Math. Struct. in Comp. Science (2007), vol. 17, pp. 527–562.

A Denotational Semantics for the Symmetric Interaction Combinators

Damiano Mazza

Institut de Mathématiques de Luminy (UMR 6206)
Campus de Luminy, Case 907 — 13288 Marseille Cedex 9
mazza@iml.univ-mrs.fr
http://iml.univ-mrs.fr/∼mazza

Received 25 April 2006; Revised 19 August 2006

The symmetric interaction combinators are a variant of Lafont's interaction combinators. They enjoy a weaker universality property with respect to interaction nets, but are equally expressive. They are a model of deterministic distributed computation, sharing the good properties of Turing machines (elementary reductions) and of the λ-calculus (higher-order functions, parallel execution). We introduce a denotational semantics for this system, inspired by the relational semantics for linear logic, proving an injectivity and full completeness result for it. We also consider the algebraic semantics defined by Lafont, and prove that the two are strongly related.

1. Introduction

Interaction nets (Lafont, 1990) are a model of distributed deterministic computation. Based on graph-rewriting, they can be seen as a generalization of multiplicative linear logic proof-nets (Girard, 1987b; Danos and Regnier, 1989; Lafont, 1995).

Interaction nets are interesting for several reasons:

- They are highly expressive: Turing machines, cellular automata, and a number of word or term rewriting systems can all be seen as special instances of interaction nets. By this we mean more than just the existence of some (maybe impractical) encoding: in many cases, interaction nets are able to "implement" a computational model preserving its fundamental properties (like sequentiality, parallelism, complexity, etc.). Two interesting examples of such translations (Turing machines and cellular automata) are given by Lafont (Lafont, 1997).
- They have been shown to be the "true" syntax underlying Girard's Geometry of Interaction (Girard, 1989; Gonthier et al., 1992). This is potentially of great theoretical interest, since the geometry of interaction is a semantics which attempts to give a mathematical meaning to the *execution* of a program, rather than just to its final result; therefore, it may offer a powerful instrument to study the dynamics of computation, like its complexity, etc.
- From a more applicative point of view, they can be seen as a programming paradigm,

and can be turned into a practical (typed or untyped) programming language, in which important properties (like deadlock-freeness) are automatically checked through similar techniques to those used in linear logic (i.e., correctness criteria (Lafont, 1990)).

Among all interaction net systems, the *interaction combinators* (Lafont, 1997) stand out as particularly interesting, because they are *universal*, in the sense that all other interaction net systems can be "compiled" in them; again, this compilation process preserves the basic properties of parallelism and complexity of the original system.

As a consequence, the interaction combinators can be seen as a computational model of its own, combining in some sense the good properties of Turing machines (local execution, transitions of elementary complexity, strong determinism) with those of the λ-calculus (higher-order functional programming, possibility of "parallelizing" the execution).

From this it would appear that studying a denotational semantics for the interaction combinators may be, at least in principle, as interesting as studying the denotational semantics of the λ-calculus. And yet to this day there have been very few efforts in this direction; the only work dealing directly with the semantics of the interaction combinators is Lafont's original paper, in which a path semantics for nets of combinators is introduced, and an interpretation in terms of stack automata is given. Another contribution of semantical flavor is that of Maribel Fernández and Ian Mackie (Fernández and Mackie, 2003), in which the fundamental operational equivalences for the interaction combinators are obtained as an application of more general results.

Our work aims precisely at deepening the semantical study of the interaction combinators. In a previous paper (Mazza, 2006), we have analyzed the notion of observational equivalence for nets of combinators. A congruence analogous to $\beta\eta$-equivalence is defined, and an internal separation result similar to Böhm's Theorem is proved for it. This paper takes this result as a basis for developing a denotational semantics, i.e., we find a mathematical structure in which $\beta\eta$-equivalence becomes an equality.

More precisely, our semantics is inspired by the relational semantics for linear logic: nets are interpreted as subsets of a certain domain $\mathcal{D}$, called *interaction sets*, which do not need to have any particular structure apart from the existence of two bijections between $\mathcal{D} \times \mathcal{D}$ and $\mathcal{D}$ itself, verifying a certain condition. As expected, this semantics is proved to be injective with respect to $\beta\eta$-equivalence, i.e., two nets are $\beta\eta$-equivalent if and only if they have the same semantical interpretation. Moreover, we prove a full-completeness result with respect to a certain class of subsets, called *balanced*, which is reminiscent of a similar result proved by Michele Pagani for multiplicative proof-nets (Pagani, 2006).

We also consider interaction sets with a minimum of algebraic structure, namely that of a monoid. These structures, called *interaction monoids*, have the property of naturally inducing an algebraic semantics for the combinators, which is a model of the geometry of interaction described by Lafont. In this semantics, a net μ is interpreted as a pair of monoid endomorphisms (u, σ), where $\sigma = \mathbf{0}$ (the everywhere-zero endomorphism) means that μ is in normal form. In case $\sigma \neq \mathbf{0}$, and if μ does have a normal form, the endomorphism interpreting it can be computed by means of Girard's *execution formula* $\mathrm{Ex}(u, \sigma)$.

The denotational and algebraic semantics are tightly connected to each other: we prove in fact that, if μ is a net admitting a normal form, given (u, σ), the denotational semantics of μ is equal to the submonoid of the fixpoints of $\mathrm{Ex}(u, \sigma)$; conversely, the denotational semantics of μ defines the endomorphism interpreting its normal form.

There is an important technical point which must be clarified though: the semantics we discuss here does not deal with the interaction combinators, but with a slightly different variant, which we call the *symmetric combinators*. This interaction net system, also introduced by Lafont (Lafont, 1997), is not universal in the same sense as the interaction combinators, but is just as expressive. In particular, every application of the interaction combinators found so far (for instance Mackie and Pinto's encoding of linear logic and the λ-calculus (Mackie and Pinto, 2002)) can be reformulated with virtually no change using the symmetric combinators. Of course, our work mentioned above on observational equivalence also applies *mutatis mutandi* to this system.

By the way, the symmetric combinators are tightly connected to the *directed combinators* (Lafont, 1997), which are an extension of multiplicative linear logic proof-structures, and may therefore have interesting logical properties.

Contents of the paper. Section 2 contains the introductory material necessary to develop the rest of the paper. The exposition is as self-contained as possible, so even a reader completely unfamiliar with interaction nets should be able to follow the technical contents. In particular, in Sect. 2.4 we give an explicit proof of the expressiveness of the symmetric combinators, by encoding the **SK** combinators in them, and in Sect. 2.5 we briefly recall the main results of the above mentioned paper (Mazza, 2006), which we use later in one of our proofs.

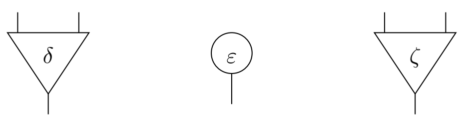
Section 3 is the heart of the paper, and contains the definition of our denotational semantics, together with the injectivity and full completeness proofs. In Sect. 4 we introduce interaction monoids, and develop the algebraic semantics described in Lafont's original paper, proving the relationship between this and our denotational semantics.

Section 5 concludes the paper with a discussion on the technical reasons behind our choice of the symmetric combinators instead of the "standard" interaction combinators, and gives some hints on future work.

2. The symmetric interaction combinators

2.1. *Cells, wires, nets*

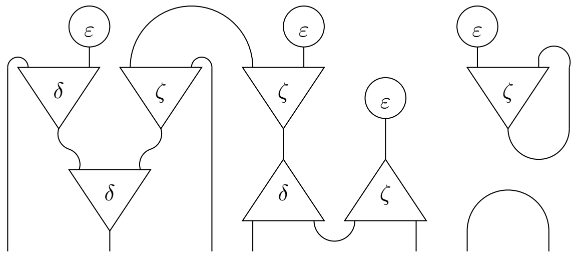
The *symmetric interaction combinators*, or, more simply, the *symmetric combinators*, are the three following *cells*:



Each cell has a number of *ports*; δ and ζ have three, ε has only one. The fundamental property of cells is that *exactly one* of their ports is *principal* (drawn at the bottom in the above graphical representation), the others being *auxiliary*.

The auxiliary ports of the two binary combinators are ordered; to distinguish them, we call one the *left* port and the other the *right* port. Of course it is arbitrary which one is “left” and which one is “right”, as long as the convention is set once and for all. In this paper, we use this terminology in reference to the picture above: when cells are drawn like this, left ports are actually on the left, and right ports actually on the right. Notice however that if a cell is drawn “upside-down” with respect to the above representation, its left port will be on the right of the picture, and vice versa.

Ports may be used to plug cells† together by means of *wires* to form *nets*, as in the following example:



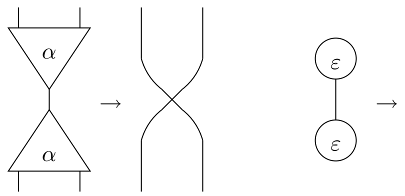
Wires can have one or both of their extremities not connected to any cell, in which case the net has a *free port*, principal or auxiliary (or neither) depending on the nature of the port of the cell connected to the other extremity of the wire. The net above has for example 7 free ports, of which 1 is principal and 4 are auxiliary. The free ports of a net are referred to as its *interface*. The set of all ports of all cells contained in a net μ, with the addition of its free ports, is denoted $\mathsf{Ports}(\mu)$.

2.2. *Interaction rules*

The distinction between principal and auxiliary ports comes into play when defining the dynamics of nets. As a matter of fact, when two cells are connected through their principal ports, they form an *active pair*, and they may be replaced by another subnet according to the appropriate *interaction rules*.

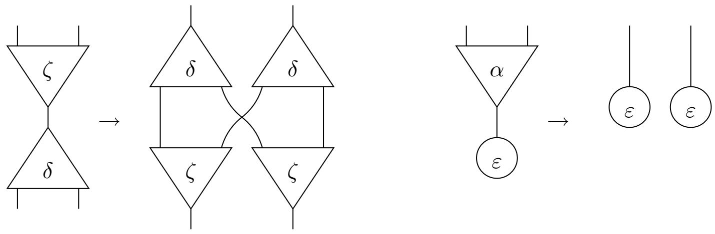
There are six interaction rules, one for each possible active pair. Interaction rules are divided into two groups: the *annihilations*, describing what happens when two cells of the same kind form an active pair, and the *commutations*, describing what happens if the active pair is composed of two cells of different kinds.

If we put $\alpha \in \{\delta, \zeta\}$, the six rules can be condensed into four basic schemes: the annihilations



† Here, and all throughout the rest of the paper, we shall make systematic confusion between cells and *occurrences* of cells.

and the commutations



When a net μ' is obtained from μ after the application of one of the above rules, we say that μ *reduces in one step* to μ', and we write $\mu \to \mu'$. We can then define the *reduction relation* $\to^*$ as the reflexive-transitive closure of $\to$. We write $\mu \simeq_\beta \mu'$ iff there exists μ'' such that $\mu \to^* \mu''$ and $\mu' \to^* \mu''$.

Notice that interaction rules are purely local; if we add to this the fact that cells have exactly one principal port, we immediately obtain

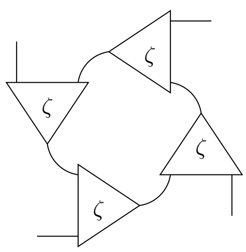
Proposition 2.1 (Strong confluence). The relation $\to$ is confluent, i.e., if μ, μ_1, μ_2 are three nets such that $\mu \to \mu_1$ and $\mu \to \mu_2$, then there exists μ' such that $\mu_1 \to \mu'$ and $\mu_2 \to \mu'$.

Proposition 2.1 means that the reduction process, i.e., the relation $\to^*$, is *strongly* confluent. Confluence implies that $\simeq_\beta$ is an equivalence relation, and that the system is deterministic in the sense that each net has at most one normal form. Strong confluence reinforces this determinism, because it implies that also the computation is unique, up to permutation of rules.

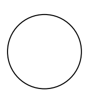
We remark here a substantial difference with respect to the λ-calculus, which is the absence of a meaningful concept of *strategy*, at least as far as the efficiency of reduction is concerned. All reductions leading from a net to its normal form have the same length and require the same work; in particular, if a net is normalizable, then it is strongly so.

2.3. *Basic nets*

Vicious circles and cut-free nets. A net may contain configurations which cannot be removed through interaction, like



in which clearly no cell can interact first (there is a sort of deadlock). The following case is yet simpler:



Deadlocked configurations like those above are called *vicious circles.*

Definition 2.1 (Straight path). Let μ be a net, let $i, j \in \mathsf{Ports}(\mu)$, and let p be a path from i to j in the graph-theoretical sense. We say that p is *straight* iff whenever p enters a cell through one of its auxiliary ports, it exits it by its principal port, and whenever p enters a cell through its principal port, it exits it by one of its auxiliary ports.

Notice that the last requirement implies that a straight path can never "bounce back" out of an ε cell.

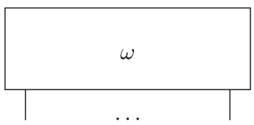
Definition 2.2 (Vicious circle). A *vicious circle* is a cyclic straight path never crossing two principal ports in a row.

A net containing no active pair and no vicious circle is said to be *cut-free* (or *reduced* in Lafont's terminology). A net admitting a cut-free form through reduction (necessarily unique by confluence) is said to be *total.*

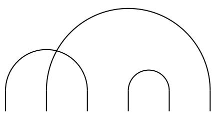
Cut-free nets are the "true" normal forms of the reduction; they can be seen as the final result of a computation. On the other hand, non-total nets represent error-bound computations, either diverging or leading to a deadlock.

Usually, when one translates classical computational models (e.g. Turing machines, the λ-calculus, etc.) into the interaction combinators (or more genrally into interaction nets), the encoding is such that vicious circles never arise through reduction. An example is the encoding of the **SK**-combinators we give in Sect. 2.4. The reader acquainted to linear logic will see that vicious circles are a legacy of proof-structures: incorrect proof-structures may reduce to nets containing cuts which are not reducible according to the standard cut-elimination procedure (for example an axiom whose conclusions are premises of a cut rule). This is indeed the reason behind our "cut-free" terminology.

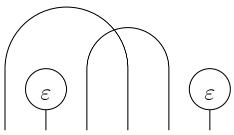
Wirings. A net containing no cells but just wires will be called a *wiring.* We shall represent the generic wiring as



The following is an example of wiring:

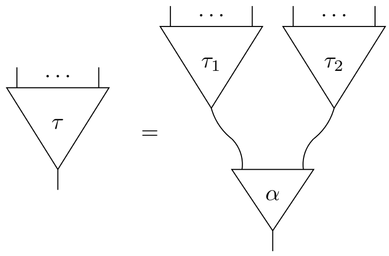


We also allow the free ports of a wiring to belong to ε cells, in which case we speak of an ε-wiring and we use the notation $\widetilde{\omega}$. The following is an example of ε-wiring:



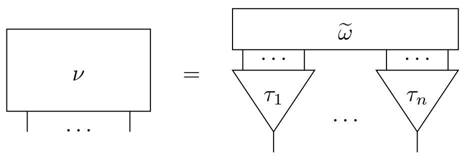
Trees. Trees are defined inductively as follows. The wiring

is a tree with one leaf (it is arbitrary which of the two extremities is the root and which is the leaf). If τ_1 and τ_2 are two trees with resp. n_1 and n_2 leaves, then we can define a tree τ with $n_1 + n_2$ leaves as



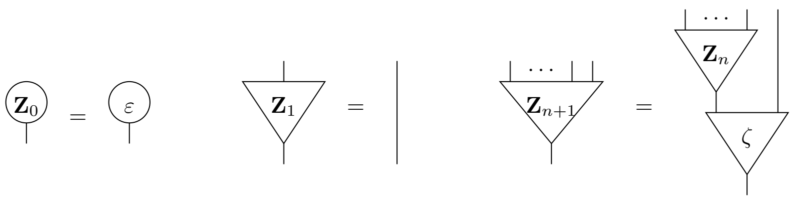
where $\alpha \in \{\delta, \zeta\}$.

It is not hard to verify that any cut-free net ν with n free ports can be decomposed in terms of trees and ε-wirings as follows:

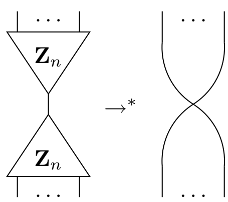


Principal nets, tests, packages. A principal net of arity n is either a single wire (in which case $n = 1$), or a cut-free net with n free auxiliary ports and 1 free principal port. If $n = 0$ (resp. $n = 1$), we say that the net is a *package* (resp. a *test*). Principal nets can be seen as “compound” cells, and will be drawn just like ordinary cells.

Notice that trees are special examples of principal nets. A particular family of principal nets of arity $n \geq 0$, the members of which are denoted $\mathbf{Z}_n$, and which are trees for $n \geq 1$, is defined as follows:



The reader can check that $\mathbf{Z}_n$ trees have the following annihilation property:



2.4. *Expressive power*

In spite of their great simplicity, the symmetric combinators are Turing-complete. This is a consequence of the following more general result:

Theorem 2.2 (Universality, Lafont (Lafont, 1997)). Any polarized interaction net system can be translated into the symmetric combinators.

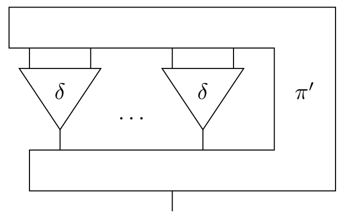
The symmetric combinators are a particular example of *non polarized* interaction net system; their polarized version is what Lafont calls the *directed combinators* (Lafont, 1997).

The definition of generic and polarized interaction net systems, and of translation from an interaction net system to another are beyond the scope of this paper. To understand the amplitude of Theorem 2.2, it is enough for the reader to know that Turing machines, one-dimensional cellular automata, the **SK** combinators, linear logic proof-nets and the λ-calculus can all be seen as polarized interaction net systems (for the latter two, see for example Lafont's survey (Lafont, 1995), the work of Ian Mackie and Jorge Pinto (Mackie and Pinto, 2002), and the work of Sylvain Lippi (Lippi, 2002)).

Even though Theorem 2.2 will not be proved here, and its meaning left to the intuition of the reader, we shall nevertheless give a direct proof of the expressive power of the symmetric combinators. As a matter of fact, in the remaining part of the section we shall see how the call-by-name **SK** combinators can be encoded inside this system.

First of all, let us introduce a fundamental construction due to Lafont (Lafont, 1997). Take a generic package π containing n δ cells; we can always write π as

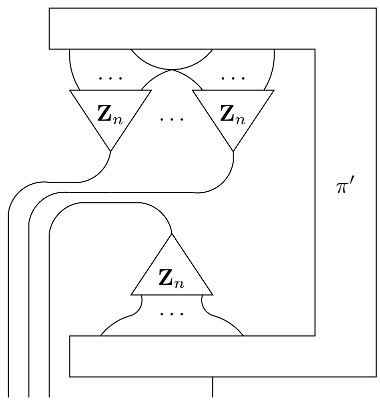
π =



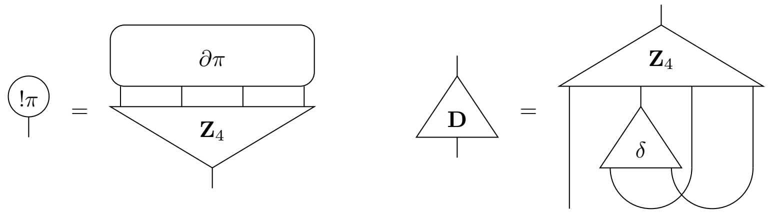
where π' contains no δ cell. We want to "abstract" the δ cells contained in π, forming a package $!\pi$ which does not contain δ cells, but from which π can be recovered.

Using the notations introduced above, we define

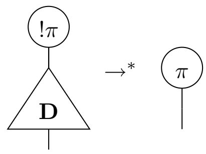
$\partial\pi$ =



Then, we put



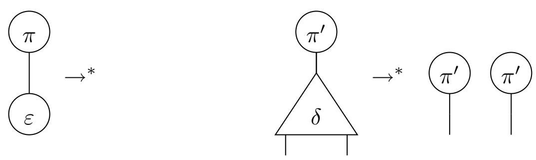
The reader can check that we have



The package $!\pi$ is called the *code* of π, and the principal net **D** is called the *decoder*.

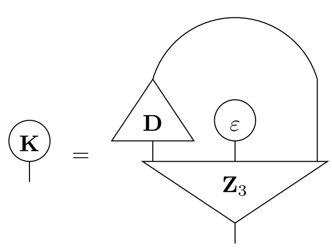
The following lemma is not hard to prove:

Lemma 2.3 (Erasure and duplication). Let π be a package, and π' a package containing no δ cells. Then, we have

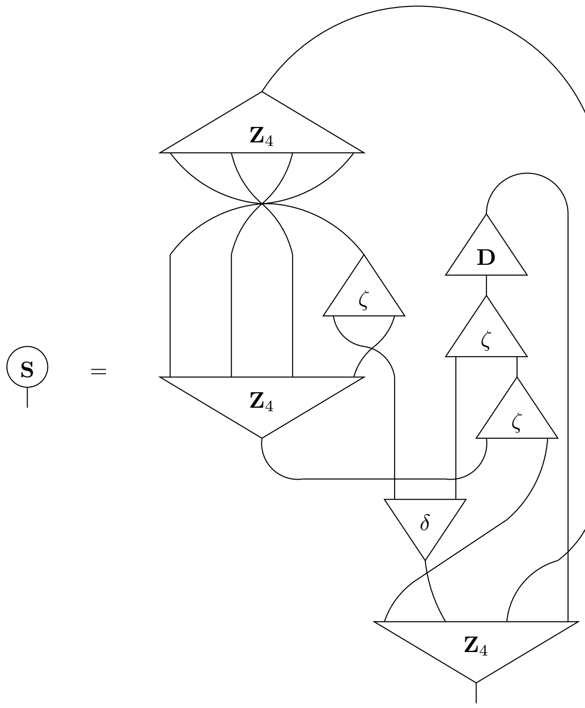


As a consequence, a package of the form $!\pi$ can be erased and duplicated.

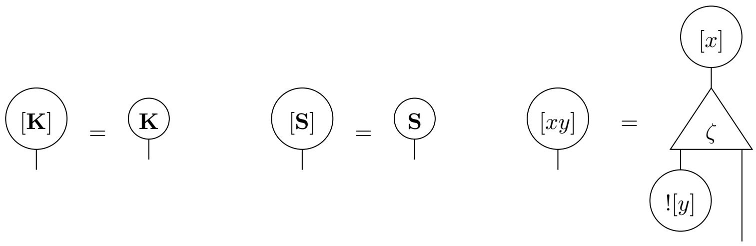
Now we build the two packages



and



and we define the translation $[\cdot]$ from **SK**-terms to nets:



If x, y are **SK**-terms, we write $x \succ y$ if x reduces to y through call-by-name reduction. For example, $\mathbf{K}(\mathbf{KSK})\mathbf{S} \succ \mathbf{KSK}$, but $\mathbf{K}(\mathbf{KSK})\mathbf{S} \not\succ \mathbf{KSS}$.

The translation defined above has the following property:

Theorem 2.4. Let x, y be **SK**-terms. Then, $x \succ^* y$ implies $[x] \to^* [y]$.

Proof. It is enough to check that, for all terms x, y, z,

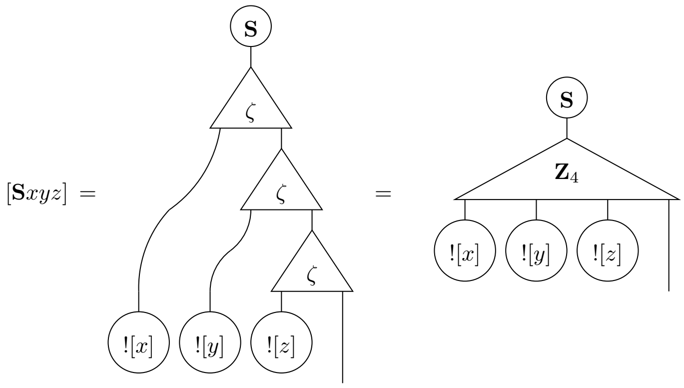
$$[\mathbf{K}xy] \to^* [x]$$

and

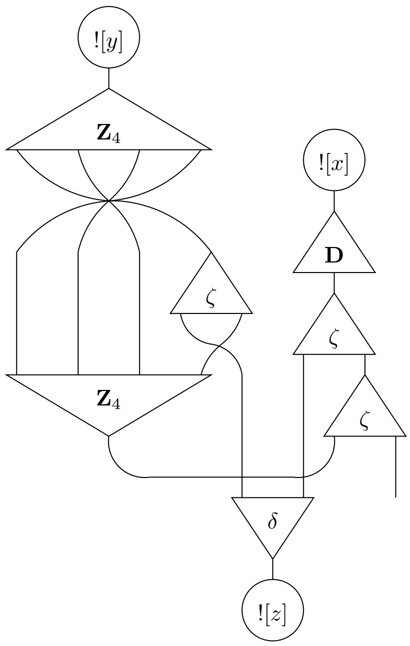
$$[\mathbf{S}xyz] \to^* [xz(yz)].$$

The first verification is easy and is left to the reader; Lemma 2.3 is needed for erasing. Here we shall concentrate on the second, which is more complex.

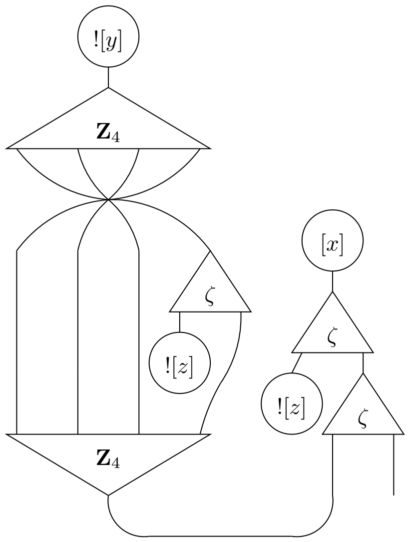
We have



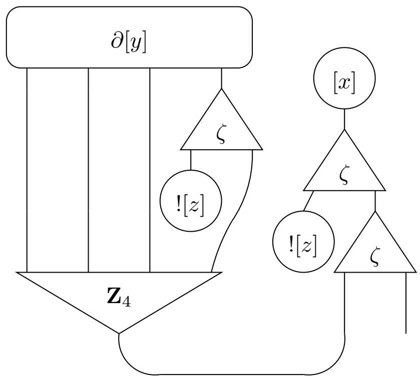
The $\mathbf{Z}_4$ tree annihilates with the $\mathbf{Z}_4$ tree contained in **S** (see p. 7), and we obtain



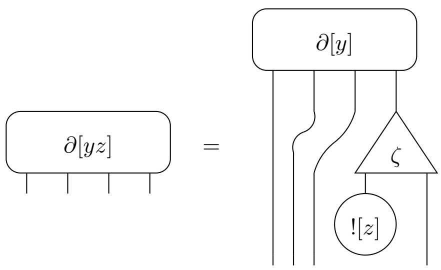
Now the decoder extracts $[x]$ from its code, and by Lemma 2.3 the code of $[z]$ is duplicated, so we get



At this point, the "topmost" $\mathbf{Z}_4$ tree annihilates with the $\mathbf{Z}_4$ tree inside $![y]$, and we are left with



Remember that $![z]$ does not contain any δ cell; this means that

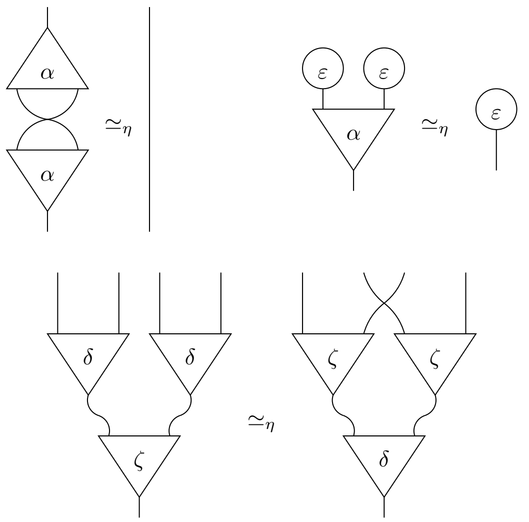


Therefore, the last net obtained is indeed equal to $[xz(yz)]$. □

2.5. *Observational equivalence*

In a recent work (Mazza, 2006), we have shown that there is a notion of observational equivalence for the symmetric combinators[‡] which can be defined directly on the syntax, and which is maximal on total nets with at least one free port.

We define the relation $\simeq_\eta$ as the reflexive, transitive, and contextual closure of the following equations:



As usual, $\alpha \in \{\delta, \zeta\}$. The above rules were already known to Lafont (top-right and bottom (Lafont, 1997)) and to Fernández and Mackie (top-left (Fernández and Mackie, 2003)). What is interesting is that they define an equivalence relation which is much like η-equivalence in the λ-calculus (this is the reason behind our notation).

We define $\simeq_{\beta\eta}$ as the transitive closure of $\simeq_\beta \cup \simeq_\eta$. If θ is a test and π a package, we write $\theta[\pi]$ for the net with one free port obtained by plugging π into the only free principal port of θ (or to any free port if θ is a wire).

In the following, ε stands for the package consisting of the sole ε combinator, while δ stands for the the package containing one δ cell whose auxiliary ports are connected by a wire.

Theorem 2.5 (Internal separation (Mazza, 2006)). Let π, π' be two packages such that $\pi \not\simeq_{\beta\eta} \pi'$. Then, there exists a test θ such that $\theta[\pi] \to^* \varepsilon$ and $\theta[\pi'] \to^* \delta$, or vice versa.

The above result, similar to Böhm's Theorem for the λ-calculus, is of fundamental importance for the theory of observational equivalence. In particular, it implies the following:

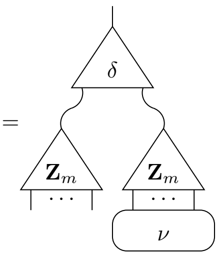
Proposition 2.6 (Maximality). $\simeq_{\beta\eta}$ is the greatest non-trivial congruence on total

[‡] Actually our results were originally formulated for the interaction combinators, but they apply without any problem to the symmetric combinators.

nets respecting reduction, i.e., if $\approx$ is a congruence on total nets such that $\simeq_\beta \subseteq \approx$, then either $\approx \subseteq \simeq_{\beta\eta}$, or $\mu \approx \mu'$ for all total nets μ, μ' with the same number of free ports.

Proof. Suppose that $\approx \not\subseteq \simeq_{\beta\eta}$. This means that there exist two total nets μ, μ', both with $n \geq 1$ free ports, such that $\mu \approx \mu'$ but $\mu \not\simeq_{\beta\eta} \mu'$. First of all, take any tree τ with n leaves and build the packages π, π' by "closing" resp. the cut-free form of μ and μ'. Since $\approx$ is a congruence, we have $\pi \approx \pi'$; moreover, it can be proved that just adding a tree "below" μ and μ' does not alter their differences with respect to $\simeq_{\beta\eta}$, so we still have $\pi \not\simeq_{\beta\eta} \pi'$. Therefore, Theorem 2.5 applies, giving us a test θ such that, for example, $\theta[\pi] \to^* \varepsilon$ and $\theta[\pi'] \to^* \delta$. Since $\approx$ is a congruence, and since $\simeq_\beta \subseteq \approx$, $\varepsilon \approx \delta$. But then consider the principal net

$\nu' =$



where ν is any total net with $m \geq 1$ free ports. If we plug ε into the only free principal port of ν' we obtain a net which is β-equivalent to a net consisting of m ε cells, which we call ε_m; on the other hand, if we do the same operation with the package δ, we obtain a net β-equivalent to ν. Hence, again by the fact that $\approx$ is a congruence containing $\simeq_\beta$, for any total ν with $m \geq 1$ free ports, $\nu \approx \varepsilon_m$; by transitivity of $\approx$, we get the thesis. □

The results of this section will guide us in our search for a denotational semantics for the symmetric combinators. In fact, once a good observational equivalence like $\simeq_{\beta\eta}$ is found, the goal of denotational semantics can be seen as the individuation of a mathematical structure in which $\simeq_{\beta\eta}$ becomes an equality.

3. Denotational semantics

As already recalled, interaction nets are a generalization of multiplicative proof-nets. Thus, in seeking a denotational semantics for the symmetric combinators, it seems natural to draw inspiration from linear logic.

The simplest denotational semantics of linear logic is the *relational semantics.* To our knowledge, relational semantics has not been formally introduced in any particular work. The best way to see it is perhaps as "coherent spaces without coherence"; it has been considered by many as the starting point for building other denotational semantics of linear logic (semantics = relational semantics + structure), as for example in the work of Thomas Ehrhard (Ehrhard, 2005). In the multiplicative case, the relational semantics is obtained from coherent spaces by simply ignoring the coherence relation: a formula A is interpreted by a set $|A|$, and a proof of A by a subset of $|A|$; no particular structure is attached or required on $|A|$. In categorical terms, the denotational interpretation takes place in the category **Rel** of sets and relations.

The denotational interpretation of a sequent calculus proof is, as usual, defined by induction. More interestingly, the interpretation can also be defined directly on proof-structures (so, in particular, on proof-nets) by means of *experiments* (Girard, 1987a). This will be our main source of inspiration, as a sequent calculus is obviously not available in our framework.

3.1. *Companion bijections*

The relational semantics of linear logic is typed; in particular, the two multiplicative connectives (which are the only ones of interest to us) are interpreted by the *cartesian product*: if the formulas A and B are interpreted resp. by $|A|$ and $|B|$, then $A \otimes B$ and $A \parr B$ are interpreted by $|A| \times |B|$. Our nets are not typed, which means that if we see our binary combinators as multiplicative rules, the natural thing to do would be to consider a set $\mathcal{D}$ in bijection with $\mathcal{D} \times \mathcal{D}$, i.e., any infinite set.

The presence of *two* combinators actually requires two bijections, and the $\delta\zeta$ commutation inspires the following:

Definition 3.1 (Companion bijections). Let $\mathcal{D}$ be an infinite set and

$$\langle \cdot, \cdot \rangle, [\cdot, \cdot] : \mathcal{D} \times \mathcal{D} \to \mathcal{D}$$

two bijections. $\langle \cdot, \cdot \rangle$ and $[\cdot, \cdot]$ are said to be *companions* iff, for all $a, b, c, d \in \mathcal{D}$,

$$\langle [a, b], [c, d] \rangle = [\langle a, c \rangle, \langle b, d \rangle].$$

Companion bijections do exist, as proved by the following example. Let $\mathcal{A}$ be any infinite set, and let $\beta : \mathcal{A} \times \mathcal{A} \to \mathcal{A}$ be a bijection. We denote by $\mathcal{S}(\mathcal{A})$ the set of all infinite sequences of elements of $\mathcal{A}$. Then we define two functions from $\mathcal{S}(\mathcal{A}) \times \mathcal{S}(\mathcal{A})$ to $\mathcal{S}(\mathcal{A})$ as follows:

$$\langle d, e \rangle_n = \begin{cases} d_k & \text{if } n = 2k \\ e_k & \text{if } n = 2k+1 \end{cases}$$

$$[d, e]_n = \beta(d_n, e_n)$$

In other words, $\langle \cdot, \cdot \rangle$ takes two sequences and builds a new one by interleaving them, while $[\cdot, \cdot]$ simply superposes the two sequences using the bijection β. The two functions are clearly bijections; moreover, given four sequences a, b, c, d, for even indexes we get

$$\langle [a, b], [c, d] \rangle_{2k} = [a, b]_k = \beta(a_k, b_k) = \beta(\langle a, c \rangle_{2k}, \langle b, d \rangle_{2k}) = [\langle a, c \rangle, \langle b, d \rangle]_{2k},$$

and similarly for odd indexes, which proves that the two bijections are indeed companions.

If $\mathcal{A}$ is a denumerably infinite pointed set, whose distinguished element we call zero, the same example can be built on $\Phi(\mathcal{A})$, the set of infinite sequences of elements of $\mathcal{A}$ which are almost everywhere zero. So companion bijections exist on countable sets as well, in particular on $\Phi(\mathbb{N})$, which by the fundamental theorem of arithmetics is isomorphic (as a monoid) to $\mathbb{N}^*$, the strictly positive integers.

3.2. *Experiments and interpretation*

The previous section suggests the following definition:

Definition 3.2 (Interaction set). An *interaction set* is an infinite pointed set $\mathcal{D}$ (the distinguished element being denoted by 0), admitting two companion bijections

$$\langle\cdot,\cdot\rangle, [\cdot,\cdot] : \mathcal{D} \times \mathcal{D} \to \mathcal{D}$$

such that $\langle 0, 0\rangle = [0, 0] = 0$.

The examples above show that interaction sets exist; for instance, in $\Phi(\mathbb{N})$, 0 is the everywhere-zero sequence.

In the following, $\mathcal{D}$ will be an interaction set.

Definition 3.3 (Experiment). Let μ be a net. An *experiment* on μ is a function $e : \mathsf{Ports}(\mu) \to \mathcal{D}$ such that:

(a) if $i, j \in \mathsf{Ports}(\mu)$ are connected by a wire, then $e(i) = e(j)$;
(b) if $i, j, k \in \mathsf{Ports}(\mu)$ are resp. the left auxiliary, right auxiliary, and principal port of a δ cell of μ, then $e(k) = \langle e(i), e(j)\rangle$;
(c) if $i, j, k \in \mathsf{Ports}(\mu)$ are resp. the left auxiliary, right auxiliary, and principal port of a ζ cell of μ, then $e(k) = [e(i), e(j)]$;
(d) if $i \in \mathsf{Ports}(\mu)$ is the principal port of an ε cell, then $e(i) = 0$.

If $k_1, \ldots, k_n$ are the free ports of μ, with $n \geq 1$, the tuple $(e(k_1), \ldots, e(k_n))$ is called the *result* of the experiment and is denoted by $|e|$.

In the following, we write $\mathcal{D}^n$ for $\mathcal{D} \times \cdots \times \mathcal{D}$ n times.

Definition 3.4 (Interpretation). Let μ be a net with $n \geq 1$ free ports. The *interpretation* of μ in $\mathcal{D}$, written $[\![\mu]\!]$, is defined to be the subset of $\mathcal{D}^n$ containing the results of all possible experiments on μ:

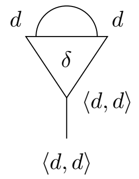
$$[\![\mu]\!] = \{|e| \ ; \ e \text{ experiment on } \mu\}.$$

We can give a few examples to see some concrete applications of the above definition. Consider the package ε consisting of a single ε cell. There is only one possible experiment on it, which assigns 0 to the principal port of the ε cell and to the free port of the package, so $[\![\varepsilon]\!] = \{0\}$. This is the smallest possible interpretation a net can receive; as a matter of fact, the interpretation of a net is always a pointed set, i.e., it is never empty. This is an immediate consequence of the definition of experiment and of the fact that 0 is a fixpoint of both bijections:

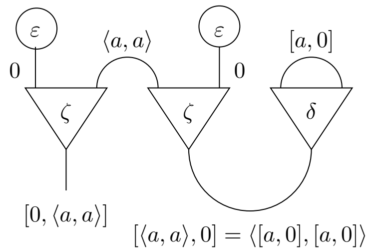
Proposition 3.1. For all μ with at least one free port, $(0, \ldots, 0) \in [\![\mu]\!]$.

Proof. The function assigning 0 to all ports of μ is an experiment. □

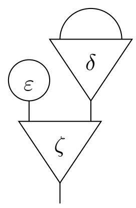
If we take the package δ consisting of a single δ cell whose auxiliary ports are connected by a wire, we clearly have that all possible experiments are of the following form:



so $[\![\delta]\!] = \{\langle d, d\rangle \ ; \ d \in \mathcal{D}\}$. Just as an axiom in linear logic, the net ω with 2 free ports consisting of a single wire is interpreted by the diagonal relation in $\mathcal{D} \times \mathcal{D}$: $[\![\omega]\!] = \{(d, d) \ ; \ d \in \mathcal{D}\}$. The following is a more involved example:



In the above picture, a label d on a wire means that the two ports connected by the wire have both been assigned the element d by the experiment; a is a generic element of $\mathcal{D}$. We therefore see that, if we call μ the above net, we have $[\![\mu]\!] = \{[0, \langle a, a\rangle] \ ; \ a \in \mathcal{D}\}$. The reader can check that this is also the interpretation of the following net



which is the cut-free of form of μ.

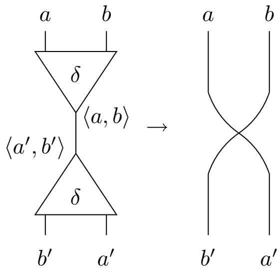
As a matter of fact, the interpretation is a denotational semantics for the symmetric combinators, i.e., it is preserved under reduction. Additionally, it also models $\simeq_\eta$.

Lemma 3.2 (Stability under reduction). Let μ, μ' be two nets with at least one free port. Then, $\mu \to \mu'$ implies $[\![\mu]\!] = [\![\mu']\!]$.

Proof. We need to show that for any experiment e on μ, there exists an experiment e' on μ' yielding the same result, and vice-versa. Since the rewriting is local, it actually suffices to show that, for all reduction rules, the assignment given by the experiment e on the interface of the left member of the rule can be reproduced by e' on the interface of the right member, and vice-versa; at this point e and e' can be assumed to be equal everywhere else, which guarantees that the results are the same.

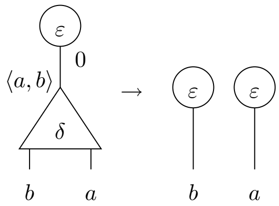
The case of the $\varepsilon\varepsilon$ annihilation is trivial: e' is just e restricted to the ports which do not disappear after the application of the rule.

The $\delta\delta$ and $\zeta\zeta$ annihilations are structurally identical, so we shall only consider the first one:



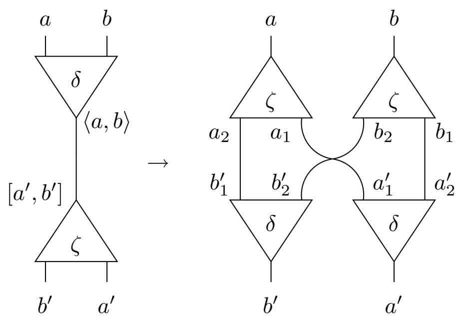
Here, a, b, a', b' are generic elements of $\mathcal{D}$. The assignment on the left hand side must satisfy $\langle a, b\rangle = \langle a', b'\rangle$, which by the injectivity of $\langle\cdot,\cdot\rangle$ implies $a = a'$ and $b = b'$, therefore the assignment on the right hand side is correct. The converse is trivial.

For what concerns the commutations, the $\delta\varepsilon$ and $\zeta\varepsilon$ commutations are again structurally identical, so we only need to consider the first one:



Again, by injectivity of $\langle\cdot,\cdot\rangle$, the requirement on the left hand side that $\langle a, b\rangle = 0$ implies $a = 0$ and $b = 0$, so the assignment on the right hand side is correct. The converse holds because of the hypothesis that $\langle 0, 0\rangle = 0$.

On the other hand, for the $\delta\zeta$ commutation, we get



In the left hand side we must have $\langle a, b\rangle = [a', b']$. By surjectivity of $\langle\cdot,\cdot\rangle$ and $[\cdot,\cdot]$, there exist $a_1, a_2, a'_1, a'_2, b_1, b_2, b'_1, b'_2 \in \mathcal{D}$ such that $a = [a_1, a_2]$, $b = [b_1, b_2]$, $a' = \langle a'_1, a'_2\rangle$, and $b' = \langle b'_1, b'_2\rangle$. The above equality and the fact that $\langle\cdot,\cdot\rangle$ and $[\cdot,\cdot]$ are companions imply $[\langle a'_1, a'_2\rangle, \langle b'_1, b'_2\rangle] = [\langle a_1, b_1\rangle, \langle a_2, b_2\rangle]$, which by injectivity of $\langle\cdot,\cdot\rangle$ and $[\cdot,\cdot]$ in turn implies $a'_1 = a_1$, $a'_2 = b_1$, $b'_1 = a_2$, and $b'_2 = b_2$. Therefore, the assignment defined above for the right hand side of the rule is correct. Conversely, if we know from the right hand side that $a'_1 = a_1 = c_1$, $a'_2 = b_1 = c_2$, $b'_1 = a_2 = c_3$, and $b'_2 = b_2 = c_4$, we have $a = [c_1, c_3]$, $b = [c_2, c_4]$, $a' = \langle c_1, c_2\rangle$, and $b' = [c_3, c_4]$, which means that $\langle a, b\rangle = \langle [c_1, c_3], [c_2, c_4]\rangle$

and $[a', b'] = [\langle c_1, c_2\rangle, \langle c_3, c_4\rangle]$. But since $\langle\cdot,\cdot\rangle$ and $[\cdot,\cdot]$ are companions, this implies that $\langle a, b\rangle = [a', b']$, so the assignment on the left hand side is correct. □

In the following, if μ is a net with n free ports, we call a *context* for μ any net C with at least $n+1$ free ports, n of which are in bijection with the free ports of μ; the application of the context to μ, written $C[\mu]$, is the net obtained by plugging each free port of μ to the corresponding free port of C.

Lemma 3.3 (Congruence). Let μ, μ' be two nets with n free ports, and let C be a context for μ, μ'. Then, $[\![\mu]\!] = [\![\mu']\!]$ implies $[\![C[\mu]]\!] = [\![C[\mu']]\!]$.

Proof. An experiment on $C[\mu]$ must be the union of an experiment e on μ and an experiment f on C, such that, if i and j are two free ports of resp. μ and C which are connected in $C[\mu]$, $e(i) = f(j)$. The same holds for $C[\mu']$, so, if m is the number of free ports of $C[\mu]$ and $C[\mu']$, we have

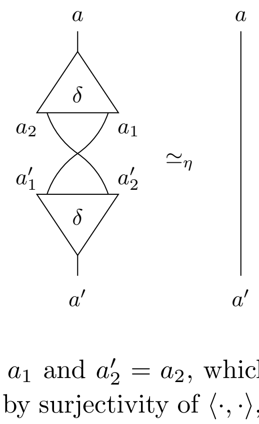
$$[\![C[\mu]]\!] = \{d \in \mathcal{D}^m \ ; \ (d, c) \in [\![C]\!] \text{ and } c \in [\![\mu]\!]\}$$

$$[\![C[\mu']]\!] = \{d' \in \mathcal{D}^m \ ; \ (d', c) \in [\![C]\!] \text{ and } c \in [\![\mu']\!]\},$$

from which we clearly see that if $[\![\mu]\!] = [\![\mu']\!]$, then $[\![C[\mu]]\!] = [\![C[\mu']]\!]$. □

Lemma 3.4 (Extensionality). Let μ, μ' be two nets with at least one free port. Then, $\mu \simeq_\eta \mu'$ implies $[\![\mu]\!] = [\![\mu']\!]$.

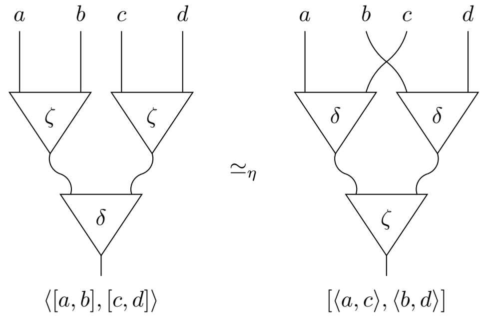
Proof. The proof follows exactly the same argument used for Lemma 3.2. We start by considering the η-expansion for δ, the corresponding rule involving ζ being structurally identical:



The left hand side imposes $a'_1 = a_1$ and $a'_2 = a_2$, which implies $a = a'$. Conversely, the right hand side imposes $a'_1 = a_1$; by surjectivity of $\langle\cdot,\cdot\rangle$, a_1 and a_2 such that $\langle a_1, a_2\rangle = a$ exist, so the assignment on the left hand side is correct.

The cases of the $\delta\varepsilon$ and $\zeta\varepsilon$ commutations are trivial, and rest upon the fact that $\langle 0, 0\rangle = [0, 0] = 0$.

The case of the $\delta\zeta$ commutation is also trivial:



The two experiments are the same thanks to the fact that $\langle\cdot,\cdot\rangle$ and $[\cdot,\cdot]$ are companions. ☐

Lemmas 3.2 and 3.3 together prove that $[\![\cdot]\!]$ is a denotational semantics of $\simeq_\beta$: it is preserved under reduction, and it is a congruence. Lemma 3.4 proves that the semantics actually models $\simeq_{\beta\eta}$. Moreover, the examples given at p. 16 show that there exist two nets μ, μ' such that $[\![\mu]\!] \neq [\![\mu']\!]$, so the semantics is non-trivial.

The fact that $\mu \simeq_{\beta\eta} \mu'$ implies $[\![\mu]\!] = [\![\mu']\!]$ holds for *any* μ, μ'; if we restrict to total nets, then the converse holds as well.

Theorem 3.5 (Injectivity on total nets). Let μ, μ' be two total nets with at least one free port. Then, $\mu \simeq_{\beta\eta} \mu'$ iff $[\![\mu]\!] = [\![\mu']\!]$.

Proof. As already said, the implication $\mu \simeq_{\beta\eta} \mu' \;\Rightarrow\; [\![\mu]\!] = [\![\mu']\!]$ is a consequence of Lemmas 3.2 and 3.4. For the converse, which is the actual injectivity property, we can restrict to packages, since:

— total nets containing active pairs can be reduced and their cut-free form considered;
— cut-free nets with more than one free port can be “closed” by means of any fixed tree, and the argument below can thus be easily generalized.

So take two packages π, π' such that $\pi \not\simeq_{\beta\eta} \pi'$. By Theorem 2.5, there exists a test θ such that $\theta[\pi] \to^* \varepsilon$ and $\theta[\pi'] \to^* \delta$ (or viceversa, but we do not lose generality in assuming this situation), where ε is the package consisting of a single ε cell and δ is the package consisting of a single δ cell with its auxiliary ports connected by a wire. By Lemma 3.2, and by what we have seen at p. 16, we have

$$\begin{array}{rcl} [\![\theta[\pi]]\!] & = & \{0\} \\ [\![\theta[\pi']]\!] & = & \{\langle d, d\rangle \;;\; d \in \mathcal{D}\}. \end{array}$$

Since $\mathcal{D}$ is infinite, clearly $[\![\mu]\!] \neq [\![\mu']\!]$, which by Lemma 3.3 implies $[\![\pi]\!] \neq [\![\pi']\!]$. ☐

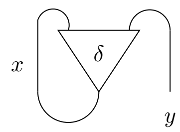
One may wonder whether injectivity with respect to $\simeq_{\beta\eta}$ can be extended to non-total nets. The answer is in general negative, as the following example shows. Let $\Phi(\mathbb{N})$ be the space of almost-everywhere-null sequences of natural numbers. The distinguished element 0 is the everywhere-zero sequence. We know that $\Phi(\mathbb{N})$ admits the two companion bijections defined in Sect. 3.1; in particular, we recall the definition of $\langle \cdot, \cdot \rangle$:

$$\langle x, y \rangle_n = \begin{cases} x_k & \text{if } n = 2k \\ y_k & \text{if } n = 2k+1 \end{cases}$$

The result below is not hard to verify:

Lemma 3.6. $\langle x, y \rangle = x$ iff $y = 0$ and x is a sequence such that, for all $n \geq 1$, $x_n = 0$.

Now consider the following non-total net μ:

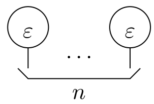


The labels indicate that the generic experiment on μ in $\Phi(\mathbb{N})$ assigns the sequence x to the left auxiliary and principal ports of the δ cell, and the sequence y to the right auxiliary port of the δ cell and to the free port of μ. Therefore, we have

$$[\![\mu]\!] = \{y \in \Phi(\mathbb{N}) \; ; \; \exists x \in \Phi(\mathbb{N}).\langle x, y \rangle = x\}.$$

But Lemma 3.6 proves that the only possible such y is 0, so $[\![\mu]\!] = \{0\} = [\![\varepsilon]\!]$, where ε is the package consisting of the sole ε combinator. Now, both μ and ε do not contain active pairs, hence the only hope of rewriting one into the other is through η-equivalence. But a simple inspection of the η-rules of p. 13 reveals that the presence of ε cells is preserved by $\simeq_\eta$: no rule can produce a net containing no ε cell from a net containing one, and no rule can add ε cells if there are none. Therefore, $\mu \not\simeq_{\beta\eta} \varepsilon$, and yet $[\![\mu]\!] = [\![\varepsilon]\!]$.

By the way, we remark that identifying the net μ above with the ε combinator is observationally sound. Indeed, if we define a *blind net* as a net such that all of its free ports are auxiliary and will never become principal through reduction, then it is consistent to identify any blind net with n free ports to any net $\beta\eta$-equivalent to the following:



As a matter of fact, they interact in the same way. In particular, this means that Proposition 2.6 does not hold for generic nets.

The reader may wonder whether this failure of injectivity goes beyond the particular model chosen above, and actually holds for any interaction set. Although at present we are not able to answer the question, we tend to believe that this is indeed the case, and we actually see this phenomenon as quite natural, considering the above remark about observational equivalence. Indeed, the ε combinator and the deadlocked net used in the counter-example can be seen as the equivalents of two *unsolvable* terms in the λ-calculus: like Ω and $\lambda x.\Omega$, they are not $\beta\eta$-equivalent, but are identified by any sensible model.

3.3. *Full completeness*

If $\mathcal{D}$ is an interaction set, even denumerable, for obvious resons of cardinality not every subset of $\mathcal{D}^n$ is the interpretation of some net. In this section we characterize those that are interpretations of *total* nets.

In the following, $\mathcal{D}$ is a generic interaction set.

Definition 3.5 (Bracket expression). Let x range over a denumerable set of variables. A *simple bracket expression* b is a syntactical expression belonging to the following grammar:

$$\mathsf{b} ::= x \mid \mathsf{0} \mid \langle \mathsf{b}, \mathsf{b} \rangle \mid [\mathsf{b}, \mathsf{b}]$$

A *bracket expression* is a tuple of simple bracket expressions.

We denote by $\mathsf{var}(\mathsf{b})$ the set of variables occurring in the simple bracket expression b. We define as usual the substitution of a variable y in place of x in b, denoted by $\mathsf{b}[y/x]$. If $x \in \mathsf{var}(\mathsf{b})$ and $z \notin \mathsf{var}(\mathsf{b})$, then we say that b and $\mathsf{b}[x/z]$ are α-equivalent.

Similarly, if $\mathsf{B} = (\mathsf{b}_1, \ldots, \mathsf{b}_n)$ is a bracket expression, we define $\mathsf{var}(\mathsf{B}) = \mathsf{var}(\mathsf{b}_1) \cup \cdots \cup \mathsf{var}(\mathsf{b}_n)$, i.e., variables are shared by the simple expressions in the tuple, and substitution is performed on the whole expression; α-equivalence is trivially extended, and bracket expressions are always considered modulo α-equivalence.

If B is a bracket expression containing n simple expressions such that $\mathsf{var}(\mathsf{B}) \subseteq \{x_1, \ldots, x_m\}$, and if $d_1, \ldots d_m \in \mathcal{D}$, we can define an element $\mathsf{B}\{x_1 := d_1, \ldots, x_m := d_m\}$ of $\mathcal{D}^n$ in the obvious way: just assign each d_i to x_i, and compute the expression considering the symbols $\mathsf{0}$, $\langle \cdot, \cdot \rangle$ and $[\cdot, \cdot]$ as resp. the distinguished element and the two bijections of $\mathcal{D}$. For example, suppose that d, e, f are three elements of $\mathcal{D}$ such that $f = [\langle d, e \rangle, d]$; then, if $\mathsf{B} = [\langle x, y \rangle, x]$, we have $\mathsf{B}\{x := d, y := e\} = f$. In this way, each bracket expression B containing n simple bracket expressions and a total of m variables defines a function from $\mathcal{D}^m$ to $\mathcal{D}^n$. Because of the obvious shortage of bracket expressions, the assignment cannot be surjective; it is not injective either, as the expressions $\mathsf{0}$ and $\langle \mathsf{0}, \mathsf{0} \rangle$ show (they both represent the constant function 0).

In the following, $\mathsf{occ}_x(\mathsf{B})$ denotes the number of occurrences of the variable x in the bracket expression B.

Definition 3.6 (Balanced bracket expression). A bracket expression B is *balanced* iff, for any variable x, either $\mathsf{occ}_x(\mathsf{B}) = 0$ or $\mathsf{occ}_x(\mathsf{B}) = 2$. A function from $\mathcal{D}^m$ to $\mathcal{D}^n$ is said to be *balanced* if it can be defined through a balanced bracket expression containing n expressions using m variables. A set $\mathcal{B} \subseteq \mathcal{D}^n$ is called *balanced* if it is the codomain of a balanced function.

Theorem 3.7 (Full completeness). If μ is a total net with $n \geq 1$ free ports, then $[\![\mu]\!]$ is balanced. Conversely, if $n \geq 1$, given a balanced set $\mathcal{B} \subseteq \mathcal{D}^n$, there exists a cut-free net μ with n free ports such that $[\![\mu]\!] = \mathcal{B}$.

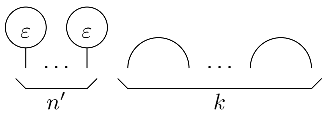
Proof. Since μ is total, we can consider its cut-free form ν. By Lemma 3.2, $[\![\mu]\!] = [\![\nu]\!]$; now, if we remember that cut-free nets are trees of δ and ζ cells with wires and ε cells "on top", it is clear that the first statement is a straight-forward consequence of the definition of experiments.

For what concerns the converse, let $\mathcal{B} \subseteq \mathcal{D}^n$ be balanced, and let B be the bracket expression such that $\mathcal{B}$ is the codomain of B. Simple bracket expressions can obviously be provided with a complexity measure $\sharp(\cdot)$, which is the total number of binary syntactical constructs used:

- $\sharp(x) = \sharp(\mathsf{0}) = 0$;
- $\sharp(\langle \mathsf{b}_1, \mathsf{b}_2 \rangle) = \sharp([\mathsf{b}_1, \mathsf{b}_2]) = \sharp(\mathsf{b}_1) + \sharp(\mathsf{b}_2) + 1$.

For an expression $\mathsf{B} = (\mathsf{b}_1, \ldots, \mathsf{b}_n)$, we pose $\sharp(\mathsf{B}) = \sharp(\mathsf{b}_1) + \cdots + \sharp(\mathsf{b}_n)$.

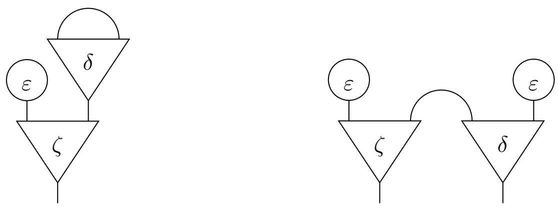
We can then reason by induction on $\sharp(\mathsf{B})$. If $\sharp(\mathsf{B}) = 0$, knowing that B is balanced, we can assume without loss of generality that $\mathsf{B} = (\mathsf{0}, \ldots, \mathsf{0}, x_1, x_1, \ldots, x_k, x_k)$. It is easy to see that if we interpret the net



(where $n' + k = n$), we obtain exactly $\mathcal{B}$.

If $\sharp(\mathsf{B}) > 0$, then we can assume without loss of generality that $\mathsf{B} = (\langle \mathsf{b}_1', \mathsf{b}_1'' \rangle, \mathsf{b}_2, \ldots, \mathsf{b}_n)$ or $\mathsf{B} = ([\mathsf{b}_1', \mathsf{b}_1''], \mathsf{b}_2, \ldots, \mathsf{b}_n)$. In both cases, the bracket expression $\mathsf{B}' = (\mathsf{b}_1', \mathsf{b}_1'', \mathsf{b}_2, \ldots, \mathsf{b}_n)$ has measure strictly smaller than B, so the induction hypothesis applies, giving us a net μ' with $n+1$ free ports such that $[\![\mu']\!]$ is the codomain of B'. Clearly, adding a δ or a ζ cell (according to the shape of B) to μ' yields a net with n free ports μ such that $[\![\mu]\!] = \mathcal{B}$. □

The above proof actually tells us that, for $n \geq 1$, the balanced subsets of $\mathcal{D}^n$ are in bijection with the cut-free nets with n free ports. As an example, take the balanced sets induced by the expressions $[\mathsf{0}, \langle x, x \rangle]$ and $([\mathsf{0}, x], \langle x, \mathsf{0} \rangle)$; they correspond to the following two nets:



This is not surprising at all: a balanced expression is just a list of trees, the connections between leaves being expressed as pairs of occurrences of the same variable. Therefore, balanced expressions are nothing but an alternative, linear syntax for cut-free nets.

It must also be mentioned that this full completeness result is very similar to that proved by Michele Pagani for multiplicative linear logic proof-nets (Pagani, 2006); as a matter of fact, this strengthens the claim that the symmetric combinators are an extension of multiplicative proof-structures, and may be deeply related to them.

4. Algebraic semantics

In this section we develop an algebraic semantics for the symmetric combinators, in the style of Girard's Geometry of Interaction (GoI) (Girard, 1989), already sketched by Lafont (Lafont, 1997). We do so in connection with the denotational semantics introduced above, and prove that there is a strong link between the two.

4.1. *Interaction monoids*

Up to here, we have seen that any infinite set $\mathcal{D}$ can serve as the domain for the denotational semantics of nets of symmetric combinators. We shall see that it may be of interest to add some algebraic structure to $\mathcal{D}$; the least we can require is that $\mathcal{D}$ is a monoid.

In the following, the composition $u \circ v$ of two monoid endomorphisms is denoted simply uv.

Definition 4.1 (Interaction monoid). An *interaction monoid* is a commutative monoid $(M, +, 0)$ admitting eight endomorphisms $\mathbf{c}, \mathbf{c}^*, \mathbf{d}, \mathbf{d}^*, \mathbf{f}, \mathbf{f}^*, \mathbf{g}, \mathbf{g}^*$ such that the functions $\langle x, y\rangle = \mathbf{c}(x) + \mathbf{d}(y)$ and $[x, y] = \mathbf{f}(x) + \mathbf{g}(y)$ are companion isomorphisms between $M \oplus M$ and M, and $\mathbf{c}^*, \mathbf{d}^*$ and $\mathbf{f}^*, \mathbf{g}^*$ are their respective projections.

Proposition 4.1. A commutative monoid $(M, +, 0)$ is an interaction monoid iff there exist eight endomorphisms $\mathbf{c}, \mathbf{c}^*, \mathbf{d}, \mathbf{d}^*, \mathbf{f}, \mathbf{f}^*, \mathbf{g}, \mathbf{g}^*$ of M such that:

1 $\mathbf{c}^*\mathbf{c} = \mathbf{d}^*\mathbf{d} = \mathbf{f}^*\mathbf{f} = \mathbf{g}^*\mathbf{g} = \mathbf{1}$, where $\mathbf{1}$ is the identity on M;
2 $\mathbf{c}^*\mathbf{d} = \mathbf{d}^*\mathbf{c} = \mathbf{f}^*\mathbf{g} = \mathbf{g}^*\mathbf{f} = \mathbf{0}$, where $\mathbf{0}$ is the everywhere-zero endomorphism on M;
3 $\mathbf{cc}^* + \mathbf{dd}^* = \mathbf{ff}^* + \mathbf{gg}^* = \mathbf{1}$;
4 $\mathbf{c}, \mathbf{c}^*, \mathbf{d}, \mathbf{d}^*$ commute with $\mathbf{f}, \mathbf{f}^*, \mathbf{g}, \mathbf{g}^*$.

Proof. Let us first prove that Definition 4.1 implies the four statements above:

1 By definition, for every $x \in M$, $\mathbf{c}(x) = \langle x, 0\rangle$, and by the hypothesis that $\mathbf{c}^*$ is the left projection of $\langle\cdot,\cdot\rangle$, we obtain $\mathbf{c}^*\mathbf{c}(x) = \mathbf{c}^*(\langle x, 0\rangle) = x$. The same applies to the other annihilations.
2 As above, for every $x \in M$ we have $\mathbf{d}(x) = \langle 0, x\rangle$, from which we obtain $\mathbf{c}^*\mathbf{d}(x) = \mathbf{c}^*(\langle 0, x\rangle) = 0$. The same applies to the other annihilations.
3 From the surjectivity of $\langle\cdot,\cdot\rangle$, given $x \in M$ we know that there exist $y, z \in M$ such that $x = \langle y, z\rangle$. Then, we have

$$(\mathbf{cc}^* + \mathbf{dd}^*)(x) = \mathbf{cc}^*(x) + \mathbf{dd}^*(x) = \mathbf{c}(y) + \mathbf{d}(z) = \langle y, z\rangle = x.$$

The case $\mathbf{ff}^* + \mathbf{gg}^* = \mathbf{1}$ is identical.
4 To prove that $\mathbf{c}, \mathbf{d}$ commute with $\mathbf{f}, \mathbf{g}$ simply consider that, by the companion hypothesis, for all $x \in M$, $\langle [x, 0], 0\rangle = [\langle x, 0\rangle, 0]$, from which we get $\mathbf{cf}(x) = \mathbf{fc}(x)$, and $\langle [0, x], 0\rangle = [0, \langle x, 0\rangle]$, from which we obtain $\mathbf{cg}(x) = \mathbf{gc}(x)$, and so on.
To prove that $\mathbf{c}, \mathbf{d}$ commute to $\mathbf{f}^*, \mathbf{g}^*$ consider, given a generic $x \in M$, the (unique) decomposition $x = [y, z]$, so that

$$\mathbf{f}^*\mathbf{c}(x) = \mathbf{f}^*\mathbf{cf}(y) + \mathbf{f}^*\mathbf{cg}(z) = \mathbf{f}^*\mathbf{fc}(y) + \mathbf{f}^*\mathbf{gc}(z) = \mathbf{c}(y) = \mathbf{cf}^*(x),$$

where we have used point 1 and 2 proved above. The other cases are handled similarly, as also the commutations between $\mathbf{f}, \mathbf{g}$ and $\mathbf{c}^*, \mathbf{d}^*$.
To prove that $\mathbf{c}^*, \mathbf{d}^*$ commute to $\mathbf{f}^*, \mathbf{g}^*$, we consider the same decomposition above for the generic $x \in M$, and we obtain

$$\mathbf{f}^*\mathbf{c}^*(x) = \mathbf{f}^*\mathbf{c}^*\mathbf{f}(y) + \mathbf{f}^*\mathbf{c}^*\mathbf{g}(z) = \mathbf{f}^*\mathbf{f}\mathbf{c}^*(y) + \mathbf{f}^*\mathbf{g}\mathbf{c}^*(z) = \mathbf{c}^*(y) = \mathbf{c}^*\mathbf{f}^*(x),$$

and similarly for the other cases.

Assume now that the eight endomorphisms verify the four statements above. The fact that the maps $(x, y) \mapsto \mathbf{c}(x) + \mathbf{d}(y)$ and $(x, y) \mapsto \mathbf{f}(x) + \mathbf{g}(y)$ are homomorphisms from $M \oplus M$ to M is obvious; we need to prove that they are bijective. We shall do it for the first map, the second being structurally identical.

Suppose that, given $x, x', y, y' \in M$, $\mathbf{c}(x) + \mathbf{d}(y) = \mathbf{c}(x') + \mathbf{d}(y')$; then, applying $\mathbf{c}^*$ (resp. $\mathbf{d}^*$) to both sides of the equation and using points 1 and 2, we get $x = x'$ (resp. $y = y'$), which proves injectivity. For what concerns surjectivity, by point 3 for any element $x \in M$ there exist $y, z \in M$ such that $x = \mathbf{c}(y) + \mathbf{d}(z)$: just pose $y = \mathbf{c}^*(x)$ and $z = \mathbf{d}^*(x)$. The fact that $\mathbf{c}^*, \mathbf{d}^*$ are the projections associated to this isomorphism is trivial.

We are left to proving that the two isomorphisms are companions. If we pose $\langle x, y \rangle = \mathbf{c}(x) + \mathbf{d}(y)$ and $[x, y] = \mathbf{f}(x) + \mathbf{g}(y)$, using point 4 we have

$$\begin{aligned} \langle [w, x], [y, z] \rangle &= \mathbf{cf}(w) + \mathbf{cg}(x) + \mathbf{df}(y) + \mathbf{dg}(z) = \\ &\qquad \mathbf{fc}(w) + \mathbf{fd}(y) + \mathbf{gc}(x) + \mathbf{gd}(z) = [\langle w, y \rangle, \langle x, z \rangle], \end{aligned}$$

which completes the proof. □

If $(A, +, 0)$ is a commutative monoid, then the set $\mathcal{S}_2(A)$ of all sequences of elements of A indexed by pairs of non-negative integers is an example of interaction monoid. Addition is defined pointwise, and the neutral element is the everywhere-zero sequence; the eight endomorphisms are defined as follows:

$$\mathbf{c}(x)_{m,n} = \begin{cases} x_{k,n} & \text{if } m = 2k \\ 0 & \text{if } m = 2k+1 \end{cases} \qquad \mathbf{c}^*(x)_{m,n} = x_{2m,n}$$

$$\mathbf{d}(x)_{m,n} = \begin{cases} 0 & \text{if } m = 2k \\ x_{k,n} & \text{if } m = 2k+1 \end{cases} \qquad \mathbf{d}^*(x)_{m,n} = x_{2m+1,n}$$

$$\mathbf{f}(x)_{m,n} = \begin{cases} x_{m,k} & \text{if } n = 2k \\ 0 & \text{if } n = 2k+1 \end{cases} \qquad \mathbf{f}^*(x)_{m,n} = x_{m,2n}$$

$$\mathbf{g}(x)_{m,n} = \begin{cases} 0 & \text{if } n = 2k \\ x_{m,k} & \text{if } n = 2k+1 \end{cases} \qquad \mathbf{g}^*(x)_{m,n} = x_{m,2n+1}$$

from which it is not hard to check that points 1, 2, and 3 of Proposition 4.1 are satisfied. For what concerns point 4, just notice that $\mathbf{c}, \mathbf{c}^*, \mathbf{d}, \mathbf{d}^*$ and $\mathbf{f}, \mathbf{f}^*, \mathbf{g}, \mathbf{g}^*$ act on separate indexes, so all operations commute.

To better understand the example, observe that the two isomorphisms defined above build a new sequence by interleaving two sequences; one of them interleaves them "horizontally", the other "vertically". More precisely, our sequences being bidimensional, i.e., "sequences of sequences", in the first case we consider them as "sequences of columns", and interleave them horizontally; in the second case, we consider them as "sequences of rows", and interleave them vertically.

Notice that the same construction can be applied to almost-everywhere-null bidimensional sequences; in case A is countable, this yields a countable interaction monoid.

Interaction monoids are clearly interaction sets (the distinguished element is the zero of the monoid), so Definitions 3.3 and 3.4 can be applied just as they are, yielding a semantics that interprets a net with $n \geq 1$ free ports as a subset of M^n, where $M^n = M \oplus \cdots \oplus M$ n times.

Since we are considering monoids, it makes sense to add experiments pointwise, i.e., given a net μ and two experiments e_1, e_2 on μ over an interaction monoid M, we can define the function $e_1 + e_2$ from $\mathsf{Ports}(\mu)$ to M that associates to a port i the element $e_1(i) + e_2(i)$. One may wonder whether this yields another experiment; the answer is indeed positive:

Lemma 4.2 (Additivity). Let μ be a net, and e_1, e_2 two experiments on μ over an interaction monoid M. Then, $e_1 + e_2$ is an experiment.

Proof. The fact that $e_1 + e_2$ respects conditions (a) and (d) of Definition 3.3 is obvious. Conditions (b) and (c) are consequences of the fact that our companion bijections are monoid isomorphisms. For example, in the case of a δ cell, whose auxiliary and principal ports are resp. i, j, and k, we have

$$\begin{aligned}(e_1 + e_2)(k) &= e_1(k) + e_2(k) = \langle e_1(i), e_1(j)\rangle + \langle e_2(i), e_2(j)\rangle = \\ &= \langle e_1(i) + e_2(i), e_1(j) + e_2(j)\rangle = \langle (e_1 + e_2)(i), (e_1 + e_2)(j)\rangle.\end{aligned}$$

The case of a ζ cell is identical. □

Therefore, if μ is a net with $n \geq 1$ free ports, $[\![\mu]\!]$ is not just any subset of M^n, it is a *submonoid*:

Corollary 4.3. Let μ be a net with $n \geq 1$ free ports, and M an interaction monoid. Then, the interpretation of μ in M is a submonoid of M^n.

Proof. The result follows from the Additivity Lemma 4.2 and Proposition 3.1. □

4.2. *The GoI semantics*

Given an interaction monoid M, we shall now define a semantics which interprets a cut-free net with $n \geq 1$ free ports as an endomorphism of $M^n = M \oplus \cdots \oplus M$. This is just a reformulation of what already done by Lafont (Lafont, 1997), so no proofs will be given in this section.

In the following, we denote by W the sub-semiring (with unit) of End(M) generated by $\mathbf{c}, \mathbf{c}^*, \mathbf{d}, \mathbf{d}^*, \mathbf{f}, \mathbf{f}^*, \mathbf{g}, \mathbf{g}^*$.

Definition 4.2 (Weight). Let M be an interaction monoid, μ a net, and p a straight path of μ (see Definition 2.1. We define the *weight* of p in M, which is an element of W and is denoted $w(p)$, by induction on the length of p:

– p contains just one port: $w(p) = \mathbf{1}$ (the identity endomorphism);
– $p = p' \cdot i$, and the ending port of p' and i do not belong to the same cell: $w(p) = w(p')$.
– $p = p' \cdot i$, where p' ends with the left (resp. right) auxiliary port of a δ cell, and i is the principal port of the same δ cell: $w(p) = \mathbf{c}w(p')$ (resp. $w(p) = \mathbf{d}w(p')$);
– $p = p' \cdot i$, where p' ends with the principal port of a δ cell, and i is the left (resp. right) auxiliary port of the same δ cell: $w(p) = \mathbf{c}^*w(p')$ (resp. $w(p) = \mathbf{d}^*w(p')$);
– $p = p' \cdot i$, where p' ends with the left (resp. right) auxiliary port of a ζ cell, and i is the principal port of the same ζ cell: $w(p) = \mathbf{f}w(p')$ (resp. $w(p) = \mathbf{g}w(p')$);
– $p = p' \cdot i$, where p' ends with the principal port of a ζ cell, and i is the left (resp. right) auxiliary port of the same ζ cell: $w(p) = \mathbf{f}^*w(p')$ (resp. $w(p) = \mathbf{g}^*w(p')$).

Given a graph-theoretical path, one can always consider its *reversal*, i.e., the same path walked from target to source. Notice that the reversal of a straight path is still straight. The unit semiring W can be equipped with an involution $(\cdot)^*$:

— $(\mathbf{c})^* = \mathbf{c}^*$, $(\mathbf{c}^*)^* = \mathbf{c}$, and similarly for the other generators;
— $\mathbf{0}^* = \mathbf{0}$, and for all $u, v \in W$, $(u + v)^* = u^* + v^*$;
— $\mathbf{1}^* = \mathbf{1}$, and for all $u, v \in W$, $(uv)^* = v^*u^*$.

It is then straight-forward to check the following:

Lemma 4.4 (Reversal). Let μ be a net, p a straight path of μ, and p' the reversal of p. Then, $w(p') = w(p)^*$.

We are now ready to define the GoI interpretation of a cut-free net:

Definition 4.3 (GoI interpretation). Let M be an interaction monoid, let ν be a cut-free net with $n \geq 1$ free ports, and let P_{ji} be the set of straight paths of ν starting from the free port j and ending into the free port i. The GoI interpretation of ν in M is an endomorphism of M^n, which we represent as a formal $n \times n$ matrix $\nu^\bullet$, whose entries are defined as follows:

$$\nu^\bullet_{ij} = \sum_{p \in P_{ji}} w(p).$$

i and j range over the free ports of ν, and the sum is intended to be equal to $\mathbf{0}$ (the everywhere-zero endomorphism) if $P_{ji} = \emptyset$.

If A is a formal matrix with coefficients in W, we can define A^* as the transpose-involute matrix of A: $(A^*)_{ij} = (A_{ji})^*$. Then, the following clearly holds from Lemma 4.4 applied to Definition 4.3:

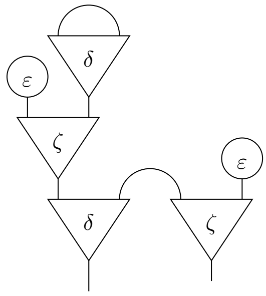
Proposition 4.5. If $\nu^\bullet$ is the GoI interpretation of a cut-free net ν, then

$$\nu^{\bullet *} = \nu^\bullet.$$

We can give a few examples to clarify the definition. If ε,δ, and ω are the three nets defined in Sect. 3.2, p. 16, we have $\varepsilon^\bullet = \mathbf{0}$, $\delta^\bullet = \mathbf{cd}^* + \mathbf{dc}^*$, while $\omega^\bullet$ is the endomorphism of $M \oplus M$ represented by the following matrix:

$$\omega^\bullet = \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} \end{bmatrix}.$$

A slightly more complicated example is the net

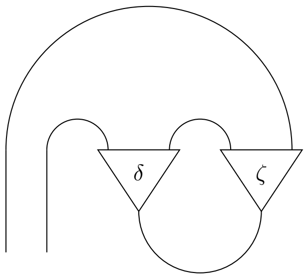


whose GoI interpretation is

$$\begin{bmatrix} \mathbf{cg}(\mathbf{cd}^* + \mathbf{dc}^*)\mathbf{g}^*\mathbf{c}^* & \mathbf{df}^* \\ \mathbf{fd}^* & \mathbf{0} \end{bmatrix}.$$

In all cases, the reader can check that Proposition 4.5 is verified.

The reader may wonder why we have restricted our interpretation to cut-free nets, in sharp contrast to Definition 3.4, where the denotational semantics is defined for *any* net. The reason is quite simple: in the absence of any restriction, Definition 4.3 would not make sense in general, since P_{ji} may contain an infinite number of non-zero-weighing paths. As a matter of fact, consider the following example:

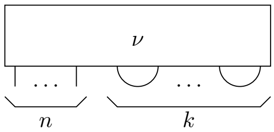


There is obviously an infinite number of straight paths from, for instance, the left free port to the right one; their weights are $\mathbf{c}^*(\mathbf{fd}^*)^n\mathbf{g}$, for every $n \in \mathbb{N}$. No element of W can be associated to the sum of all these paths, so the interpretation would be undefined. On the other hand, the following result assures us that Definition 4.3 is sound as we formulated it:

Proposition 4.6. Let ν be a cut-free net with at least one free port, and let i, j be two free ports (maybe the same) of ν. Then, P_{ji} is finite.

Proof. Remember the general decomposition of a cut-free net with $n \geq 1$ free ports: n trees $\tau_1, \ldots, \tau_n$ with an ε-wiring $\widetilde{\omega}$ "on top". Now, a straight path p from port j to port i is necessarily of the following shape: p goes up along one branch of τ_j from the root to its leaf, which is connected through a wire of $\widetilde{\omega}$ to a leaf k of τ_i; p follows this connection, and then goes down the branch of τ_i leading from k to its root. Therefore, the number of straight paths in P_{ji} is bounded by the number of leaves of τ_j, which is of course finite. $\square$

Any net with $n \geq 1$ free ports and k active pairs and/or vicious circles can be decomposed as follows:



where ν is cut-free and has $n+2k$ free ports. Notice that, because of the possible presence of vicious circles, ν is not unique in general. Nevertheless, given an interaction monoid M and a net μ with $n \geq 1$ free ports and k active pairs and/or vicious circles, we can associate to μ at least one endomorphism $\mu^\bullet$ of M^{n+2k}, which is the GoI interpretation of the net ν in one of the possible decompositions; the association will be unique exactly when μ does not contain vicious circles.

We also consider the endomorphism $\sigma_{n,k}$ of M^{n+2k} defined by the formal matrix

$$\sigma_{n,k} = \begin{bmatrix} \mathbf{0} & & & & & & \\ & \ddots & & & & & \\ & & \mathbf{0} & & & & \\ & & & \mathbf{0} & \mathbf{1} & & \\ & & & \mathbf{1} & \mathbf{0} & & \\ & & & & & \ddots & \\ & & & & & & \mathbf{0} & \mathbf{1} \\ & & & & & & \mathbf{1} & \mathbf{0} \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ b \\ c \end{matrix}}\right\} n \\ \left.\vphantom{\begin{matrix} a \\ b \\ c \\ d \\ e \end{matrix}}\right\} 2k \end{matrix}$$

(the entries not specified are $\mathbf{0}$) and the monomorphism $\pi_{n,k}$, which is the inclusion of M^n into M^{n+2k}:

$$\pi_{n,k} = \begin{bmatrix} \mathbf{1} & & \\ & \ddots & \\ & & \mathbf{1} \\ \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ b \\ c \end{matrix}}\right\} n \\ \left.\vphantom{\begin{matrix} a \\ b \\ c \end{matrix}}\right\} 2k \end{matrix}$$

We can now state the fundamental theorem of the geometry of interaction semantics:

Theorem 4.7 (Lafont (Lafont, 1997)). Let μ be a total net with $n \geq 1$ free ports and k active pairs, and let $\sigma = \sigma_{n,k}$ and $\pi = \pi_{n,k}$. Then, $\sigma\mu^\bullet$ is nilpotent, and the GoI interpretation of the cut-free form of μ is given by Girard's *execution formula*

$$\mathrm{Ex}(\mu^\bullet, \sigma) = \pi^t \left(\sum_{i=0}^{\infty} \mu^\bullet (\sigma\mu^\bullet)^i \right) \pi,$$

where π^t is the transpose of π.

The result above comes from the fact that the execution formula is an invariant of reduction. Therefore, if $\mu^\bullet$ is associated to a total net μ, then $\mathrm{Ex}(\mu^\bullet, \sigma)$ can be seen as a semantics for μ. In particular, if μ is cut-free, $\sigma = \mathbf{0}$ and $\mathrm{Ex}(\mu^\bullet, \mathbf{0}) = \mu^\bullet$.

4.3. *Relationship between denotational semantics and GoI*

We shall now see that the two semantics defined in the previous sections are strongly related to each other. We have already seen (Corollary 4.3) that, given a net μ with $n \geq 1$ free ports, the denotational semantics $[\![\mu]\!]$ of μ in an interaction monoid M is a submonoid of M^n. In case μ is total, then $[\![\mu]\!]$ is the submonoid of the fixpoints of $\mathrm{Ex}(\mu^\bullet, \sigma)$.

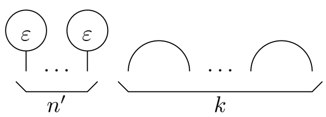
We prove the result stated above for cut-free nets only; by the preservation of both $[\![\cdot]\!]$ and $\mathrm{Ex}(\cdot, \cdot)$ under reduction, this is enough for the result to hold in the more general case of total nets. In the following, we write $\mathrm{fix}(u)$ for the set of fixpoints of a function u.

Theorem 4.8. Let ν be a cut-free net with $n \geq 1$ free ports. Then

$$[\![\nu]\!] = \mathrm{fix}(\nu^\bullet),$$

where the interpretations are taken in any interaction monoid M.

Proof. We prove both inclusions by induction on the number m of binary cells of ν. If $m = 0$, then ν is an ε-wiring, containing n' ε cells and k wires, with $n' + 2k = n$. We can assume without loss of generality that ν has the following shape

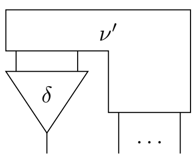


hence $\nu^\bullet = \sigma_{n',k}$. Now, if $x \in [\![\nu]\!]$, then

$$x = (\underbrace{0, \ldots, 0}_{n'}, x_1, x_1, \ldots, x_k, x_k),$$

so that we obviously have $\nu^\bullet(x) = x$. Conversely, it is trivial to check that every fixpoint of $\sigma_{n',k}$ is of this form.

Now let $m > 0$; then, since ν is cut-free, at least one of its free ports is the principal port of a binary cell. Suppose it is a δ cell; we can assume without loss of generality that ν has the following shape



where ν' is also cut-free. Now, if we order from left to right the free ports of ν and ν', we have that their GoI interpretations are given by the following formal matrices:

$$\nu'^{\bullet} = \left[\begin{array}{cc|c} h & u^* & A^* \\ u & k & B^* \\ \hline A & B & C \end{array}\right]$$

$$\nu^{\bullet} = \left[\begin{array}{c|c} \mathbf{c}h\mathbf{c}^* + \mathbf{d}u^*\mathbf{c}^* + \mathbf{c}u\mathbf{d}^* + \mathbf{d}k\mathbf{d}^* & \mathbf{c}A^* + \mathbf{d}B^* \\ \hline A\mathbf{c}^* + B\mathbf{d}^* & C \end{array}\right].$$

We have used the block notation to represent an arbitrary number (even zero) of entries in the bottom-right part of the matrices, corresponding to the $n-1$ ports that are free both in ν and ν'. Here, h, u, and k are endomorphisms of W, with $h^* = h$ and $k^* = k$, while A, B and C are resp. $(n-1) \times 1$ and $(n-1) \times (n-1)$ matrices with entries in W, with $C^* = C$.

Let us now take $x \in [\![\nu]\!]$. We know that x is an element of M^n, so $x = (y, z)$, where $y \in M$ and $z \in M^{n-1}$. Moreover, since y is associated to the free ports of a δ cell, we have $y = \mathbf{c}(y') + \mathbf{d}(y'')$ for some $y', y'' \in M$ such that $x' = (y', y'', z) \in [\![\nu']\!]$. If we apply $\nu'^{\bullet}$ to x', we get

$$\left[\begin{array}{cc|c} h & u^* & A^* \\ u & k & B^* \\ \hline A & B & C \end{array}\right] \cdot \left[\begin{array}{c} y' \\ y'' \\ \hline z \end{array}\right] = \left[\begin{array}{c} h(y') + u^*(y'') + A^*(z) \\ u(y') + k(y'') + B^*(z) \\ \hline A(y') + B(y'') + C(z) \end{array}\right].$$

But by induction hypothesis, $\nu'^{\bullet}(x') = x'$, so the following equalities hold:

$$\begin{array}{rcl} h(y') + u^*(y'') + A^*(z) & = & y' \\ u(y') + k(y'') + B^*(z) & = & y'' \\ A(y') + B(y'') + C(z) & = & z\ . \end{array}$$

From this, if we compute $\nu^{\bullet}(x)$, we get

$$\left[\begin{array}{c|c} \mathbf{c}h\mathbf{c}^* + \mathbf{d}u^*\mathbf{c}^* + \mathbf{c}u\mathbf{d}^* + \mathbf{d}k\mathbf{d}^* & \mathbf{c}A^* + \mathbf{d}B^* \\ \hline A\mathbf{c}^* + B\mathbf{d}^* & C \end{array}\right] \cdot \left[\begin{array}{c} \mathbf{c}(y') + \mathbf{d}(y'') \\ \hline z \end{array}\right] =$$

$$= \left[\begin{array}{c} \mathbf{c}(h(y') + u^*(y'') + A^*(z)) + \mathbf{d}(u(y') + k(y'') + B^*(z)) \\ \hline A(y') + B(y'') + C(z) \end{array}\right] = x.$$

Consider now a fixpoint x of $\nu^{\bullet}$. Again, this is an element of M^n, and can be decomposed into (y, z) with $y \in M$ and $z \in M^{n-1}$. Now, by surjectivity there exist $y', y'' \in M$ such that $y = \mathbf{c}(y') + \mathbf{d}(y'')$, so we can define $x' = (y', y'', z)$, for which the same computations done above show that $\nu'^{\bullet}(x') = x'$. By induction hypothesis, $x' \in [\![\nu']\!]$, which means that there is an experiment of ν' with result (y', y'', z); the "same" experiment then gives (y, z) on ν, which proves that $x \in [\![\nu]\!]$.

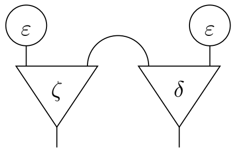
The proof in the case of a ζ combinator is identical: we just need to replace $\mathbf{c}, \mathbf{d}$ with $\mathbf{f}, \mathbf{g}$. $\square$

Theorem 4.8 tells us that, for any cut-free net ν with at least one free port, if we know $\nu^\bullet$, we also know $[\![\nu]\!]$. Is the converse true? The rest of the section is devoted to prove that it is actually the case.

Let M be an interaction monoid, and let ν be a cut-free net with $n \geq 1$ free ports. By Theorem 3.7, we know that the elements of $[\![\nu]\!]$ are described by a balanced bracket expression B. From this expression, it is not hard to build an endomorphism ϕ of M^n such that $\mathrm{fix}(\phi) = [\![\nu]\!]$:

- Suppose that $\mathsf{var}(\mathsf{B}) = \{x_1, \ldots, x_m\}$. We make a new expression B' (no longer balanced) which is structurally identical to B, but such that, for each variable x_i of B, one occurrence of x_i is replaced by x_i' and one by x_i'', with x_i', x_i'' distinct and fresh, and each occurrence of $\mathsf{0}$ is replaced by a distinct fresh variable z_j; this is always possible since B is balanced. Notice that B' contains every variable at most once.
- From the fact that $\langle \cdot, \cdot \rangle$ and $[\cdot, \cdot]$ are bijections, we know that, for each $d \in M^n$, there exist unique $d_1', d_1'', \ldots, d_m', d_m'', e_1, \ldots, e_k \in M$ such that $d = \mathsf{B}'\{\ldots\ x_i' := d_i', x_i'' := d_i''\ \ldots\ z_j := e_j\ \ldots\}$ (see Sect. 3.3, p. 22). We then define ϕ as follows:
$$\phi(d) = \mathsf{B}'\{\ldots\ x_i' := d_i'', x_i'' := d_i'\ \ldots\ z_j := 0\ \ldots\},$$
i.e., we "swap" the elements assigned to x_i' and x_i'', and we set each z_j to 0.
- Clearly, $\phi(0) = 0$, and because $\langle \cdot, \cdot \rangle, [\cdot, \cdot]$ are isomorphisms, we also have $\phi(x + y) = \phi(x) + \phi(y)$, so ϕ is indeed an endomorphism of M^n (not an isomorphism though, since in general some non-zero elements may be mapped to zero). It is not hard to check that ϕ verifies $\phi^3 = \phi$, i.e., it is a *partial symmetry.*
- By construction, the fixpoints of ϕ are those elements described by B, so $\mathrm{fix}(\phi) = \mathrm{Im}(\mathsf{B})$.

Let us look at an example to clarify the construction above. The balanced expression $([\mathsf{0}, x], \langle x, \mathsf{0}\rangle)$, which generates the interpretation of



is turned into $([z_1, x'], \langle x'', z_2\rangle)$. Now, for each element $x \in M \oplus M$, there exist unique $z_1, x', x'', z_2 \in M$ such that $x = ([z_1, x'], \langle x'', z_2\rangle)$ (we have used the same notations for the variables in the expression and the elements of M to avoid writing the substitution explicitly). We then define ϕ so that

$$\phi(x) = \phi([z_1, x'], \langle x'', z_2\rangle) = ([0, x''], \langle x', 0\rangle).$$

The zero of $M \oplus M$ is $(0, 0) = ([0, 0], \langle 0, 0\rangle)$, hence $\phi(0, 0) = (0, 0)$, and if we take $x, y \in M \oplus M$, we decompose them as $x = ([x_1, x_2], \langle x_3, x_4\rangle)$ and $y = ([y_1, y_2], \langle y_3, y_4\rangle)$, and we have $x + y = ([x_1 + y_1, x_2 + y_2], \langle x_3 + y_3, x_4 + y_4\rangle)$, from which we obtain

$$\begin{aligned}\phi(x + y) &= ([0, x_3 + y_3], \langle x_2 + y_2, 0\rangle) = \\ &= ([0, x_3], \langle x_2, 0\rangle) + ([0, y_3], \langle y_2, 0\rangle) = \phi(x) + \phi(y),\end{aligned}$$

so ϕ is an endomorphism of $M \oplus M$. We also have

$$\mathrm{fix}(\phi) = \{([0, x], \langle x, 0\rangle) \in M \oplus M \ ; \ x \in M\},$$

which is exactly the interpretation of the above net.

All that is left to do is verifying that $\phi = \nu^\bullet$. This is proved by induction on the number m of binary cells in ν. If $m = 0$, ν is an ε-wiring, and its interpretation can be assumed without loss of generality to be generated by a balanced expression of the form

$$(\mathsf{0}, \ldots, \mathsf{0}, x_1, x_1, \ldots, x_k, x_k),$$

where the symbol $\mathsf{0}$ appears n' times, with $n' + 2k = n$ (the number of free ports of ν). Then, ϕ is the endomorphism such that

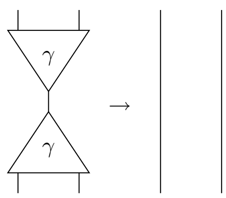
$$\phi(x_1, \ldots, x_{n'}, y'_1, y''_1, \ldots, y'_k, y''_k) = (0, \ldots, 0, y''_1, y'_1, \ldots, y''_k, y'_k),$$

which is exactly the endomorphism we introduced at p. 29 under the name $\sigma_{n',k}$, and which is equal to $\nu^\bullet$. If $m > 0$, calculations virtually identical to those of the proof of Theorem 4.8 show that we have $\phi = \nu^\bullet$ in this case as well; the details are left to the reader.

5. Conclusions

Why the symmetric interaction combinators? The reader may wonder why we have chosen to work with the *symmetric* combinators instead of the "standard" ones, which enjoy a stronger universality property (Theorem 2.2 holds for *any* interaction net system, not just polarized ones). The answer is technical: there is a detail in the reduction rules of the interaction combinators which renders impossible the formulation of a relational semantics like the one considered here.

We remind that the "standard" interaction combinators are defined exactly as the symmetric ones, except that instead of ζ there is a binary cell γ, which interacts with itself as follows:

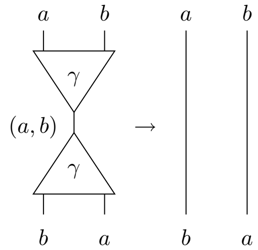


All other interaction rules are unchanged. Notice that the above rule "exchanges" the auxiliary ports of the γ cell: according to our convention (p. 3), the left port of each occurrence of γ is connected to the right port of the other occurrence. On the contrary, the $\delta\delta$ (and $\zeta\zeta$) annihilation connects left with left and right with right.

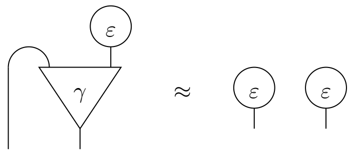
Now, in a relational semantics, reduction is modeled by composition of relations: if the "rightmost" free port of a net μ is connected to the "leftmost" free port of a net μ', the denotational semantics of the resulting net will be (see the proof of Lemma 3.3)

$$[\![\mu]\!] \circ [\![\mu']\!] = \{(a,c)\ ;\ (a,b) \in [\![\mu]\!],\ (b,c) \in [\![\mu']\!]\}.$$

This is ensured by our definition of experiment. But if we try to define experiments in the presence of the $\gamma\gamma$ annihilation, we see that the only way for the interpretation to model reduction is that both of the auxiliary ports of γ cells receive the same value. In fact, in the rule



we clearly need $a = b$. This is an unreasonable restriction; for example, it would imply that the following two nets receive the same semantical interpretation:



These two nets are not $\beta\eta$-equivalent; from Proposition 2.6, we infer that in such a situation, if we ever managed to model $\simeq_\beta$, we would do so by identifying all total nets with a non-empty interface.

The argument given here of course does not rule out the possibility of finding a denotational semantics for the interaction combinators; it simply shows that the standard definitions do not work, and justifies our shift towards the symmetric combinators.

Further work. Our efforts are now concentrating on a *typed* semantics for the symmetric combinators. This would not only yield a typing discipline for the combinators, which would ensure good properties like deadlock-freeness and termination, but more interestingly a new logical system, which should be an extension of multiplicative linear logic.

The existence of such a system would be very intriguing, since it would combine the simplicity of the multiplicative fragment of linear logic with the high expressive power of the symmetric combinators.

References

Danos, V. and Regnier, L. (1989). The structure of multiplicatives. *Archive for Mathematical Logic*, 28:181–203.

Ehrhard, T. (2005). Finiteness spaces. *Mathematical Structures in Computer Science*, 15(4):615–646.

Fernández, M. and Mackie, I. (2003). Operational equivalence for interaction nets. *Theoretical Computer Science*, 297(1–3):157–181.

Girard, J.-Y. (1987a). Linear logic. *Theoretical Computer Science*, 50(1):1–102.

Girard, J.-Y. (1987b). Multiplicatives. In Lolli, G., editor, *Logic and Computer Science: New Trends and Applications*, pages 11–34. Rendiconti del Seminario Matematico dell'Università e Politecnico di Torino.

Girard, J.-Y. (1989). Geometry of interaction I: interpretation of System F. In *Proceedings of the Logic Colloquium '88*, pages 221–260. Noth Holland.

Gonthier, G., Abadi, M., and Lévy, J.-J. (1992). The geometry of optimal lambda reduction. In *Conference Record of the Nineteenth ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 92)*, pages 15–26. ACM Press.

Lafont, Y. (1990). Interaction nets. In *Conference Record of POPL'90*, pages 95–108. ACM SIGACT and SIGPLAN, ACM Press.

Lafont, Y. (1995). From proof nets to interaction nets. In Girard, J.-Y., Lafont, Y., and Regnier, L., editors, *Advances in Linear Logic*, pages 225–247. Cambridge University Press.

Lafont, Y. (1997). Interaction combinators. *Information and Computation*, 137(1):69–101.

Lippi, S. (2002). Encoding left reduction in the lambda-calculus with interaction nets. *Mathematical Structures in Computer Science*, 12(6):797–822.

Mackie, I. and Pinto, J. S. (2002). Encoding linear logic with interaction combinators. *Information and Computation*, 176(2):153–186.

Mazza, D. (2006). Observational equivalence for the interaction combinators and internal separation. In Mackie, I., editor, *Proceedings of TERMGRAPH 2006*, Electronic Notes in Theoretical Computer Science, pages 7–16. Elsevier.

Pagani, M. (2006). Proofs, denotational semantics, and observational equivalence in multiplicative linear logic. To appear in *Mathematical Structures in Computer Science*.