On Context Semantics and Interaction Nets
Matthieu Perrinel

To cite this version:
Matthieu Perrinel. On Context Semantics and Interaction Nets. CSL-LICS, Jul 2014, Austria. pp.10, 10.1145/2603088.2603155. hal-00992579

HAL Id: hal-00992579
https://hal.archives-ouvertes.fr/hal-00992579
Submitted on 19 May 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
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 traversing along the program according to some rules. It has 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 data-types (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. Thus, 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 \( W_M = W_N - 1 \). 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.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Copyright © 2014 ACM 978-1-4503-2886-9/14...$15.00.
http://dx.doi.org/10.1145/2603088.2603155
paths, for a special library called interaction combinators [14]. Then, he defines a GOL 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 $S = (S, \alpha)$ with $S$ a countable set whose elements will be called symbols and $\alpha$ a mapping from $S$ 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 $P$ of ports. There are three types of ports: ports attached to a cell (the set $P_C$), free ports (the set $P_f$) and merging ports (the set $P_m$).

**Definition 1.** A net $N$ is a tuple $(P, C, \sigma)$ with $P = P_C \cup P_f \cup P_m$ is a finite set called set of ports.

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

**Example 1.** Let $S_{comb} = \{\alpha, \delta, \epsilon\}$ be symbols with $\alpha(\alpha) = \alpha(\delta) = 2$ and $\alpha(\epsilon) = 0$. Then, Figure 1 represents the net $\mathcal{R}$ with:

\[ P = \{a_0, a_1, b_0, b_1, c_0, c_1, d_1, f_1, f_2, m_1, m_2\}, \]
\[ C = \{A, B, C, E\}, \]
\[ \sigma^N = \{a_0 \mapsto a_1, a_2 \mapsto b_0, b_1 \mapsto c_0, c_1 \mapsto c_2, c_0 \mapsto f_1, f_2 \mapsto f_2, m_0 \mapsto m_1, m_2 \mapsto m_1, m_2 \mapsto m_1\}, \]
\[ \sigma^N = \{a_0 \mapsto (A, 0), a_1 \mapsto (A, 1), a_2 \mapsto (A, 2), b_0 \mapsto (B, 0), b_1 \mapsto (B, 1)\}. \]

Figure 1: Net $\mathcal{R}$. Names of ports and labels of cells are represented while names of cells are not.
combinators. The net of Figure 1 successively reduces to the nets of Figure 5:

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 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 $\text{Pot}^N$ of potential ports of net $N$ is the set of lists $\{((p_0, N_1); (p_1, N_{s_1, t_1}); \cdots; (p_k, N_{s_k, t_k})) \mid$ such that for each $j$: $p_j$ is a port of $N_{s_j, t_j}$, and $p_{j-1}$ is the principal port of a cell labelled by $t_j$. For instance, in Figure 6, the potential ports of $N$ $\{([b_0, N]), ([b_0, N_2_2]); [b_2, N_{3, 2}]; ([b_0, N]); (d_2, N_{3, 2}); (e_1, N_{s_i, t})\}$ point to the ports of the potential net of $N$ they represent. For $P(p, N') \in \text{Pot}^N$, we set $\text{Pot}(p, N') = P(p, N')$. Note that $\text{Pot}(p, N')$ 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 $\Rightarrow$ on contexts. The trace represents information about the beginning of the $\Rightarrow$ path, we need this information for the $\Rightarrow$ paths to be stable by reduction. In Figure 6, we represent by thick arrows the path $([b_2, N]), ([a_0, N_2_2]); ([\zeta_2, 2]); ([a_0, N]); ([b_2, N_{3, 2}); ([e_1, N_{s_i, t})]).$ On the potential net of $N$ and its reduction $([b_2, N_1]), ([a_0, N_2_2]); ([\zeta_2, 2]); ([a_0, N]); ([b_2, N_{3, 2}); ([e_1, N_{s_i, t})])$.

A positive trace element is $(s, i)$ with $s \in S$ 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 $\text{Tr}^+$.

A negative trace element is $(s, i)$ with $s \in S$ and $1 \leq i \leq \alpha(s)$. The meaning of $(s, i)$ 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 a negative trace element.

Example 2. As another example, let us consider an ordered set $(A, \leq)$, and the symbols $\{S, [], \} \cup \{a \mid a \in A \} \cup \{\pi \mid a \in A \}$. The arities and the library $L_{\text{sort}}$ 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 $[a_1; \cdots; a_n]$, $l_1 \cdot l_2$ represents the concatenation of $l_1$ and $l_2$, $l_1$ represents “push” $(a_1; \cdots; a_n), b = [a_1; \cdots; a_n], b$ and $|l|$ is the length of $l$.
a negative trace element. A trace is a list of trace elements. The set of traces is written $\text{Tra}$.

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

**Definition 4.** For any net $N$, we define a relation $\rightarrow$ on $\text{Cont}^N$ by the rules of Figure 7. In those rules, we suppose $s, s' \in S$, $c, c' \in C^N$, $t^N(c) = s$, $t^N(c') = s'$, $1 \leq k \leq \alpha(s)$, $1 \leq k' \leq \alpha(s')$ and $m, m' \in P^N_{\text{cont}}$ with $\sigma^N_{\text{cont}}(m) = m'$.

The intuition underlying the definition of $\rightarrow$ is that, if $(P, \emptyset) \rightarrow^* (Q, \{(s_1, i_1); \cdots ; (s_n, i_n)\})$ 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 indice of the trace. We will use this construction again, representing the net corresponding to $T$ by

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

If $p$ is an auxiliary port, the cell does not exist on the trace (rule a).

If $p$ is a principal 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, $C \not\rightarrow^* k$):

- 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 $\rightarrow$, $N$ 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 if $I_k$ will be merged with $p_k(c)$ during the reduction.

- Else if $t$ is negative (rule b), then $t = (s', k')$. According to the intuitive meaning we gave to the 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 $T_k$ (rule e). In the first case, we know that during reduction $O_{k'}$ will be merged with $p_{k'}(c')$ 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 $p_k(c)$. 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 $p_{k'}(c)$? The idea is to use the intuition underlying the $\rightarrow$ relation: we know that $c$ and $c'$ will form an active pair, so $(p(c'), (s, k')) \rightarrow^* (p_{k'}(c), (s, k'))$. According to rule b, $(p(c'), (s, k)) \rightarrow^* (p_{k'}(c), (s, k)) \rightarrow^* (p_k(c), (s, k))$. Thus, to find $p_{k'}(c)$, we move the context to $p_{k'}(c')$ with $(s, k)$ on the trace.

Let us recall that we consider the interaction nets up to $\alpha$-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 $\rightarrow$ relation, so $\rightarrow$ is the same on $\alpha$-equivalent nets (up to the renaming of the ports in the potential ports of the contexts). We can verify that whenever $\left( (p_1, M_1); \cdots ; (p_k, M_k) \right), T) \rightarrow^* \left( (q_1, N_1); \cdots ; (q_l, N_l) \right), U)$, with:

- For all $1 \leq i \leq k$, $p_i$ is a merging port and there exists a net $M_i'$ such that $M_i \equiv_m M_i'$.

Then, the following path is valid:

\[
\left( (p_1, M_1'); \cdots ; (p_k, M_k') \right), T) \rightarrow^* \left( (q_1, N_1'); \cdots ; (q_l, N_l') \right), U)
\]

**Example 3.** The following $\rightarrow$-path in the net $N$ of Figure 1, goes from the principal port of $E$ to the $\delta$ cell which will form an active pair with $E$. Notice that this $\delta$ cell does not exist.
yet (it will be created by the $\zeta/\delta$ reduction): $(((b_2, N), []]) \mapsto (((a_0, N), [(\zeta, 2)])) \mapsto (((a_0, N); (I_2, N_c,s), []))$.  

**Example 4.** As a more involved example, we will study in the net $N$, the path between the two $o$ cells created during the $\delta e$ step of reduction (first step of Figure 4).

$\begin{align*}
((e_0, N); (r_1, N_c,s), []) &\mapsto (((a_0, N); ([\tilde{b}2], (\zeta, 2))); (\delta, 1)) \\
&\mapsto (((a_0, N); ([\tilde{b}2], (\zeta, 2))); (\delta_1, 1)) \\
&\mapsto (((a_0, N); ([\tilde{b}2], N_c,s)); [], (\delta, 2)) \\
&\mapsto (((a_0, N); ([\tilde{b}2], N_c,s)); [], (\delta, 2)) \\
&\mapsto (((a_0, N); ([\tilde{b}2], N_c,s)); [], (\delta, 2))
\end{align*}

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

In this section, we suppose that $N \rightarrow N'$ 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 $(R_2, \phi_2) = L(s_1, s_2)$. Then $N = N_0R_1s_2R_2s_1N'$ and $N' = N_0R_2s_1R_1s_2N$. We define a mapping $P$ from $Pots^N$ to $Pots^N$ which depends on the leftmost port $p$:

- If $p \in Pots^N$, we set $\Pi(\{p, N\} = \{p, N'\}) \mapsto P$.
- If $p = p_0(c_i)$ for $i \in \{1, 2\}$, we set $\Pi(\{p_0(c_i), N\} = \{r, R_i\}) \mapsto P$.

Otherwise, $P$ is undefined.

The next two 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 $P' \in Pots^N$, then there exists $P \in Pots^N$ such that $\Pi(P) = P'$.

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

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

We set $(p, N) \in \mathcal{P}_3 = (q, N) \in \mathcal{Q}_1 = Q$.

- If $p, q \in Pots^N$, then $(P, T) \rightarrow^+(Q, U)$ (the path has length 1). $P' = (p, N') \mid \mathcal{P}_3$ and $Q' = (q, N') \mid \mathcal{Q}_1$. Given that all $\mapsto$ rules are local, and that ports $N_0$ are unaffected by the reduction, $(P', T) \rightarrow^+(Q', U)$.

- If $p$ and $q$ belong respectively to $Pots^R$ and $Pots^R$ with $i \in \{1, 2\}$, then a careful observation of the $\mapsto$ rules shows that the only possibility is $i = j$ and $(P, T) \rightarrow^+(Q, U)$. Because $\Pi(P)$ and $\Pi(Q)$ are defined, we have $(p_0(c_1), N); (r, R_i) \in \mathcal{P}_3$, and $Q = (p_0(c_1), N); (s, R_i) \in \mathcal{Q}_1$. For some $r, s \in Pots^R$. So $P' = (\{r, N'\}) \mid \mathcal{P}_3$ and $Q' = (\{s, N'\}) \mid \mathcal{Q}_1$. Considering that $i \mapsto j$ is local, $(P', T) \rightarrow^+(Q', U)$.

- If $p \in Pots^N$ and $q \in Pots^N$, then $j = 3-i$ (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 of $\mathcal{P}_3$. $r \in N'$ is merged with the free port $\psi_0(i)$ of $R_i$. So, we have $(P, T) = (p_0(c_1), N) \in \Pi(P, T) \rightarrow (p_0(c_1), N) \mid \mathcal{P}_3$. We can notice that $P' = (p, N') \in \Pi(P, T) \rightarrow (p, N') \mid \mathcal{Q}_1$. We get $(P', T) \rightarrow^+(\{\sigma_{N'0}(p, N')\}, T) = (Q', U)$.

- If $P \in Pots^R$ (with $i \in \{1, 2\}$, we will write $j = 3-i$ and $q \in Pots^N$, then $\eta$ is a free port of $N_0$ which, in $N$, is merged with a free port of $\mathcal{P}_3$, and, in $N'$, is merged with a free port $\psi_0(i)$ of $R_i$. And, either $p = \mathcal{P}_3 = \psi_0(ik)$ (and $1 \leq k \leq \alpha(s_j)$) or $p = \mathcal{P}_3$ (and $1 \leq k < \alpha(s_j)$).

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.
Let $\Pi_1, \Pi_2, \Pi_3$ and $\Pi_4$ be the projections corresponding to the reduction steps of Figures 1 and 4. If $\Pi_1([(e_0,N);(r_1,R_{\delta,s})]) = [(e_0,N_1);(r_1,R_{\delta,s})]$, then $\Pi_2([(e_0,N_1);(r_1,R_{\delta,s})]) = [(b_2,N_2)]$, and $\Pi_4([(b_2,N_2)])$ is not defined.

The path $[(e_0,N);(r_1,R_{\delta,s})],[\mu] \rightarrow^1 [(e_0,N);(r_2,N_2)],[\mu] \rightarrow^2 [(d_2,N_2)_1],[(\delta,1)],\rightarrow^4 [(\overrightarrow{2},N_2)_1],[\mu] \rightarrow^5 [(e_0,N),(e_2,N)],[\mu]$ in $N_1$, then $[(b_1,N_2)_1],[\mu] \rightarrow^3 [(i_{b_1},N_2)_1],[\mu]$ in $N_2$ and $[(i_{b_2},N_3)_1],[\mu] \rightarrow^0 [(i_{b_2},N_3)_1],[\mu]$ 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 $M \rightarrow N$, 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, $(P_2.(p_0(c_2),N_2),[\mu]) \rightarrow (P_3.(p_0(c_3),N_3),[\mu]),I^{P_1}(c_1) \equiv s_1, I^{N_1}(c_2) \equiv s_2$ and $L(s_1, s_2)$ is defined. Then for every cell $c$ of $N$, $[(b_{s_1},s_1)]$. $P_3.(p_0(c_1),N_1).(p_0(c_3),N_{s_2},s_2) \in Can^N$.

Lemma 2. Let us suppose that $N \rightarrow L$ by reducing the active pair $(c_1,c_2)$ and $\Pi$ 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 $C_1 = [(e_0,N);(e_1,R_{\delta,s})]$ is a canonical cell. Indeed, $b_{\delta}$ is a principal port of $N$ so $[(b_0,N)]$, is a canonical cell. We know that $[(\overrightarrow{b_0},N)],[\mu] \rightarrow^0 [(\overrightarrow{a_0},N)],[\mu]$ and $L(\xi,\delta)$ is defined so $[(b_0,N);(d_2,N_{\delta,s})]$ is a canonical cell. Finally, $[(\overrightarrow{b_0},N);(d_2,N_{\delta,s}),[\mu] \rightarrow^1 [(\overrightarrow{c_0},N)],[\mu]$ and $L(\xi,\delta)$ is defined so $[(b_0,N);(d_2,N_{\delta,s});(e_1,R_{\delta,s})]$ is canonical.

Similarly, $C_2 = [(e_0,N);(d_2,N_{\delta,s});(e_1,N_{\delta,s})]$ and $C_3 = [(e_0,N);(e_1,N_{\delta,s})]$ are canonical cells. Let $\Pi_1, \Pi_2, \Pi_3$ be the projections corresponding to $N \rightarrow N_1$ and $N_1 \rightarrow N_2$ (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:

$$T_N = \sum_{P \in Can^N} \frac{1}{2^{P_1'}}$$

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

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

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

$$\sum_{P \in Can^{N'}} \frac{1}{2^{P_1'}} = \frac{1}{2^{P_1'}}$$

This gives the following equations:

$$T_N = \sum_{P \in Can^{N'} \cap D} \frac{1}{2^{P_1'}} + \sum_{P \in Can^{N'} \cap D} \frac{1}{2^{P_1'}}$$

$$T_N = \sum_{P \in Can^{N'}} \frac{1}{2^{P_1'}} + \frac{1}{2^{\|p_0(c),N\|}} + \frac{1}{2^{\|p_0(c),N\|}}$$

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

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 $\approx$ defined by Mazza [19]. We modified a bit the equivalence, because in some farfetched libraries, $b \cong a \lhd c \otimes d$. 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 cannot be equivalent to a net where $a$ can only interact with $d$. In every system studied by Mazza in [19], the property $(N_1 \approx N_2) \Rightarrow (N_1 \simeq N_2)$ 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 $p_i = q_j(c)$ (with $j > 0$), then $p_{i+1} = p_0(c)$ (crossing a cell from an auxiliary port to the principal port).
- If $p_i = q_j(c)$, then either there exists $j > 0$ such that $p_{i+1} = q_j(c)$ (crossing a cell from the principal port to an auxiliary port) or $p_{i+1} = p_0(c)$ (bouncing on a principal port).
- If $p_i \in P_N$, $p_{i+1} = \sigma_{N_i}(p_i)$ (crossing a merging port).

The observable and $\rightarrow$-path are closely linked. If $(P_1, T_1) \rightarrow \cdots \rightarrow (P_n, T_n)$, and $[(p_1, N_1), \cdots, [p_k, N_k]]$ is the subsequence of $P_1, \cdots, P_n$, $P_k$ 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 cannot be eliminated by reduction.

Let $p, q$ be free ports of $N$. If $N \not\rightarrow^* N'$ and there exists an observable path from $\sigma_N(p)$ to $q$, then we write $N \parallel p_q$

**Definition 6** (observational equivalence). Let $N_1, N_2$ be nets with $P_{N_1}^f = P_{N_2}^f$, we write $N_1 \approx N_2$ if for all nets $N$, $\phi$ partial injection from $P_{N_1}^f$ to $P_{N_2}^f$, $\exists p, q \in P_{N_1}^{\{\phi\}}: (N_1 \parallel q) \not\Rightarrow (N_2 \parallel p)$

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

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

\[
\{[p, S], (q, V) \mid \exists P \in P_{\text{net}}^f, (P, \parallel) \Rightarrow ([p, N], T), S, V \in \text{Tra}^+, [S, T, U, V] \neq \emptyset \}\]

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

### 5.3 Stability of [ ] by reduction and gluing

**Theorem 2.** If $N \not\rightarrow N'$, then $[N] \simeq [N']$

**Proof.** Follows from Lemma 1 and the definition of $[N]$. □

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.** $\bigcup_{(Z,W)\in [\cdot,\cdot,\cdot,\cdot]} [X, T, U, Z] \simeq [X, T, U @ V, Y]$

**Lemma 4.** $\bigcup_{(X,Y)\in [\cdot,\cdot,\cdot,\cdot]} [S, X @ T, Y, V] \simeq [S, T, U, Z @ V]$
Lemma 5. Let $P,Q \in \text{Pot}^N$ and $T,U \in \text{Tra}$,

$$
(P,[]) \mapsto (R,T), (Q,[]) \mapsto (R,U), ([],[],T,U) = (S,V)
$$

$$
\Rightarrow \exists R' \in \text{Pot}^N, (R',[]) \mapsto (P,S), ([],[[],T,U]) = (S,V)
$$

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

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 $p \xrightarrow{r} q$ and then we can apply the above reasoning to get paths in $N_1$ of the shape $(R,[])$ to $(\{(p,N_1), S\})$ and $(R,[]) \mapsto (\{(q,N_1), V\})$. 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 $\phi$ an injection from $P_{f}^{M_1} \cup P_{f}^{N_1}$ to $P_{f}^{M_2} \cup P_{f}^{N_2}$, then $[M_1] \supseteq [N_1] = [M_2] \supseteq [N_2]$.

Proof. For concision, we will write $G_1 = M_1 \supseteq N_1$ and $G_2 = M_2 \supseteq N_2$. We will not consider the $\supseteq$-norm 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 $\phi$) untouched. We need a notion of $\rightarrow$ to 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 $\rightarrow_{i}$ on $\text{Cont}^{G_2}$ by: $(P,T) \rightarrow_{i} (Q,U)$ if we are in one of those cases:

$$
\begin{aligned}
&i = 0 \\
&i = 0 \\
&(P,T) \rightarrow (Q,U) \\
&(P,T) \rightarrow (Q,U) \text{ with } R \notin \{([P] \cup [P]),[G_1]\}
\end{aligned}
$$

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

"Let $p,q \in P_{f}^{M_1} \cup P_{f}^{N_1}$ and $P_1 \in \text{Pot}^{G_1}$ such that $(P_1,[]) \rightarrow_{i} (([p,G_1],T_1)), ([P_1],[]) \rightarrow_{j} (([q,G_1],U_1))$ and $[S,T_1,U_1,V_1]$ is defined, then there exists $P_2 \in \text{Pot}^{G_2}$ such that $(P_2,[]) \rightarrow_{i+j} (([r,G_2],T_2),([T_2],[])) \rightarrow_{i} (([q,G_2],U_2))$ and $[S,T_2,U_2,V_2]$ 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 $P(i+j-1)$ is true. Let $p,q,r \in P_{f}^{M_1} \cup P_{f}^{N_1}$ and $P_1 \in \text{Pot}^{G_1}$ such that $(P_1,[]) \rightarrow_{i-1} (([r,G_1],U_1),([q,G_1],T_1)), ([P_1],[]) \rightarrow_{j} (([q,G_1],T_1))$ and $[X,T_1,T_2,Y]$ is defined.
5.4 Soundness and full abstraction

Lemma 6. If \( P, Q \in \mathit{Can}^N \) and \((P, T) \rightarrow^\ast (Q, U)\), then we can reduce \( N \) to a net \( N' \) such that \( \Pi \) is the associated composition of projections, \( \Pi(P) \) and \( \Pi(Q) \) have shape \( [(p, N')] \) and \( [(q, N')] \), and the path \( ([(p, N')], T) \rightarrow^\ast ([(q, N')], U) \) 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 \( ([(p, N')], T) \rightarrow^\ast ([(q, N')], U) \) is strictly shorter than the path \( (P, T) \rightarrow^\ast (Q, U) \). So we get the result after finitely many such reductions.

Else, \( P = P_1 \cdot (p_0(c), N_1), (r, R_{a}), (P_1 \cdot p_0(c), N_1), [[]] \rightarrow^\ast ([(p_2 \cdot (p_0(d), N_2)], []) \) 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 \( \Pi(P) \prec |P| \) and \( \Pi(Q) \leq |Q| \), so we can apply the induction hypothesis.

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

Lemma 7. If \( [N_1] = [N_2] \) and \( p, q \in P_{\mathit{f}}^{N_1} = P_{\mathit{f}}^{N_2} \), then:
\[
N_1 \bowtie_p^q N_2 \bowtie_p^q
\]

Proof. We consider the \( \rightarrow_m \) -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 \( N_1 \bowtie_p^q N_2 \) then there exists some net \( N' \) such that \( N_1 \rightarrow^\ast N' \) and \( N_2 \rightarrow^\ast N' \) and there exists an observable path in \( N_1 \) from \( p \) 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 \( p \) to \( q \) with only auxiliary ports (except \( p \) which is free). Thus there exists \( T_1, U_1 \in T \) such that \( (([(p, N_1)], [] \rightarrow^\ast (([(p, N_1)], [r, (\tau, N_2)]), []) \rightarrow^\ast ([(p, N_1), U_1], U_2) \). 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 \( ([(\tau, N'_2)], []) \rightarrow^\ast ([(p, N'_2), U_2]), ([(s, N'_2)], []) \rightarrow^\ast ([(q, N'_2), U_2]) \) do not cross active pairs.

Thus, in \( N'_2 \), there are observable paths from \( p \) to \( q \) and from \( s \) to \( q \) with only auxiliary ports. This means that there is an observable paths, in \( N'_2 \), from \( p \) to \( q \). So \( N_1 \bowtie_p^q N_2 \).

Theorem 4 (soundness). If \( P_{\mathit{f}}^{N_1} = P_{\mathit{f}}^{N_2} \) and \( [N_1] = [N_2] \), then \( N_1 \approx N_2 \).

Proof. Let us consider a net \( N \) and \( \phi \) a partial injection from \( P_{\mathit{f}}^{N_1} \) to \( P_{\mathit{f}}^{N_2} \) such that \( p, q \in P_{\mathit{f}}^{N_1} \# N \), we need to prove that \( (N_1 \# N) \bowtie_p^q (N_2 \# N) \bowtie_p^q \).

By Theorem 3, we know that \( [N_1 \# N] = [N_2 \# N] \). So, the result is given by Lemma 7.

Theorem 5 (full abstraction). If \( P_{\mathit{f}}^{N_1} = P_{\mathit{f}}^{N_2} \) 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 \( P \in \mathit{Can}^N \) and \( T, U \in T \) such that \( (P, []) \rightarrow^\ast ([(p, N_1)], T), [(P, []) \rightarrow^\ast ([(q, N_1)], U) \) and \( |S, T, U, V| \neq \emptyset \). We use Lemma 6 to reduce \( N_1 \) to a net \( N_1' \) such that the projection \( \Pi(P_1) = 1 \) and the paths \( (\Pi(P), []) \rightarrow^\ast ([(p, N_1), T), (\Pi(P), []) \rightarrow^\ast ([(q, N_1), U] \) do not cross active pairs. So \( p \rightarrow^\ast \phi \rightarrow^\ast \) and \( q \) is a subterm of \( N_1' \). We set \( N = \alpha\rightarrow^\ast \phi \rightarrow^\ast \) and \( \phi = (p \rightarrow^\ast q \rightarrow^\ast \) then \( N_1 \bowtie_p^q N \) reduces to a net which has \( \alpha \rightarrow^\ast \phi \rightarrow^\ast \) as a sub-net. We know that \( |S, T, U, V| \neq \emptyset \) so \( (N_1 \bowtie_p^q N) \bowtie_p^q \). Thus, \( N_1 \bowtie_p^q N \) 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 \( ((s, N_2'), []) \rightarrow^\ast (\alpha, N_2)), (\tau, N_2]), []) \rightarrow^\ast (\alpha, N_2)), (\tau, U_2)) \). By Lemma 1, we know that \( s \) is the projection of \( Q \in \mathit{Can}^N \# N \) and that the paths exist in \( N_2 \bowtie_p^q N \). Those paths begin in \( N_2 \) and end \( N_2 \), we can use the traces \( T_2, U_2, V_2 \neq \emptyset \).

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 multisets \( (p, S, S_1, q, V, V_1) \) where \( p, q \) are free parts of \( N \), \( S_1, S_2, V_1, V_2 \in \{1, 2\}^N \). 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 \( S_1 \) corresponding to the \( \delta \) cells and \( S_2 \) corresponding to the \( \zeta \) cells. The link is made more precise by the mappings \( (\cdot, S_1) \) and \( (\cdot, S_2) \) from traces \( S \) to finite sequences of \( \{1, 2\} \), defined by induction on \( |S| \) : \( |S| = 0 \Rightarrow \emptyset \), \( |S| = 1 \Rightarrow \{1\} \), \( |S| = 2 \Rightarrow \{1, 2\} \), \( |S| = 3 \Rightarrow \{1, 2, 3\} \), \( |S| = 4 \Rightarrow \{1, 2, 3, 4\} \)

Let \( N \) be a net, the edifice of \( N \) is the set \( \mathcal{E}(N) = \)
In other words, as the pairs are unordered, we compare them in the two possible ways and choose the best matching. Finally, we define $\llbracket N \rrbracket$ as the metric completion of $\mathcal{E}(N)$.

Our semantics $\llbracket \cdot \rrbracket$ is based on the $\lfloor \cdot \rfloor$ function, which we will study its behaviour on symmetric combinators. We denote the prefix order on sequences by $\preceq$ (i.e. $T \preceq U \iff \exists V : T \triangleright V = U$), and we define $\llbracket \cdot \rrbracket$ as $\llbracket \cdot \rrbracket$.

We can verify that $\llbracket N \rrbracket$ is the set of prefixes of merging (meaning that the $\{1,2\}$ sequences for $\delta$ and $\zeta$ are merged into traces) of elements of $\mathcal{E}(N)$:

$$\llbracket N \rrbracket = \{ ((p, S), (q, V)) \in \mathcal{E}(N) \mid \exists S' : V' \in \text{Tra}^+, S \preceq S', V \preceq V' \}$$

We wrote that in the library $L$ every pair of nets with the same number of free ports are equivalent. In particular, in $L_{\text{comb}}$, 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 $\text{LLL}$ (subsystem of linear logic) and $\text{LPL}$ (type system for $\lambda$-calculus with pattern matching) in a uniform way. This may ease the transformation of other linear logic subsystems ($\text{QBA}, L^3$) 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


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 $\llbracket N \rrbracket$ 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 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 $\text{LLL}$ (subsystem of linear logic) and $\text{LPL}$ (type system for $\lambda$-calculus with pattern matching) in a uniform way. This may ease the transformation of other linear logic subsystems ($\text{QBA}, L^3$) 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