if N normalizes, W_N is the length of the reduction path, else $W_N = \infty$. This could be used to prove complexity bounds on programming languages which are either defined or encodable in interaction nets.
- We define a notion of observational equivalence for each library. Then we define a denotational semantics which is, on a class of libraries named *crossing libraries*, sound and fully abstract with respect to our equivalence.

Related works

As CS is a model of GoI, the closest work to this paper, is the definition of a GoI for an arbitrary library by De Falco [7]. De Falco defines a notion of paths in nets and a notion of reduction of those paths. Then, he defines a GoI of a library as a weighing of paths by elements of a semi-group such that the weights are stable along reduction. However, he exhibits such a semi-group only for some particular libraries (based on linear logic). Thus, there is no complete GoI model of interaction nets yet.

Concerning our first application, we are not aware of other works aiming at proving complexity bounds on generic interaction nets. There are also few tools to analyze the semantics of generic libraries. Lafont defined an observational equivalence, based on

* Special thanks to Damiano Mazza and Patrick Baillot for fruitful discussions concerning this work.

CSL-LICS 2014, July 14–18, 2014, Vienna, Austria.
Copyright © 2014 ACM 978-1-4503-2886-9…$15.00.
http://dx.doi.org/10.1145/2603088.2603155

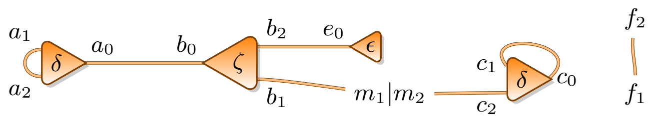


Figure 1: Net N. Names of ports and labels of cells are represented while names of cells are not.

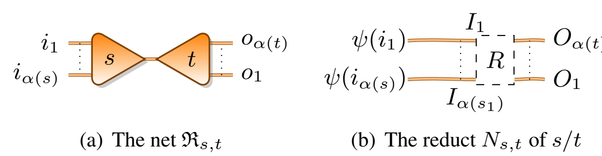


Figure 2: Interaction rule with explicit bijection ($O_k = \psi(o_k)$).

paths, for a special library called interaction combinators [14]. Then, he defines a GoI for interaction combinators: he assigns a weight to each path in the nets such that two nets are equivalent if and only if their paths have the same weights. Thus, the set of weights of paths is a denotational semantics sound and fully abstract for his equivalence.

In [19], Mazza designed an observational equivalence for every library. This equivalence is similar, but not equal to Lafont's on interaction combinators. Then, he defines a denotational semantics for symmetric combinators, a variant of interaction combinators [3, 19]. Symmetric combinators are Turing-complete and can encode a large class of libraries (called *polarized* libraries). However, as we will detail later, defining the semantics of a net as the semantics of its translation in interaction combinators does not give quite a good semantics. It would differentiate nets that behave similarly. Our definition of observational equivalence is strongly inspired from Mazza's.

Finally, in [8], Fernandez and Mackie define an observational equivalence for every library. This equivalence is stronger than Mazza's semantics on symmetric combinators but, in general, they are orthogonal.

2. Interaction nets

Interaction nets can be defined in many ways. Here, to define properly the CS paths, we had to use a formal definition.

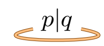
We fix a *symbol set* $S = (\mathcal{S}, \alpha)$ with $\mathcal{S}$ a countable set whose elements will be called *symbols* and α a mapping from $\mathcal{S}$ to $\mathbb{N}$ associating an *arity* to each symbol.

A net is a set of cells joined by wires. Wires may have one (or both) ends unattached. We will often connect nets, those connections are made by those unattached ends. Formally, the ends of wires will be represented by a set P^N of *ports*. There are three types of ports: ports attached to a cell (the set P_c^N), *free ports* (the set P_f^N) and *merging ports* (the set P_m^N).

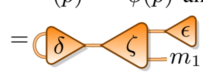
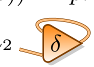
Definition 1. A *net* N is a tuple $(P^N, C^N, l^N, \sigma_w^N, \sigma_m^N, \sigma_c^N)$ with:

- $P^N = P_c^N \uplus P_f^N \uplus P_m^N$ is a finite set called set of *ports*.
- C^N is a finite set whose elements will be called *cells*
- $l^N : C^N \to \mathcal{S}$ labels each cell with a symbol.
- σ_w^N is an involution on P^N with no fixpoint. We also write $\overline{p}$ for $\sigma_w^N(p)$. σ_w^N represents the wires: if there is a wire between the ports p and p', then $\overline{p} = p'$ and $\overline{p'} = p$.
- σ_m^N is an involution on P_m^N with no fixpoint. This mapping associates two merging ports.
- σ_c^N is a bijection from P_c^N to $\{(c,i) | c \in C^N, 0 \leq i \leq \alpha(l^N(c))\}$. σ_c^N represents the cells.

Example 1. Let $S_{comb} = \{\zeta, \delta, \epsilon\}$ be symbols with $\alpha(\zeta) = \alpha(\delta) = 2$ and $\alpha(\epsilon) = 0$. Then, Figure 1 represents the net N with: $P^N = \{a_0, a_1, a_2, b_0, b_1, b_2, c_0, c_1, c_2, e_0, f_1, f_2, m_1, m_2\}$, $C^N = \{A, B, C, E\}$, $l^N = \{A \mapsto \delta, B \mapsto \zeta, C \mapsto \delta, E \mapsto \epsilon\}$, $\sigma_w^N = \{a_1 \leftrightarrow a_2, a_0 \leftrightarrow b_0, b_2 \leftrightarrow e_0, b_1 \leftrightarrow m_1, m_2 \leftrightarrow c_2, c_1 \leftrightarrow c_0, f_1 \leftrightarrow f_2\}$, $\sigma_m^M = \{m_1 \leftrightarrow m_2\}$ and $\sigma_c^N = \{a_0 \mapsto (A,0), a_1 \mapsto (A,1), a_2 \mapsto (A,2), b_0 \mapsto (B,0), b_1 \mapsto (B,1), b_2 \mapsto (B,2), c_0 \mapsto (C,0), c_1 \mapsto (C,1), c_2 \mapsto (C,2), e_0 \mapsto (E,0)\}$.

The merging ports are introduced for technical reasons but are not essential. Let p, q be merging ports of a net N such that $p \neq \overline{q}$. Let N' be the net equal to N where $\overline{p}$ — $p|q$ — $\overline{q}$ is replaced by $\overline{p}$ — $\overline{q}$, then we write $N \to_m N'$ and we say that p is *merged with* $\overline{q}$. We define the equivalence relation $\leftrightarrows_m$ as the reflexive symmetric transitive closure of $\to_m$. The nets will be considered up to $\leftrightarrows_m$ equivalence and α-equivalence (renaming of the ports and cells). Notice that $\to_m$ is confluent and strongly normalizing, we will usually represent a net by its $\to_m$ normal form (the only merging ports are the cycles of shape ).

Let c be a cell of N. We write $p_i(c)$ the port p such that $\sigma_c^N(p) = (c,i)$. The *principal port* of c denotes $p_0(c)$. If $i \geq 1$, $p_i(c)$ is called the *i-th auxiliary port* of c.

The interaction between two nets is done by merging some of their free ports. This operation is called *gluing* and will be the main tool to define the dynamics of nets. Let M and N be nets and ϕ be a partial injection from P_f^M to P_f^N, then $M \bowtie_\phi N$ is the net whose ports and cells are those of M and N, the free ports in the domain and codomain of ϕ become merging nodes with $\sigma_m^{M \bowtie_\phi N}(p) = \phi(p)$ and $\sigma_m^{M \bowtie_\phi N}(\phi(p)) = p$. For instance, let $M =$  and $N = m_2$  f_1 — f_2 and $\phi = \{m_1 \mapsto m_2\}$, then $M \bowtie_\phi N$ is the net of Figure 1.

The computation in interaction nets is done by reduction of *active pairs*. An active pair is a set of two cells linked by their principal ports. *Libraries* will define which pairs of symbols can interact. When an active pair is labelled by symbols which can interact together, we may reduce it: those cells are replaced by a net $N_{s,t}$ which only depends on the symbols of the active pair. The rest of the interaction net is left untouched.

Definition 2. Let $s, t \in \mathcal{S}$, $\mathfrak{R}_{s,t}$ is the net of Figure 2a.

An interaction rule for (s,t) is a tuple (R, ψ) where R is a net and ψ is a bijection from $P_f^{\mathfrak{R}_{s,t}}$ to P_f^R. For $1 \leq j \leq \alpha(s)$, we name I_j the edge $\overline{\psi(i_j)}$ of R. For $1 \leq j \leq \alpha(t)$, we name O_j the edge $\psi(o_j)$ of R, as in Figure 2b.

In practice, we will describe interaction rules by displaying an active pair and the reduct linked by an arrow as in Figure 3. The bijection is given implicitly by the position of the ports.

Definition 3 (library)**.** A *library* for the symbol set $(\mathcal{S}, \alpha)$ is a partial mapping L on $\mathcal{S} \times \mathcal{S}$. To each (s_1, s_2) in the domain of L, L associates an interaction rule for (s_1, s_2). Let us suppose that $L(s_1, s_2) = (R, \psi)$. Then we require that $L(s_2, s_1)$ is defined and equal to the *symmetric* of $L(s_1, s_2)$ where inputs and outputs are switched, i.e. $L(s_2, s_1) = (R, \psi \circ \{i_k \leftrightarrow o_k\})$. Reduction $\to$ is defined by $N \bowtie_\phi \mathfrak{R}_{s_1,s_2} \to (N \bowtie_{\psi \circ \phi} R)$.

Because of the symmetry condition, the rules shown in Figure 3 are enough to describe the whole library L_{comb} of symmetric

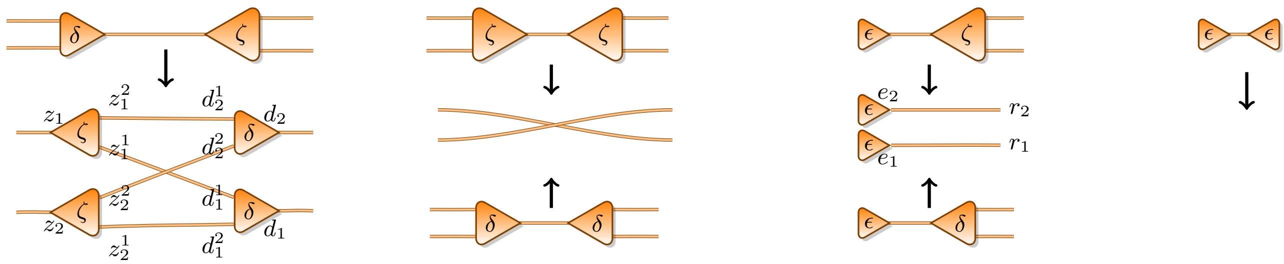


Figure 3: The symmetric combinators, library L_{comb} for S_{comb}

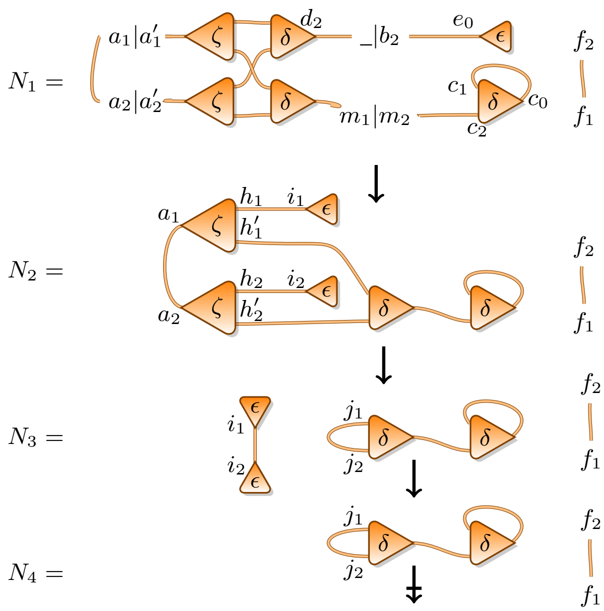


Figure 4: Example of reduction with the library L_{comb}.

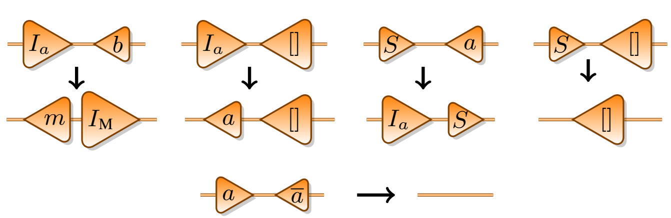
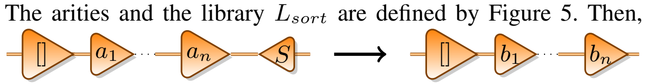


Figure 5: L_{sort} library. The rules stand for any $a, b \in A$, $m = \min(a, b)$ and $M = \max(a, b)$

combinators. The net of Figure 1 successively reduces to the nets of Figure 4 (note that we use the notation _ to denote an object whose name and value has no importance).

Example 2. As another example, let us consider an ordered set $(A, \leq)$, and the symbols $\{S, []\} \cup A \cup \{I_a | a \in A\} \cup \{\overline{a} | a \in A\}$. The arities and the library L_{sort} are defined by Figure 5. Then,



with $[b_1; \cdots; b_n]$ the sorted list corresponding to $[a_1; \cdots; a_n]$. More precisely, it is an implementation of insertion sort.

3. Context semantics

In this section, we fix a library L. For any (s_1, s_2) in the domain of L, we write $(N_{s_1,s_2}, \phi_{s_1,s_2}) = L(s_1, s_2)$. This section uses many lists. Lists are written in the form $[a_1; \cdots; a_n]$, $l_1 @ l_2$ represents the concatenation of l_1 and l_2, . represents "push" ($[a_1; \cdots; a_n].b = [a_1; \cdots; a_n; b]$) and $|l|$ is the length of l.

Let N be a net, we will represent the ports that can appear during the reduction of N by objects called *potential ports*. However, the definition of potential ports may be difficult to grasp. Therefore, we first present informally a notion of *potential net* to guide the intuition on potential ports. The potential net of N aims to represent all the cells and ports that can appear during the reduction of N. The potential net of N is a tree of nets of root N such that, if the cell c labelled by s will interact with a cell c' labelled by s', the net $N_{s',s}$ (which replaces the active pair c, c' during reduction) is stacked on c. As an example, we present in Figure 6 a part[1] of the potential nets of N (Figure 1) and N_1 (Figure 4). A potential port can be understood as an address of a port in a potential net:

The set Pot^N of *potential ports* of net N is the set of lists $[(p_0, N); (p_1, N_{s_1,t_1}); \cdots; (p_k, N_{s_k,t_k})]$ such that for each i: p_i is a port of N_{s_i,t_i} and p_{i-1} is the principal port of a cell labelled by t_i. For instance, in Figure 6, the potential ports of N $[(b_0, N)]$, $[(b_0, N); (d_2, N_{\delta,\gamma})]$ and $[(b_0, N); (d_2, N_{\delta,\gamma}); (e_1, N_{\epsilon,\delta})]$ point to the ports of the potential net of N they represent. For $P.(p, N') \in Pot^N$, we set $\overline{P.(p, N')} = P.(\overline{p}, N')$. Note that $\overline{P.(p, N')}$ corresponds to the port wired with $P.(p, N')$ in the potential net of N.

We can notice that when we reduce a net, it flattens its potential net. Moreover, if N is a net in normal form, then the potential net of N is equal to N (the root has no child). We will define paths in potential nets. Those paths will be stable by reduction, thus they will give us information about the reduction of N. Concretely, we define *contexts* as tuples (P, T) with P a potential port and T a *trace*. Then we define a relation $\mapsto$ on contexts. The trace represents information about the beginning of the $\mapsto$ path, we need this information for the $\mapsto$ paths to be stable by reduction. In Figure 6, we represent (by thick arrows) the path $([(b_2, N)], []) \mapsto ([(a_0, N)], [(\zeta, 2)]) \mapsto ([(a_0, N); (d_2, N_{\zeta,\delta})])$ on the potential net of N and its reduction $([(b_2, N_1)], []) \mapsto ([(d_2, N_1)], [])$.

A *positive trace element* is (s, i) with $s \in \mathcal{S}$ and $1 \leq i \leq \alpha(s)$. The meaning of (s, i) is "I have crossed a cell of symbol s, from its i-th auxiliary port to the principal port". A positive trace is a list of positive trace elements. The set of positive traces is written Tra^+.

A *negative trace element* is $\overline{(s, i)}$ with $s \in \mathcal{S}$ and $1 \leq i \leq \alpha(s)$. The meaning of $\overline{(s, i)}$ is "I will arrive at the principal port of a cell of symbol s. When this happens I will choose to leave it by its i-th auxiliary port". A *trace element* is either a positive trace element or

[1] On the complete potential net of N, $N_{\zeta,\zeta}$ should also be stacked on the lower ζ cells of $N_{\delta,\zeta}$, they were omitted for the sake of clearness.

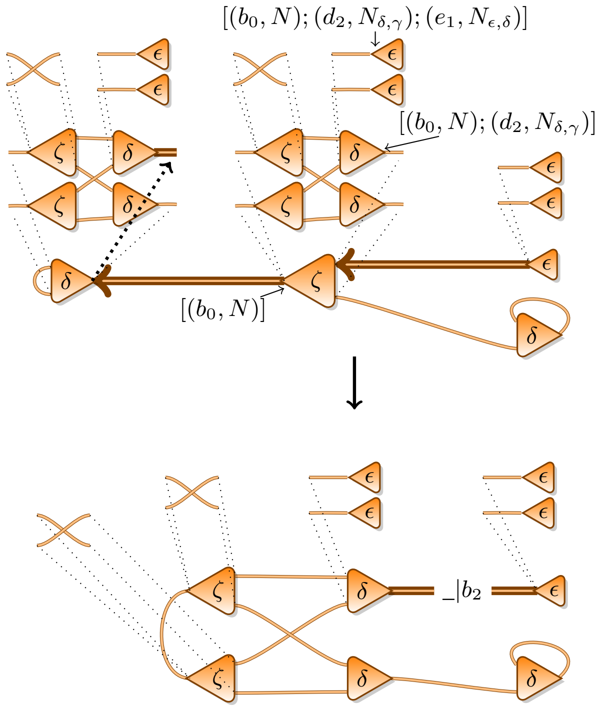


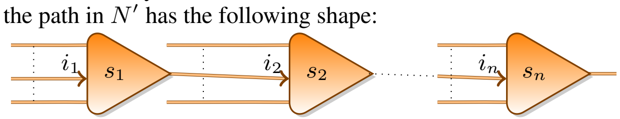
Figure 6: Intuitive representation of potential ports.

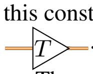
a negative trace element. A *trace* is a list of trace elements. The set of traces is written Tra.

The set of *contexts* of N_1 is $Cont^{N_1} = Pot^{N_1} \times Tra$.

Definition 4. For any net N, we define a relation $\mapsto$ on $Cont^N$ by the rules of Figure 7. In those rules, we suppose $s, s' \in \mathcal{S}$, $c, c' \in C^N$, $l^N(c) = s$, $l^N(c') = s'$, $1 \leq k \leq \alpha(s)$, $1 \leq k' \leq \alpha(s')$ and $m, m' \in P^N_m$ with $\sigma^N_m(m) = m'$.

The intuition underlying the definition of $\mapsto$ is that, if $(P, []) \mapsto^* (Q, [(s_1, i_1); \cdots; (s_n, i_n)])$ then there is a path in the potential net of N from P to Q such that: N reduces to a net N' and the reduct of the path in N' has the following shape:



Let us notice that the trace is transformed into a net consisting of a line of cells labelled by the symbols of the trace, the wire linking the cells according to the indices of the trace. We will use this construction again, representing the net corresponding to T by .

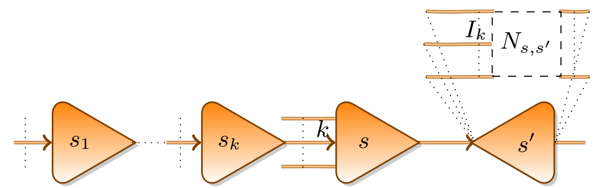
The $\mapsto$ relation is deterministic and incomplete (there are contexts C such that $C \not\mapsto$, i.e. $\forall D \in Cont^N, \neg(C \mapsto D)$). Let $C = (P, T) \in Cont^N$, the possible context D such that $C \mapsto D$ is defined depending on the rightmost port p of P.

If p is an auxiliary port, we cross the cell and add the information on the trace (rule a).

If p is a merging port, we cross the merging port (rule f).

If p is a principal port, the behaviour depends on whether the rightmost trace element t is positive or negative (if the trace is empty, $C \not\mapsto_L$):

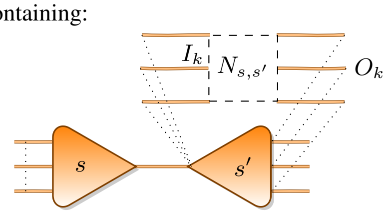
- If t is positive (rule c), then $t = (s, k)$ it corresponds to an active pair $\{c, c'\}$ of symbols $\{s, s'\}$. According to the intuition we gave of $\mapsto$, N reduces to a net N' and the reduct of the path in N' has the following shape:



So, the cell c' is reduced with a cell c labelled by s. So, by definition of the potential net, $N_{s,s'}$ is stacked on c'. We can jump to the port I_k of $N_{s,s'}$ without breaking our invariant because I_k will be merged with $p_k(c)$ during the reduction.

- Else if t is negative (rule b), then $t = \overline{(s', k')}$. According to the intuitive meaning we gave to negative trace elements, we have to leave c' by its k'-th auxiliary port. We will see below how negative trace elements can appear.

If p is free, we are in the net $N_{s,s'}$ corresponding to the interaction of the future active pair $\{c, c'\}$.So N reduces to a net N' containing:



The behaviour depends on whether p is a $O_{k'}$ (rule d) or a $\overline{I_k}$ (rule e). In the first case, we know that during reduction $O_{k'}$ will be merged with $\overline{p_{k'}(c')}$ so we may move the token to $p_{k'}$ without breaking the invariant. In the second case, we know that I_k will be merged with $\overline{p_k(c)}$. However, we do not know where is c. We know that c will form an active pair with c' in N', but c and c' can be very far apart in the potential net of N. How can we find $\overline{p_k(c)}$? The idea is to use the intuition underlying the $\mapsto$ relation: we know that c and c' will form an active pair, so $(\overline{p_0(c')}, []) \mapsto^* (p_0(c), [])$. According to rule b, $(\overline{p_0(c')}, [\overline{(s,k)}]) \mapsto^* (p_0(c), [\overline{(s,k)}]) \mapsto (\overline{p_k(c)}, [])$. Thus, to find $\overline{p_k(c)}$, we move the context to $\overline{p_0(c')}$ with $\overline{(s,k)}$ on the trace.

Let us recall that we consider the interaction nets up to α-conversion and port merging. So we need to verify that the relation is the same on equivalent nets. The names of the ports play no role in the definition of the $\mapsto$ relation, so $\mapsto$ is the same on α-equivalent nets (up to the renaming of the ports in the potential ports of the contexts). We can verify that whenever $([(p_1, M_1); \cdots; (p_k, M_k)], T) \mapsto^* ([(q_1, N_1); \cdots; (q_l, N_l)], U)$, with:

- For all $1 \leq i \leq k$, p_i is not a merging port and there exists a net M'_i such that $M_i \leftrightarrows_m M'_i$.
- For all $1 \leq j \leq l$, q_j is not a merging port and there exists a net N'_j such that $N_j \leftrightarrows_m N'_j$.

Then, the following path is valid:

$$([(p_1, M'_1); \cdots; (p_k, M'_k)], T) \mapsto^* ([(q_1, N'_1); \cdots; (q_l, N'_l)], U)$$

Example 3. The following $\mapsto$-path in the net N of Figure 1, goes from the principal port of E to the δ cell which will form an active pair with E. Notice that this δ cell does not exist

$$
\begin{array}{llllll}
a) & (P.(p_k(c),N) & ,T &) \mapsto & (P.(\overline{p_0(c)},N) & ,T.(s,k)) \\
b) & (P.(p_0(c),N) & ,T.\overline{(s,k)} &) \mapsto & (P.(\overline{p_k(c)},N) & ,T) \\
c) & (P.(p_0(c'),N) & ,T.(s,k) &) \mapsto & (P.(p_0(c'),N).(I_k,N_{s,s'}) & ,T) \\
d) & (P.(p_0(c'),N).(O_{k'},N_{s,s'}) & ,T &) \mapsto & (P.(\overline{p_{k'}(c)},N) & ,T) \\
e) & (P.(p_0(c'),N).(\overline{I_k},N_{s,s'}) & ,T &) \mapsto & (P.(\overline{p_0(c)},N) & ,T.\overline{(s,k)}) \\
f) & (P.(m,N) & ,T &) \mapsto & (P.(\overline{m'},N) & ,T)
\end{array}
$$

Figure 7: Rules of context-semantics

yet (it will be created by the ζ/δ reduction): $([(b_2,N)],[]) \mapsto ([(a_0,N)],[(\zeta,2)]) \mapsto ([(a_0,N);(I_2,N_{\zeta,\delta})],[])$.

Example 4. As a more involved example, we will study in the net N, the path between the two ϵ cells created during the δ/ϵ step of reduction (first step of Figure 4).

$$
\begin{aligned}
&([(e_0,N);(r_1,N_{\delta,\epsilon})],[]) \mapsto ([(b_2,N)],[\overline{(\delta,1)}]) \\
&\mapsto ([(a_0,N)],[\overline{(\delta,1)};(\zeta,2)]) \mapsto ([(a_0,N);(d_2,N_{\zeta,\delta})],[\overline{(\delta,1)}]) \\
&\mapsto ([(a_0,N);(z_1^2,N_{\zeta,\delta})],[]) \mapsto ([(a_0,N);(O_1,N_{\zeta,\delta})],[(\zeta,2)]) \\
&\mapsto ([(\overline{a_1}=a_2,N)],[(\zeta,2)]) \mapsto ([(b_0,N)],[(\zeta,2);(\delta,2)]) \\
&\mapsto ([(b_0,N);(z_2,N_{\delta,\zeta})],[(\zeta,2)]) \\
&\mapsto ([(b_0,N);(z_2,N_{\delta,\zeta});(I_2=O_2,N_{\zeta,\zeta})],[]) \\
&\mapsto ([(b_0,N);(d_2^2,N_{\delta,\zeta})],[]) \mapsto ([(b_0,N);(\overline{d_2},N_{\delta,\zeta})],[(\delta,2)]) \\
&\mapsto ([(e_0,N)],[(\delta,2)]) \mapsto ([(e_0,N);(e_2,N)],[])
\end{aligned}
$$

We wrote that $\mapsto$ simulates the reduction of the net. We will prove that the $\mapsto$-paths are stable by reduction. Formally, if $N \to N'$, we will define a projection Π from the potential ports of N to potential ports of N' so that $(P,T) \mapsto^* (Q,U) \Leftrightarrow (\Pi(P),T) \mapsto^* (\Pi(Q),U)$.

In this section, we suppose that $N \to N'$ by reducing the active pair $\{c_1,c_2\}$ labelled by s_1,s_2. We set $(R_1,\phi_1) = L(s_2,s_1)$ and $(R_2,\phi_2) = L(s_1,s_2)$. So $N = N_0 \bowtie_{\phi_2} \mathfrak{R}_{s_1,s_2}$ and $N' = N_0 \bowtie_{\psi\circ\phi_2} R_2$. We define a mapping Π from Pot^N to $Pot^{N'}$ which depends on the leftmost port p:

- If $p \in P^{N_0}$, we set $\Pi([(p,N)]@P) = [(p,N')]@P$.
- If $p = p_0(c_i)$ for $i \in \{1,2\}$. We set:

$$\Pi([(p_0(c_i),N);(r,R_i)]@P) = [(r,N')]@P$$

- Otherwise, Π is undefined.

The two next propositions show that the projection behaves as expected. Lemma 1 shows that the paths are preserved along reduction. It requires the potentials P and Q to be in the domain of the projection, this condition is the counterpart of the "long enough" condition on paths in GoI settings [7].

Proposition 1. Let $P' \in Pot^{N'}$, then there exists $P \in Pot^N$ such that $\Pi(P) = P'$.

Proposition 2. Let $P \in Pot^N$ such that $\Pi(P)$ is defined, then $\Pi(\overline{P})$ is defined and $\overline{\Pi(P)} = \Pi(\overline{P})$.

Lemma 1. If $T,U \in Tra$, $P,Q \in Pot^N$, $\Pi(P) = P'$ and $\Pi(Q) = Q'$ then $(P,T) \mapsto^* (Q,U) \Rightarrow (P',T) \mapsto^* (Q',U)$

If $T,U \in Tra$, $P',Q' \in Pot^{N'}$ and $(P',T) \mapsto^+ (Q',U)$ then there exists P,Q such that $\Pi(P) = P'$, $\Pi(Q) = Q'$ and $(P,T) \mapsto^+ (Q,U)$

Proof. We will only prove the first statement. The proof of the second is quite similar.

We will prove it for minimal such paths: let us suppose that $(P,T) \mapsto^* (Q,U)$ and that for every other context (R,V) in the path, $\Pi(R)$ is undefined. We will show that $(P',T) \mapsto^* (Q',U)$. Then, the lemma is straightforward because any $\mapsto^*$ path between potentials in the domain of Π can be decomposed in such smaller paths.

We set $[(p,N)]@P_1 = P$ and $[(q,N)]@Q_1 = Q$.

- If $p,q \in P^{N_0}$, then $(P,T) \mapsto (Q,U)$ (the path has length 1), $P' = [(p,N')]@P_1$ and $Q' = [(q,N')]@Q_1$. Given that all $\mapsto$ rules are local, and that ports of N_0 are unaffected by the reduction, $(P',T) \mapsto (Q',U)$.
- If p and q belong respectively to P^{R_i} and P^{R_j} with $i,j \in \{1,2\}$, then a careful observation of the $\mapsto$ rules shows that the only possibility is $i = j$ and $(P,T) \mapsto (Q,U)$. Because $\Pi(P)$ and $\Pi(Q)$ are defined, we have $P = [(p_0(c_i),N);(r,R_i)]@P_1$ and $Q = [(p_0(c_i),N);(s,R_i)]@Q_1$ for some $r,s \in P^{R_i}$. So $P' = [(r,N')]@P_1$ and $Q' = [(s,N')]@Q_1$. Considering that $\mapsto$ is local, $(P',T) \mapsto (Q',U)$.
- If $p \in P^{N_0}$ and $q \in P^{R_i}$ (with $i \in \{1,2\}$, we will write $j = 3-i$ to refer to the other cell), then the only possibility is that p is a free port of N_0 which, in N, is merged with the free port i_k of $\mathfrak{R}_{s_1,s_2}$ and, in N', is merged with the free port $\psi(i_k)$ of R_i. So, we have $(P,T) = ([(p,N)],T) \mapsto ([(p_k(c_j),N)],T) \mapsto ([(p_0(c_i),N)],T.(s_j,k)) \mapsto ([(p_0(c_i),N);(I_k,R_i)],T) = (Q,U)$. We can notice that $P' = [(p,N')]$ and $Q' = [(I_k,N')]$. We get $(P',T) \mapsto ([(\overline{\sigma_m^{N'}(p)},N')],T) = (Q',U)$.
- If $p \in P^{R_i}$ (with $i \in \{1,2\}$, we will write $j = 3-i$) and $q \in P^{N_0}$, then $\overline{q}$ is a free port of N_0 which, in N, is merged with a free port of $\mathfrak{R}_{s_1,s_2}$ and, in N', is merged with a free port $\psi(i_k)$ of R_i. And, either $p = \overline{I_k} = \psi(i_k)$ (and $1 \le k \le \alpha(s_j)$) or $p = O_k$ (and $1 \le k \le \alpha(s_i)$).
  - If $p = \overline{I_k}$, $(P,T) = ([(p_0(c_i),N);(\overline{I_k},R_i)],T) \mapsto ([(p_0(c_j),N)],T.\overline{(s_j,k)}) \mapsto ([(\overline{p_k(c_j)},N)],T) \mapsto ([(q,N)],T) = (Q,U)$. We observe that $P' = [(\psi(i_k),N')]$ and $Q' = [(q,N')]$. In N', $\psi(i_k)$ is merged with $\overline{q}$ so $(P',T) \mapsto (Q',T) = (Q',U)$.
  - If $p = O_k$, $(P,T) = ([(p_0(c_i),N);(O_k,R_i)],T) \mapsto ([(\overline{p_k(c_i)},N)],T) \mapsto ([(q,N)],T) = (Q,U)$. We can notice that $P' = [(O_k,N')]$ and $Q' = [(q,N')]$. In N', O_k is merged with $\overline{q}$ so $(P',T) \mapsto (Q',U)$.

$\square$

In particular, the successive projections of free ports of a net will always be defined along a reduction sequence. So a path between two free ports of a net will always be stable along reduction, as stated by Corollary 1.

Corollary 1. If $M \to^* N, p, q \in P_f^M$ and $T, U \in Tra$, then

$$([(\overline{p}, M)], T) \mapsto^* ([(q, M)], U) \Leftrightarrow ([(\overline{p}, N)], T) \mapsto^* ([(q, N)], U)$$

Let Π_1, Π_2, Π_3 and Π_4 be the projections corresponding to the reduction steps of Figures 1 and 4. If $\Pi_1([(e_0, N); (r_1, R_{\delta,\epsilon})]) = [(e_0, N_1); (r_1, R_{\delta,\epsilon})]$, then $\Pi_2([(e_0, N_1); (r_1, R_{\delta,\epsilon})]) = [(h_1, N_2)]$, next $\Pi_3([(h_1, N_2)]) = [(i_2, N_3)]$ and $\Pi_4([(i_2, N_3)])$ is not defined.

The path $([(e_0, N); (r_1, R_{\delta,\epsilon})], []) \mapsto^{13} ([(e_0, N); (e_2, N)], [])$ reduces to $([(e_0, N_1); (r_1, R_{\delta,\epsilon})], []) \mapsto^2 ([(d_2, N_1)], [\overline{(\delta, 1)}]) \mapsto^4 ([(\overline{a'_2}, N_1)], [(\zeta, 2)]) \mapsto^5 ([(e_0, N); (e_2, N)], [])$ in N_1, then $([(h_1, N_2)], []) \mapsto^3 ([(i_2, N_2)], [])$ in N_2 and $([(i_2, N_3)], []) \mapsto^0 ([(i_2, N_3)], [])$ in N_3.

4. Context semantics for complexity bounds

In this section, we define *canonical cells*, which are the potential ports which correspond to cells that will really appear during reduction. Then we use the canonical cells to define a weight $W_N \in \mathbb{N} \cup \{\infty\}$ for any net N such that, if $M \to N$, then $W_M \geq W_N + 1$. It follows that the length of any reduction sequence from M is bounded by W_M. Notice that it is not true that $W_M > W_N$ because if $W_M = \infty$, then $W_N = \infty$.

The approach is inspired by Dal Lago's context semantics for linear logic [5]. First, Dal Lago's weight allowed to show that every proof-net of some linear logic subsystem verified complexity properties (e.g. every proof-net of LLL reduces in polynomial time w.r.t the size of the argument, whatever the reduction strategy). These bounds were previously known, but Dal Lago's proofs were much shorter. Then, his tool was used to prove strong bounds which were previously unknown [1, 20]. We hope that our tool will lead to similar results.

We want to capture the "cells which will appear during reductions beginning by N". Such a cell is either a cell of N, or appears during the reduction of two cells c_1 and c_2 such that: c_1 and c_2 both appear during reductions beginning by N, and $\{c_1, c_2\}$ will form an active pair. This is the intuition behind the following definition of canonical cells.

Definition 5. We define the set Can^N of canonical cells of N by induction:

- For every cell c of N, $[(p_0(c), N)]$ is a canonical cell
- If $P_1.(p_0(c_1), N_1)$ is a canonical cell, $(P_1.(\overline{p_0(c_1)}, N_1), []) \mapsto (P_2.(p_0(c_2), N_2), [])$, $l^N(c_1) = s_1$, $l^N(c_2) = s_2$ and $L(s_1, s_2)$ is defined. Then for every cell c of N_{s_2,s_1}:

$$P_1.(p_0(c_1), N_1).(p_0(c), N_{s_2,s_1}) \in Can^N$$

Lemma 2. Let us suppose that $N \to_L N'$ by reducing the active pair $\{c_1, c_2\}$ and Π is the associated projection.

If $P \in Can^N$, then either $\Pi(P)$ is defined and $\Pi(P) \in Can^{N'}$ or P corresponds to one of the ports of the active pair: $P \in \{[(p_0(c_1), N)], [(p_0(c_2), N)]\}$.

If $\Pi(P)$ exists and is in $Can^{N'}$, then $P \in Can^N$.

Example 5. Let us consider the net N of Figure 1. We can show that $C_1 = [(e_0, N); (e_1, R_{\delta,\epsilon})]$ is a canonical cell. Indeed, b_0 is a principal port of N so $[(b_0, N)]$ is a canonical cell. We know that $([(\overline{b_0}, N)], []) \mapsto^0 ([(a_0, N)], [])$ and $L(\zeta, \delta)$ is defined so $[(b_0, N); (d_2, N_{\delta,\zeta})]$ is a canonical cell. Finally, $([(b_0, N); (\overline{d_2}, N_{\delta,\zeta})], []) \mapsto^1 ([(e_0, N)], [])$ and $L(\delta, \epsilon)$ is defined so $[(b_0, N); (d_2, N_{\delta,\zeta}); (e_1, R_{\epsilon,\delta})]$ is canonical.

Similarly, $C_2 = [(a_0, N); (d_2, N_{\zeta,\delta}); (e_1, N_{\epsilon,\delta})]$ and $C_3 = [(e_0, N); (e_1, N_{\delta,\epsilon})]$ are canonical. Let Π_1, Π_2, be the projections corresponding to $N \to N_1$ and $N_1 \to N_2$ (Figures 1 and 4). We can observe that $\Pi_2 \circ \Pi_1(C_1) = \Pi_2 \circ \Pi_1(C_2) = \Pi_2 \circ \Pi_1(C_3)$ so, intuitively, there are three canonical cells corresponding to the same future cell.

The following theorem corresponds to the main result of [5]. The intuition behind it is that each reduction step erases two canonical potentials: the ones corresponding to the active pair.

Theorem 1. For every interaction-net N, the length of any interaction sequence beginning by N is equal to:

$$T_N = \sum_{P \in C^N} \frac{1}{2^{|P|}}$$

Proof. We suppose that N reduces to N' by reducing the active pair $\{c, d\}$ labelled by s, t, Π is the associated projection and D its domain. For any $P' \in Can^{N'}$,

- Either $P' = [(p', N')]@Q$ with p' a port of $R_{s,t}$ then p' is also a port of $R_{t,s}$ (or vice versa). So, $\Pi^{-1}(P')$ is equal to $\{[(p_0(c), N); (p', R_{t,s})]@Q, [(p_0(d), N); (p', R_{s,t})]@Q\}$.
- Or $P' = [(p', N')]@Q$ and $\Pi^{-1}(P') = \{[(p', N)]@Q\}$.

So, for any $P' \in Can^{N'}$, we have:

$$\sum_{P \in \Pi^{-1}(P')} \frac{1}{2^{|P|}} = \frac{1}{2^{|P'|}}$$

This gives the following equations:

$$T_N = \sum_{P \in C^N \cap D} \frac{1}{2^{|P|}} + \sum_{P \in C^N - D} \frac{1}{2^{|P|}}$$

$$T_N = \sum_{P' \in C^{N'}} \frac{1}{2^{|P'|}} + \frac{1}{2^{|[(p_0(c), N)]|}} + \frac{1}{2^{|(p_0(c), N)]|}}$$

$$T_N = T_{N'} + 1$$

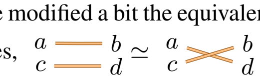
□

However it seems we need further tools (corresponding to the notion of copies, acyclicity of proof-nets and subtree properties in [5]) to ease the use of Theorem 1 to prove bounds for interaction nets system. This is left for future work.

5. Context semantics as a denotational semantics

5.1 Observational equivalence

Corollary 1 shows us that the paths from a free port to a free port are stable along the reduction. Hence, it seems natural to define a denotational semantics based on those paths. We would like our semantics to enjoy a *full abstraction* property, i.e. a theorem stating that two interaction nets have the same semantics if and only if they are *observationally equivalent*.

Let us recall that, in general, two programs P and Q are said observationally equivalent if for all contexts $C[\ ]$, such that the execution of $C[P]$ outputs some value v, the execution of $C[N]$ outputs the same value v[2]. In a framework as general as interaction nets, there are several possible notions of "outputting a value", each gives a different observational equivalence. The observational equivalence $\approx$ we will consider is based on an observational equivalence $\simeq$ defined by Mazza [19]. We modified a bit the equivalence, because in some farfetched libraries, . In our point of view, interaction nets are about "what can interact with

[2] Notice that the word "context" is not used here in our meaning of "token travelling through the net", but in the usual meaning of a "program with a hole".

what". So, if in a net a can only interact with b, it can not be equivalent to a net where a can only interact with d. In every system studied by Mazza in [19], the property $(N_1 \approx N_2) \Leftrightarrow (N_1 \simeq N_2)$ holds.

Both observational equivalences are based on *observable paths*. Let N be an interaction net, an observable path of N is a sequence $p_0, p_1, \cdots, p_k$ of ports of N such that we do not cross active pairs (if p_i is an auxiliary port, for $i < j \leq k$, p_j is not a principal port) and for every $i < k$:

- If $p_i = p_j(c)$ (with $j > 0$), then $p_{i+1} = \overline{p_0(c)}$ (crossing a cell from an auxiliary port to the principal port).
- If $p_i = p_0(c)$, then either there exists $j > 0$ such that $p_{i+1} = \overline{p_j(c)}$ (crossing a cell from the principal port to an auxiliary port) or $p_{i+1} = \overline{p_0(c)}$ (bouncing on a principal port).
- If $p_i \in P_m^N$, $p_{i+1} = \overline{\sigma_m^N(p_i)}$ (crossing a merging port).

The observable and $\mapsto$-path are closely linked. If $(P_1, T_1) \mapsto \cdots \mapsto (P_n, T_n)$, and $[(p_1, N)], \cdots, [(p_k, N)]$ is the subsequence of $P_1, \cdots, P_n$ of potentials of length 1, then $p_1, \cdots, p_k$ is an observable path. In fact the observable paths which can be obtained in this way are exactly the observable paths which can not be eliminated by reduction.

Let p, q be free ports of N. If $N \to^* N'$ and there exists an observable path from $\sigma_w^{N'}(p)$ to q, then we write $N \Downarrow_q^p$.

Definition 6 (observational equivalence)**.** Let N_1, N_2 be nets with $P_f^{N_1} = P_f^{N_2}$, then we write $N_1 \approx N_2$ if for all nets N, ϕ partial injection from $P_f^{N_1}$ to P_f^N, and $p, q \in P_f^{N_1 \bowtie_\phi N}$:

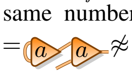
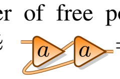
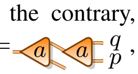
$$(N_1 \bowtie_\phi N) \Downarrow_q^p \;\Leftrightarrow (N_2 \bowtie_\phi N) \Downarrow_q^p$$

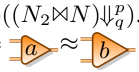
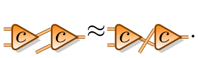
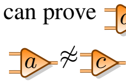
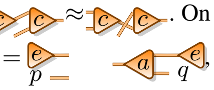
We wrote that our definition is inspired by Mazza's observational equivalence. Mazza defines $N_1 \simeq N_2$ as: for every net N and ϕ partial injection from $P_f^{N_1}$ to P_f^N,

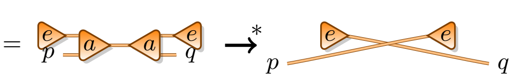
$$\exists p, q \in P_f^{N_1 \bowtie_\phi N} (N_1 \bowtie_\phi N) \Downarrow_q^p \Leftrightarrow \exists p, q \in P_f^{N_2 \bowtie_\phi N} (N_2 \bowtie_\phi N) \Downarrow_q^p$$

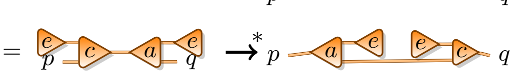
We can notice that $(N_1 \approx N_2) \Rightarrow (N_1 \simeq N_2)$. However, the other implication is not true in general.

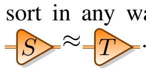
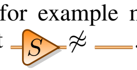
Example 6. Let us define the library L (resp. L_e) whose symbols are $\{a, b\}$ (resp. $\{a, b, c, e\}$), the reduction rules are given in Figure 8. One can observe that c duplicates every cell, e erases every cell, the other interactions (a/a, a/b and b/b) create wires between the free ports (b/b also creates a cycle).

In the library L, for any net N and $p \in P_f^N$, there exists $q \in P_f^N$ such that $N \Downarrow_q^p$. So, for any N_1 and N_2 with the same number of free ports, $N_1 \simeq N_2$. On the contrary, $N_1 =$  $\not\approx$  $= N_2$. Indeed, let $N =$ , then $(N_1 \bowtie N) \Downarrow_q^p$ and $\neg((N_2 \bowtie N) \Downarrow_q^p)$.

In L_e, we can prove  and . On the contrary, . Indeed, let $N =$ , then $\neg\left((N_1 \bowtie N) \Downarrow_q^p\right)$ and $(N_2 \bowtie N) \Downarrow_q^p$ as we can observe by reduction:

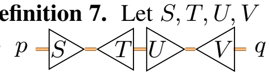
$N_1 \bowtie_\phi N =$ 

$N_2 \bowtie_\phi N =$ 

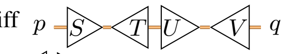
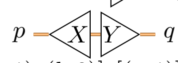
Finally, if we extended the library L_{sort} with another cell T performing sort in any way (for example merge sort), then we would have . But .

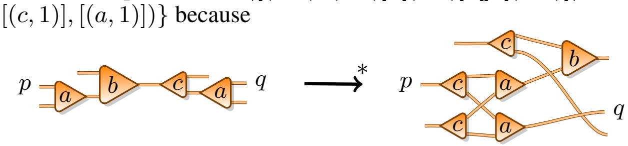
5.2 Definition of a denotational semantics

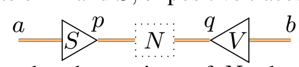
To define a denotational semantics matching our observational equivalence $\approx$, we need a mapping $|\_,\_,\_,\_|$ from $(Tra^+)^4$ to set of pairs of positive traces.

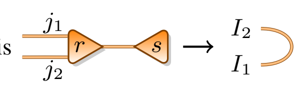
Definition 7. Let $S, T, U, V \in Tra^+$ and let us define the net N as , then we define $|S, T, U, V|$ as

$$\left\{ (X, Y) \in (Tra^+)^2 \middle| \exists P \in Can^N, \begin{array}{l} (P, []) \mapsto^* ([(p, N)], X) \\ (\overline{P}, []) \mapsto^* ([(q, N)], Y) \end{array} \right\}$$

We have $(X, Y) \in |S, T, U, V|$ iff  reduces to a net N' such that  is a subnet of N'. For example, in L_e, $|[(a, 1); (b, 2)], [(c, 1)], [], [(a, 2)]| = \{([(c, 1)], [(a, 1)])\}$ because



The interpretation $[N]$ of a net will be the set of unordered pairs $\{\!\{(p, S), (q, V)\}\!\}$ with p, q free ports of N and S, V positive traces such that, if we define M as the net  then $M \Downarrow_b^a$. So, $[N]$ corresponds to the observations of N when glued with a net consisting of only two lines of cells. Thus, the full abstraction of the semantics means "If for every net N, $N_1 \bowtie N$ and $N_2 \bowtie N$ have the same observations, then they have the same observations when N consists of two lines of cells." Thus, the proof of the full abstraction offers no real difficulty.

The soundness means that "If whenever N consists of two lines of cells, $N_1 \bowtie N$ and $N_2 \bowtie N$ have the same observations, then this is also true for an arbitrary net N". In fact, soundness is not true in the general case. However, we did prove soundness in the case of *crossing* libraries. A library is said *bouncing* if there is an interaction rule (R, ψ) and free ports $\overline{I_k}, \overline{I_l}$ of R such that $R \Downarrow_{\overline{I_l}}^{\overline{I_k}}$. A typical bouncing rule is . A crossing library is a library which is not bouncing. For the rest of the paper, we consider that L is crossing.

Definition 8. Let N be an interaction net, $[N]$ is the set

$$\left\{ \begin{array}{l} \{\!\{(p, S), (q, V)\}\!\} \\ p, q \in P_f^N \\ S, V \in Tra^+ \end{array} \middle| \exists \begin{array}{l} P \in Pot_+^N \\ T, U \in Tra \end{array}, \begin{array}{c} (P, []) \mapsto^* ([(p, N)], T) \\ (\overline{P}, []) \mapsto^* ([(q, N)], U) \\ |S, T, U, V| \neq \varnothing \end{array} \right\}$$

Where $\{\!\{(p, S), (q, V)\}\!\}$ represents a multiset (unordered pair in this case).

5.3 Stability of [_] by reduction and gluing

Theorem 2. If $N \to N'$, then $[N] = [N']$

Proof. Follows from Lemma 1 and the definition of $[N]$. $\square$

The proof of stability of [ ] by gluing is the most complex of this paper. It is necessary to prove the soundness of [ ] with respect to $\approx$. The proof requires the following lemmas.

Lemma 3. $\bigcup_{(Z,W) \in |[],[],V,Y|} |X, T, U, Z| \sim |X, T, U@V, Y|$

Lemma 4. $\bigcup_{(X,Y) \in |[],[],U,V|} |S, X@T, Y, V| \sim |S, T, U, Z@V|$

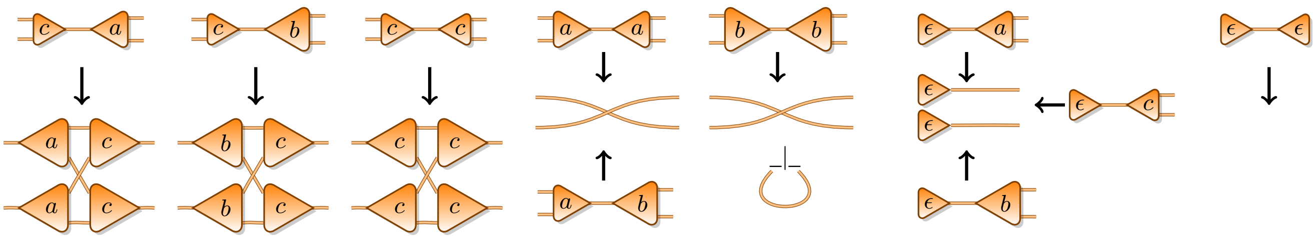
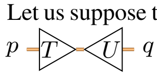
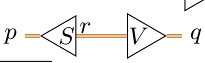
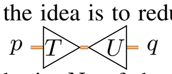


Figure 8: Reduction rules for libraries L and L_e

Lemma 5. Let $P, Q \in Pot^N$ and $T, U \in Tra$,

$$\left.\begin{array}{l}(P,[]) \mapsto (R,T)\\ (Q,[]) \mapsto (\overline{R},U)\\ |[],[],T,U| = (S,V)\end{array}\right\} \Rightarrow \exists R' \in Pot^N,\ \begin{array}{l}(R',[]) \mapsto (\overline{P},S)\\ (\overline{R'},[]) \mapsto (\overline{Q},V)\end{array}$$

Sketch of the proof. Let us suppose that $|[],[],T,U| = (S,V)$, then by definition $N = $  reduces to a net N' containing . So $([(r,N')],[]) \mapsto ([(p,N')],S)$ and $(\overline{[(r,N')]},[]) \mapsto ([(q,N')],V)$. By Lemma 1, there exists some potential port R of N such that $(R,[]) \mapsto ([(p,N')],S)$ and $(\overline{R},[]) \mapsto ([(q,N')],V)$.

The lemma is stated for arbitrary potential ports P, Q and R. In this case, the idea is to reduce the net until we reach a net N_1 of the shape  then we can apply the above reasoning to get paths in N_1 of the shape $(R_1,[]) \mapsto ([(p,N_1)],S)$ and $(\overline{R_1},[]) \mapsto ([(q,N_1)],V)$. Finally, we use Lemma 1 to get the paths in N. □

Theorem 3. Let M_1, N_1, M_2, N_2 be nets such that $[M_1] = [N_1]$, $[M_2] = [N_2]$ and ϕ an injection from $P_f^{M_1} = P_f^{M_2}$ to $P_f^{N_1} = P_f^{N_2}$, then $[M_1 \bowtie_\phi N_1] = [M_2 \bowtie_\phi N_2]$.

Proof. For concision, we will write $G_1 = M_1 \bowtie_\phi N_1$ and $G_2 = M_2 \bowtie_\phi N_2$. We will not consider the $\mapsto_m$ normal versions of R_1 and R_2 but will leave the merging ports created on the connecting ports (the ports in the domain or codomain of ϕ) untouched. We need a notion of $\mapsto$-path with a bounded number of alternations between ports of M_1 and ports of N_1. For every $i \in \mathbb{N}$, we define a relation $\rightarrowtail_i$ on $Cont^{G_1}$ by: $(P,T) \rightarrowtail_i (Q,U)$ iff we are in one of those cases:

$$\left\{\begin{array}{l} i = 0 \text{ and } (P,T) = (Q,U)\\ (P,T) \mapsto ([(p,G_1)],V) \rightarrowtail_{i-1} (Q,U) \text{ with } p \in P_f^{M_1} \cup P_f^{N_1}\\ (P,T) \mapsto (R,V) \rightarrowtail_i (Q,U) \text{ with } R \notin [((P_f^{M_1} \cup P_f^{N_1}), G_1)]\end{array}\right.$$

We define the $\rightarrowtail_i$ relations on $Cont^{G_2}$ similarly. We will prove the following property $\mathcal{P}(i+j)$ by induction on $i+j$:

"Let $p, q \in P_f^{M_1} \cup P_f^{N_1}$ and $P_1 \in Pot^{G_1}$ such that $(P_1,[]) \rightarrowtail_i ([(p,G_1)],T_1)$, $(\overline{P_1},[]) \rightarrowtail_j ([(q,G_1)],U_1)$ and $|S,T_1,U_1,V|$ is defined, then there exists $P_2 \in Pot^{G_2}$ such that $(P_2,[]) \mapsto^* ([(p,G_2)],T_2)$, $(\overline{P_2},[]) \mapsto^* ([(q,G_2)],U_2)$ and $|S,T_2,U_2,V|$ is defined."

This directly implies that $[G_1] \subseteq [G_2]$ because $P_f^{G_1} \subseteq P_f^{M_1} \cup P_f^{N_1}$. Because the roles of (M_1,N_1) and (M_2,N_2) are symmetrical, it will imply $[G_2] \subseteq [G_1]$ so $[G_1] = [G_2]$.

Let us suppose that $\mathcal{P}(i+j-1)$ is true. Let $p, q, r \in P_f^{M_1} \cup P_f^{N_1}$ and $P_1 \in Pot^{G_1}$ such that $(P_1,[]) \rightarrowtail_i ([(r,G_1)],U_1^r) \rightarrowtail_1 ([(p,G_1)],T_1^p)$, $(\overline{P_1},[]) \rightarrowtail_j ([(q,G_1)],T_1^q)$ and $|X,T_1^q,T_1^p,Y|$ is defined.

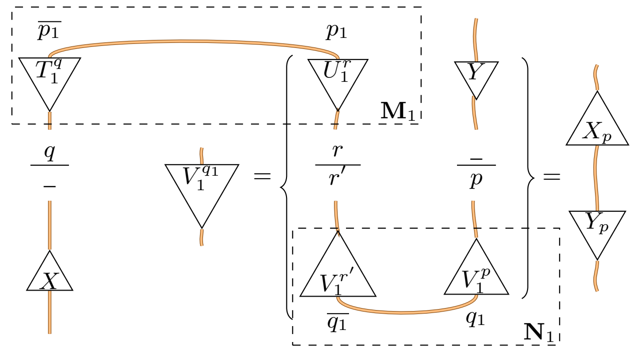


Figure 9: Sketch of the net G_1 in the proof of Theorem 3.

Without loss of generality, we will suppose that $r \in P_f^{M_1}$. Thus, $r \in P_m^{G_1}$, let us write $r' = \phi(r) = \sigma_m^{G_1}(r)$. Then, $([(r,G_1)],U_1^r) \mapsto ([(\overline{r'},G_1)],U_1^r) \rightarrowtail_1 ([(p,G_1)],T_1^p)$.

We reduce M_1 and N_1 to nets M_1' and N_1' such that, if we write $G_1' = M_1 \bowtie_\phi N_1$, $\Pi(P_1)$ has shape $[(p_1,G_1')]$ and the paths $([(p_1',G_1')],[]) \rightarrowtail_i ([(r,G_1')],U_1^r)$, $([(\overline{p_1'})],[]) \rightarrowtail_j ([(q,G_1')],T_1^q)$ and $([(r,G_1')],U_1^r) \rightarrowtail_1 ([(p,G_1)],T_1^p)$ do not cross active pairs. The net G_1' is sketched in Figure 9.

The path $([(\overline{r'},G_1')],U_1^r) \rightarrowtail_1 ([(p,G_1')],T_1^p)$ does not cross active pairs so the potential ports are first principal ports, then auxiliary ports and finally the free port $[(p,G_1')]$. Let $[(q_1,G_1')]$ be the first non-principal potential port of length 1 of the path. We have $([(p,G_1')],[]) \mapsto^* ([(q_1,G_1')],V_1^{q_1}) \mapsto^* ([(p,G_1')],V_1^{q_1}@V_1^p)$ with $T_1^p = V_1^{q_1}@V_1^p$.

We supposed that L is crossing, so there exists $V_1^{r'} \in Tra^+$ such that $([(q_1,G_1')],[]) \mapsto^* ([(r',G_1')],V_1^{r'})$ and $|[],[],V_1^{r'},U_1^r| = ([],V_1^{p_1})$.

By Lemma 3, there exists $(Y^p,X^p) \in |[],[],V_1^p,Y|$, such that $|X,T_1^q,U_1^r,Y^p@V_1^{r'}|$ is not empty. By induction hypothesis, there exists $P_2 \in Pot^{G_2}$ such that $(P_2,[]) \mapsto^* ([(r,G_2)],U_2^r)$, $(\overline{P_2},[]) \mapsto^* ([(q,G_2)],T_2^q)$ and $|X,T_2^q,U_2^r,Y^p@V_1^{r'}|$ is defined.

By Lemma 3, there exists $(Y^q,X^q) \in |X,T_2^q,[],[]|$, such that $|X^q,[],U_2^r,Y^p@V_1^{r'}| \neq \varnothing$. But $|X^q,[],U_2^r,Y^p@V_1^{r'}| = |X^q@U_2^r,V_1^{r'},[],Y^p|$ so, by Lemma 3 $|X^q@U_2^r,V_1^{r'},V_1^p,Y| \neq \varnothing$. We know that $[N_1'] = [N_1] = [N_2]$ so there exists some $Q_2 \in Can^{N_2}$ such that $(Q_2,[]) \mapsto^* ([(p,N_2)],V_2^p)$, $(\overline{Q_2},[]) \mapsto^* ([(r',N_2)],V_2^{r'})$ and $|X^q@U_2^r,V_2^{r'},V_2^p,Y|$ is defined.

By Lemma 4, there exists $(W_2^p,W_2^q) \in |[],[],V_2^{r'},U_2^r|$ such that $|X^q,W_2^q,W_2^p@V_2^p,Y| \neq \varnothing$. By Lemma 3, we can deduce that $|X,W_2^q@T_2^q,W_2^p@V_2^p,Y| \neq \varnothing$. From Lemma 5, there exists some potential port $R_2 \in Can^{G_2}$ such that $(R_2,[]) \mapsto^* (\overline{\overline{Q_2}},W_2^p) \mapsto^* ([(p,G_2)],W_2^p@V_2^p)$ and $(\overline{R_2},[]) \mapsto^* (\overline{P_2},W_2^q) \mapsto^* ([(q,G_2)],W_2^q@T_2^q)$. □

5.4 Soundness and full abstraction

Lemma 6. If $P, Q \in Can^N$ and $(P,T) \mapsto^* (Q,U)$, then we can reduce N to a net N' such that Π is the associated composition of projections, $\Pi(P)$ and $\Pi(Q)$ have shape $[(p,N')]$ and $[(q,N')]$, and the path $([(p,N')],T) \mapsto^* ([(q,N')],U)$ does not cross active pairs

Proof. We prove it by induction on $|P|+|Q|$. If $|P|+|Q| = 2$ and the path crosses an active pair, then we can reduce the pair. Notice that the path $([(p,N')],T) \mapsto^* ([(q,N')],U)$ is strictly shorter than the path $(P,T) \mapsto^* (Q,U)$. So we get the result after finitely many such reductions.

Else, $P = P_1.(p_0(c),N_1).(r,R_{s,t})$ and $(P_1.(\overline{p_0(c)},N_1),[]) \mapsto^* ([P_2.(p_0(d),N_2)],[])$ (with $l^N(c) = s$ and $l^N(d) = t$). By induction hypothesis, we can reduce N so that this path does not cross active pairs. So this path has length 0, $\{c,d\}$ becomes an active pair that we can reduce. Then $|\Pi(P)| < |P|$ and $|\Pi(Q)| \le Q$, so we can apply the induction hypothesis. ☐

Lemma states that if $[N_1] = [N_2]$ then the observations (the $(N_1) \Downarrow^p_q$) on N_1 and N_2 are the same. As we proved that $[\ ]$ is stable by context, we will get that if $[N_1] = [N_2]$, for any N, the observations on $N_1 \bowtie N$ and $N_2 \bowtie N$ are the same. This is exactly the soundness of $[\ ]$ with respect to $\approx$.

Lemma 7. If $[N_1] = [N_2]$ and $p, q \in P_f^{N_1} = P_f^{N_2}$, then:

$$N_1 \Downarrow^p_q \Leftrightarrow N_2 \Downarrow^p_q$$

Proof. We consider the $\to_m$-normal representations of N_1 and N_2. Notice that N_1 and N_2 play symmetric roles so we only need to prove one implication. Let us suppose that $N_1 \Downarrow^p_q$, then there exists some net N_1' such that $N_1 \to^* N_1'$ and there exists an observable path in N_1' from $\overline{p}$ to q.

By definition, the observable path is a (possibly empty) sequence of principal ports followed by a (possibly empty) sequence of auxiliary ports and the free port q. Let us consider r, the first port of the path which is not a principal port.

Then there is an observable path from r to q with only auxiliary ports (except q which is free), and there is an observable path from $\overline{r}$ to p with only auxiliary ports (except p which is free). Thus there exists $T_1, U_1 \in Tra^+$ such that $([(\overline{r},N_1')],[]) \mapsto^* ([(p,N_1')],T_1)$ and $([(r,N_1')],[]) \mapsto^* ([(q,N_1')],U_1)$. We can notice that $|[],T_1,U_1,[]|$ is defined.

We know that $(p,q,[],[]) \in [N_1'] = [N_1] = [N_2]$. Thus, there exists $Q \in Can^{N_2}$, $T_2, U_2 \in Tra^+$ such that $(\overline{Q},[]) \mapsto^* ([(p,N)],T_2)$ and $(Q,[]) \mapsto^* ([(q,N)],U_2)$. Thanks to Lemma 6, we know that we can reduce N_2 to a net N_2' such that the projection of Q has shape $[(s,N_2')]$ and the paths $([(\overline{s},N_2')],[]) \mapsto^* ([(p,N)],T_2)$ and $([(s,N_2')],[]) \mapsto^* ([(q,N)],U_2)$ do not cross active pairs.

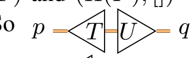
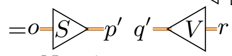
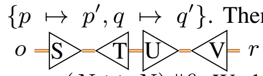
Thus, in N_2', there are observable paths from $\overline{s}$ to p and from s to q with only auxiliary ports. This means that there is an observable paths, in N_2', from $\overline{p}$ to q. So $N_2 \Downarrow^p_q$. ☐

Theorem 4 (soundness)**.** If $P_f^{N_1} = P_f^{N_2}$ and $[N_1] = [N_2]$, then $N_1 \approx N_2$

Proof. Let us consider a net N and ϕ a partial injection from $P_f^{N_1}$ to P_f^N and $p, q \in P_f^{N_1 \bowtie_\phi N}$, we need to prove that $(N_1 \bowtie_\phi N) \Downarrow^p_q \Leftrightarrow (N_2 \bowtie_\phi N) \Downarrow^p_q$.

By Theorem 3, we know that $[N_1 \bowtie_\phi N] = [N_2 \bowtie_\phi N]$. So, the result is given by Lemma 7. ☐

Theorem 5 (full abstraction)**.** If $P_f^{N_1} = P_f^{N_2}$ and $N_1 \approx N_2$, then $[N_1] = [N_2]$

Proof. Let us consider $\{\!\{(p,S),(q,V)\}\!\} \in [N_1]$, we will prove that $\{\!\{(p,S),(q,V)\}\!\} \in [N_2]$. We know that there exists $P \in Can^N$ and $T, U \in Tra^+$ such that $(P,[]) \mapsto^* ([(p,N_1)],T)$, $(\overline{P},[]) \mapsto^* ([(q,N_1)],U)$ and $|S,T,U,V| \neq \varnothing$. We use Lemma 6 to reduce N_1 to a net N_1' such that the projection $|\Pi(P)| = 1$ and the paths $(\Pi(P),[]) \mapsto^* ([(p,N_1)],T)$ and $(\overline{\Pi(P)},[]) \mapsto^* ([(q,N_1)],U)$ do not cross active pairs. So  is a subterm of N_1'. We set $N =$  and $\phi = \{p \mapsto p', q \mapsto q'\}$. Then $N_1 \bowtie_\phi N$ reduces to a net which has  as a subnet. We know that $|S,T,U,V| \neq \varnothing$ so $(N_1 \bowtie_\phi N) \Downarrow^o_r$. We know that $N_1 \approx N_2$ so $(N_2 \bowtie_\phi N) \Downarrow^o_r$. Thus, $N_2 \bowtie_\phi N$ reduces to a net N_2' with an observable path from o to r. We consider s the first port of the path which is not a principal port. Then $([(s,N_2')],[]) \mapsto^* ([(o,N_2')],T_2)$ and $([(\overline{s},N_2')],[]) \mapsto^* ([(r,N_2')],U_2)$. By Lemma 1, we know that s is the projection of $Q \in Can^{N_2 \bowtie_\phi N}$ and that the paths exist in $N_2 \bowtie_\phi N$. Those paths begin in N_2 and end N, let us consider the traces T_2' and U_2' at the interfaces. Then, we have $(\overline{Q},[]) \mapsto^* ([(o,N_2 \bowtie_\phi N)],T_2')$, $(Q,[]) \mapsto^* ([(r,N_2 \bowtie_\phi N)],U_2')$ and $|S,T_2',U_2',V'| \neq \varnothing$. ☐

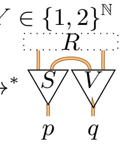
6. Application on interaction combinators

As we stated, our observational equivalence is strongly inspired by Mazza's equivalence [19]. If he defines it for any interaction net library, he only defines a sound and fully abstract semantics $[\![\_]\!]$ for symmetric combinators. The two equivalences coincide on symmetric combinators. In particular, $[\![N_1]\!] = [\![N_2]\!] \Leftrightarrow [N_1] = [N_2]$. Here, we will even see that the structures of those semantics are quite similar.

Mazza defines an arch for the interaction N as a multiset $\{\!\{(p,S_\delta,S_\zeta),(q,V_\delta,V_\zeta)\}\!\}$ where p, q are free ports of N, and $S_\delta, S_\zeta, V_\delta, V_\zeta \in \{1,2\}^{\mathbb{N}}$. We can notice that the shape is similar to our semantics, highlighted by the use of similar names for corresponding objects. One of the differences is that the information in the trace S is divided in a sequence S_δ corresponding to the δ cells and a sequence S_ζ corresponding to the ζ cells. The link is made more precise by the mappings $(\_)_\delta$ and $(\_)_\zeta$ from traces S to finite sequences on $\{1,2\}$, defined by induction on $|S|$: $[]_\delta = []_\zeta = []$, $(T.(\delta,i))_\zeta = T_\zeta$, $(T.(\zeta,i))_\delta = T_\delta$, $(T.(\delta,i))_\delta = T_\delta.i$ and $(T.(\zeta,i))_\zeta = T_\zeta.i$.

Let N be a net, the edifice of N is the set $\mathfrak{E}(N) =$

$$\left\{ \{\!\{(p,S_\delta@X,S_\zeta@Y),(q,V_\delta@X,V_\zeta@Y)\}\!\} \;\middle|\; \begin{array}{l} X,Y \in \{1,2\}^{\mathbb{N}} \\ N \to^* \end{array} \right\}$$



However, it is possible that nets are observationally equivalent but have different edifices. To be fully abstract, we will define a distance on arches and consider the metric completion of edifices. First, let us consider the usual distance on infinite sequences: if $S, V \in \{1,2\}^{\mathbb{N}}$, we define $d(S,V) = 2^{-k}$ where k is the length of the longest common prefix between S and V. On P_f^N, we will use the discrete topology: if $p = q$ then $d_{disc}(p,q) = 0$, else $d_{disc}(p,q) = 1$. We use those distances to define a distance on $P_f^N \times \{1,2\}^{\mathbb{N}} \times \{1,2\}^{\mathbb{N}}$:

$$d\left((p,S,S'),(q,V,V')\right) = \max\left\{d(S,V), d(S',V'), d_{disc}(p,q)\right\}$$

Finally, we can define a distance on arches. If $a = \{\!\{\mu,\mu'\}\!\}$ and $b = \{\!\{\nu,\nu'\}\!\}$, then

$$d(a,b) = \min\{d(\mu,\nu) + d(\mu',\nu'), d(\mu,\nu') + d(\mu',\nu)\}$$

In other words, as the pairs are unordered, we compare them in the two possible ways and we choose the best matching. Finally, we define $[\![N]\!]$ as the metric completion of $\mathfrak{E}(N)$.

Our semantics $[\_]$ is based on the $|\_,\_,\_,\_|$ function, we will study its behaviour on symmetric combinators. We denote the prefix order on sequences by $\leq$ (i.e. $T \leq U \Leftrightarrow \exists V, T@V = U$), and we define $\lessgtr$ as $\leq \cup \geq$. We also define $T - U$ as $(T@T') - T = T'$ and otherwise $T - U = []$. Then, we can observe that for every $T, U \in Tra^+$,

$$|S,T,U,V| \neq \varnothing \Leftrightarrow \begin{cases} S_\delta \lessgtr T_\delta, U_\delta \lessgtr V_\delta, S_\delta - T_\delta \lessgtr V_\delta - U_\delta \\ S_\zeta \lessgtr T_\zeta, U_\zeta \lessgtr V_\zeta, S_\zeta - T_\zeta \lessgtr V_\zeta - U_\zeta \end{cases}$$

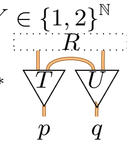
We can verify that $[N]$ is the set of prefixes of merging (meaning that the $\{1,2\}$ sequences for δ and ζ are merged into traces) of elements of $\mathfrak{E}(N)$:

$$[N] = \left\{ \{\!\{(p,S),(q,V)\}\!\} \;\middle|\; \begin{array}{c} \exists S', V' \in Tra^+, S \leq S', V \leq V' \\ \{\!\{(p, S'_\delta, S'_\zeta), (q, V'_\delta, V'_\zeta)\}\!\} \in \mathfrak{E}(N) \end{array} \right\}$$

We wrote that one of the differences between $[N]$ and $[\![N]\!]$ is that, in the first, the information corresponding to δ and ζ are merged whereas they are separated in the second. On this point, Mazza's specialized semantics is better than our general semantics, because $[\![N]\!]$ is closer to full-completeness. Indeed, the structure of $[N]$ allows to have $\{\!\{(p,[(\delta,1);(\zeta,2)]),(q,[])\}\!\} \in [N]$ and $\{\!\{(p,[(\zeta,2);(\delta,1)]),(q,[])\}\!\} \notin [N]$ but the operational semantics of interaction combinators makes this impossible.

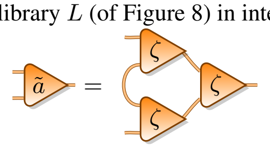
The second difference is that $[N]$ is defined by prefixes of arches, while $[\![N]\!]$ is defined by a metric completion. Thus, we noticed that if $E, F \subseteq \{1,2\}^{\mathbb{N}}$, then the completions of E and F are equal iff $\{S | \exists T \in E, S \leq T\} = \{S | \exists T \in F, S \leq T\}$. So we could interpret nets by the following semantics $\lfloor\_\rfloor$ which is equivalent to $[\![\_]\!]$ but which we consider simpler to understand because it does not use metric completions.

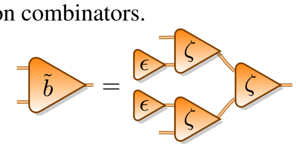
$$\lfloor N \rfloor = \left\{ \{\!\{(p, S_1, S_2), (q, V_1, V_2)\}\!\} \;\middle|\; \begin{array}{c} X, Y \in \{1,2\}^{\mathbb{N}} \\ N \to^* \text{[diagram]} \\ S_1 \leq T_\delta @ X, S_2 \leq T_\zeta @ Y \\ V_1 \leq U_\delta @ X, V_2 \leq U_\zeta @ Y \end{array} \right\}$$



6.1 Comparison with semantics of encodings in symmetric combinators

As one can encode numerous libraries in symmetric combinators, one could try to define the semantics of a net N as $[\tilde{N}]$ with $\tilde{N}$ the encoding of N in interaction combinators. However, this semantics does not match $\approx$. Indeed, let us consider the following encoding of library L (of Figure 8) in interaction combinators.





We wrote that in the library L every pair of nets with the same number of free ports are equivalent. In particular, in L, $a \approx b$. One can observe that $\{\!\{(p,[(\zeta,1)]),(p,[(\zeta,2)])\}\!\} \in [\tilde{a}]$ whereas $\{\!\{(p,[(\zeta,1)]),(p,[(\zeta,2)])\}\!\} \notin [\tilde{b}]$ so $\tilde{a} \not\approx \tilde{b}$ in L_{comb}. The difference is that, in L_{comb}, we can test nets with traces which do not exist in L.

7. Conclusion

We defined a context semantics for any library of interaction nets, and explored some possible applications.

Our weight could for example be used to prove the $Ptime$ soundness of LLL (subsystem of linear logic) and LPL (type system for λ-calculus with pattern matching) in a uniform way. This may ease the transformation of other linear logic subsystems ($QBAL$, L^4) into programming languages.

Our semantics could be used as a first step towards more abstract or fully complete semantics for systems definable in interaction nets.

References

[1] P. Baillot, P. Coppola, and U. Dal Lago. Light logics and optimal reduction: Completeness and complexity. *Information and Computation*, 209(2), 2011.

[2] P. Baillot and M. Pedicini. Elementary complexity and geometry of interaction. *Fundamenta Informaticae*, 45(1-2), 2001.

[3] U. Dal Lago. The geometry of linear higher-order recursion. In *Logic in Computer Science, 2005. LICS 2005. Proceedings. 20th Annual IEEE Symposium on*. IEEE, 2005.

[4] U Dal Lago. Context semantics, linear logic, and computational complexity. *ACM Transactions on Computational Logic*, 10(4), 2009.

[5] V. Danos and L. Regnier. Proof-nets and the Hilbert space. *London Mathematical Society Lecture Note Series*, 1995.

[6] M. De Falco. Géométrie de l'interaction et réseaux différentiels. *These de doctorat, Université Aix-Marseille*, 2, 2009.

[7] M. Fernández and I. Mackie. Operational equivalence for interaction nets. *Theoretical Computer Science*, 297(1), 2003.

[8] M. Fernández, I. Mackie, S. Sato, and M. Walker. Recursive functions with pattern matching in interaction nets. *Electronic Notes in Theoretical Computer Science*, 253(4), 2009.

[9] J.Y. Girard. Proof-nets: the parallel syntax for proof-theory. *Logic and Algebra*, 180, 1996.

[10] G. Gonthier, M. Abadi, and J.J. Lévy. The geometry of optimal lambda reduction. In *Proceedings of the 19th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*. ACM, 1992.

[11] G. Gonthier, M. Abadi, and J.J. Lévy. Linear logic without boxes. In *Logic in Computer Science, 1992. LICS'92., Proceedings of the Seventh Annual IEEE Symposium on*. IEEE, 1992.

[12] Y Lafont. Interaction nets. In *Principles of programming languages, 17th ACM SIGPLAN-SIGACT symposium on*. ACM, 1989.

[13] Yves Lafont. Interaction combinators. *Information and Computation*, 137(1), 1997.

[14] S. Lippi. Encoding left reduction in the lambda-calculus with interaction nets. *Mathematical Structures in Computer Science*, 12(6), 2002.

[15] I. Mackie. The geometry of interaction machine. In Ron K. Cytron and Peter Lee, editors, *POPL*. ACM Press, 1995.

[16] I. Mackie and J.S. Pinto. Encoding linear logic with interaction combinators. *Information and Computation*, 176(2), 2002.

[17] D. Mazza. Multiport interaction nets and concurrency. In *CONCUR 2005–Concurrency Theory*. Springer, 2005.

[18] D. Mazza. Interaction nets: Semantics and concurrent extensions. *These de doctorat, Université Aix-Marseille II/Universita degli Studi Roma Tre*, 2006.

[19] D. Mazza. Observational equivalence and full abstraction in the symmetric interaction combinators. *Logical Methods in Computer Science*, 5(4:6), 2009.

[20] M. Perrinel. Ston paths-based criteria for polynomial time complexity in proof-nets (long version). `http://arxiv.org/abs/1201.2956`, 2013.