Observational Equivalence for the Interaction Combinators and Internal Separation

(Extended Abstract)

Damiano Mazza [1]

Institut de Mathématiques de Luminy
Marseille, France

Abstract

We define an observational equivalence for Lafont's interaction combinators, which we prove to be the least discriminating non-trivial congruence on *total nets* (nets admitting a deadlock-free normal form) respecting reduction. More interestingly, this equivalence enjoys an internal separation property similar to that of Böhm's Theorem for the λ-calculus.

Key words: Interaction nets, interaction combinators, observational equivalence, internal separation, Böhm's Theorem.

1 Introduction

Lafont's interaction combinators [4] are a graph-rewriting model of deterministic distributed computation. As in Turing machines, transitions are local, but they can be performed in parallel; the determinism comes from the fact that the resulting computation is unique up to permutation of rewriting steps. They can be seen as a generalization of multiplicative linear logic *proof-nets* [1], and in fact fit into the wider framework of *interaction nets* [3].

The original motivation behind the introduction of the interaction combinators was indeed the definition of a simple *universal interaction net system*, i.e., a system capable of simulating all other interaction net systems. The universality of the combinators, which automatically entails their Turing-completeness, together with their extreme simplicity and elegance, are in our opinion enough to justify the interest of studying this system as an autonomous computational model, ignoring its relationship to general interaction nets.

In this paper, we define a notion of observational equivalence for normal nets of combinators, which is interesting because it is maximal (Proposi-

[1] Email: mazza@iml.univ-mrs.fr

This paper is electronically published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

tion 3.4) and because it can be characterized by means of a rewriting relation analogous to η-equivalence in the λ-calculus. In other words, we prove an internal separation result similar to Böhm's Theorem (Theorem 3.7): given two non-equivalent normal nets, there exists a context separating them.

It is important to observe however that "separating" does not have exactly the same sense as in Böhm's classical result: in the λ-calculus, two distinct $\beta\eta$-normal forms can be separated by sending them to *any* pair of distinct $\beta\eta$-normal forms (the typical choice being the projections $\lambda xy.x$ and $\lambda xy.y$); in interaction combinators, the uninformative behavior of the ε combinator forces it to be one the separation values, as no context can extract any information from it. Therefore, we actually obtain something more akin to Hyland's Theorem (sometimes referred to as "semi-separation"), which extends Böhm's result to non-normal terms. This reveals a sharp difference between interaction combinators and the λ-calculus, as "full" separation already fails for normal nets.

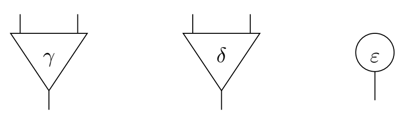
Non-normal (or rather non-normalizable) nets will not be addressed in this extended abstract; indeed, even though an analysis of nets with possibly infinite behavior is of clear interest, we prefer to keep it as a future extension of the present work.

As to related work, we must mention Maribel Fernández and Ian Mackie's in-depth study on observational equivalence for interaction nets [2], in which one can find the η-like rules we give at page 6. The other equivalence rules we consider were already found by Lafont through a semantical analysis. Nevertheless, no special property about these equivalences had previously been proved (like maximality or internal separation).

2 Interaction Combinators

2.1 Combinators, nets, interaction rules

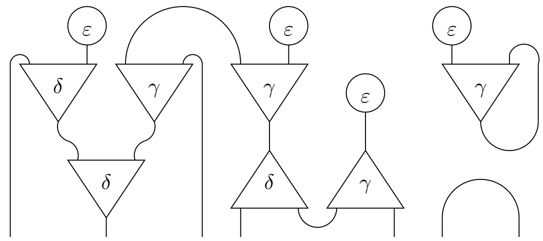
The interaction combinators are the three following *cells*:



Each cell has a number of *ports*; γ and δ 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*.

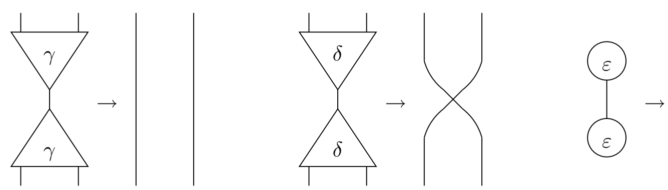
Ports may be used to plug cells[2] together by means of *wires* to form *nets*, as in the following example:

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

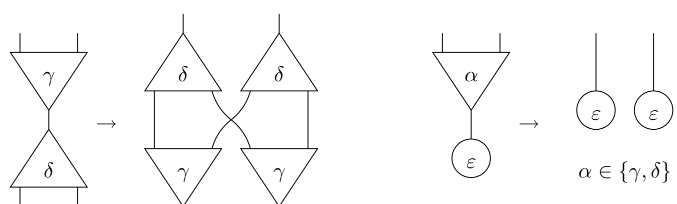


Wires can have one or both of their extremities not connected to any cell, in which case the net has a *free port*, principal or auxiliary (or neither) depending on the nature of the port of the cell connected to the other extremity of the wire. The net above has for example 7 free ports, of which 1 is principal and 4 are auxiliary. The free ports of a net are referred to as its *interface*.

The 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 following *interaction rules*: the *annihilations*



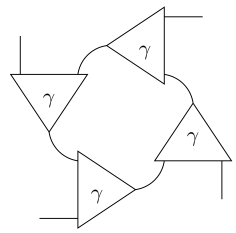
and the *commutations*



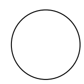
When a net μ' is obtained from μ after the application of one of the above rules, we say that μ *reduces in one step* to μ', and we write $\mu \to \mu'$. We can then define the *reduction relation* $\to^*$ on nets of combinators as the reflexive-transitive closure of $\to$. We write $\mu \simeq_\beta \mu'$ 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 that $\to$ is confluent, which implies that the reduction process is also (strongly) confluent. This ensures that the computation is unique up to permutation of rules, and that $\simeq_\beta$ is an equivalence relation. We remark here a substantial difference with respect to reduction in the λ-calculus: if a net is normalizable, then it is strongly so.

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*:

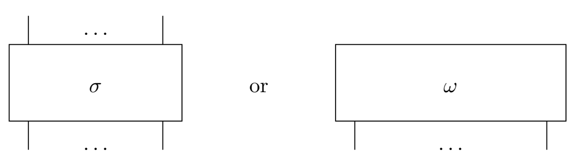
Definition 2.1 (Vicious circle) A *straight path* in a net is a simple path (i.e., not using any wire more than once) which never crosses consecutively two auxiliary ports of the same cell. A *vicious circle* is a cyclic straight path.

A net containing no active pair and no vicious circle is said to be *cut-free*.[3] A net admitting a cut-free form (necessarily unique by confluence) is said to be *total*; on the contrary, a net whose reduction goes on forever, or which reduces to a net containing a vicious circle will be called *partial*.

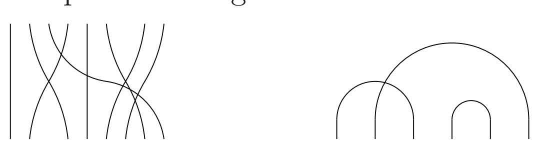
Cut-free nets are the "true" normal forms of the reduction; they can be seen as the final result of a computation. On the other hand, partial nets represent deadlocked or divergent computations.

2.2 Basic nets

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

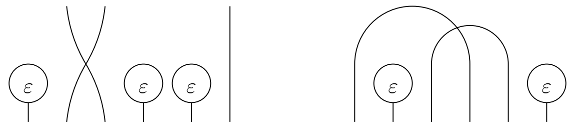


The following are examples of wirings:



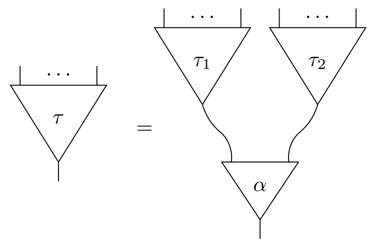
We also allow the free ports of a wiring to belong to ε cells, in which case we speak of an ε-wiring and we use the notation $\widetilde{\sigma}$ or $\widetilde{\omega}$. The following are examples of ε-wirings:

[3] Lafont uses the term *reduced*; we prefer cut-free on the grounds that it does not conflict with standard rewriting terminology and that it makes sense (even though there is no room for explanations here) to say that vicious circles actually hide irreducible logical cuts.



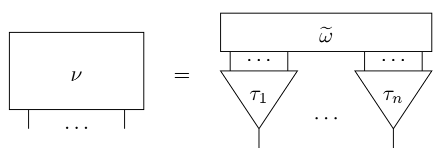
Trees. Trees are defined inductively as follows. The wiring

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 $\alpha \in \{\gamma, \delta\}$.

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 and 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 particular examples principal nets.

3 Observational equivalence

The system of interaction combinators can be seen as an abstract programming language, in which total nets are deadlock-free and terminating programs/data. As such, it may be of interest to define on it a notion of *observational* equivalence. In this paper, we shall restrict our attention to total nets with one free port, although everything we shall say can be generalized to total nets with arbitrary interfaces. Therefore, unless otherwise stated, in the sequel "net" will mean "net with one free port".

Let μ be a total net, and θ a test. If we plug the free port of μ to the principal free port of θ (or to any of its free ports in case θ is a wire), we obtain a net μ'. If μ' is total[4], and its cut-free form is π, we write $\theta[\mu] = \pi$, otherwise we write $\theta[\mu] = \Omega$.

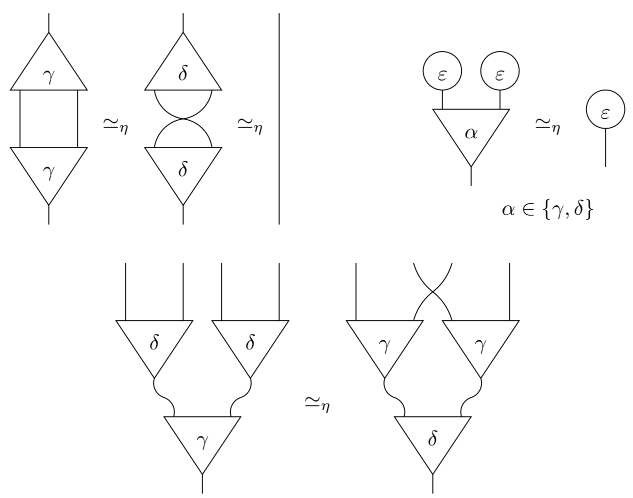
Definition 3.1 (Finite test) Let μ, μ' be two total nets. A test θ is said to be (μ, μ')-finite iff $\theta[\mu] \neq \Omega$ and $\theta[\mu'] \neq \Omega$.

Definition 3.2 (ε-package) An ε-package is a tree whose leaves are all connected to principal ports of ε cells.

Definition 3.3 (Observational equivalence) Let μ, μ' be two total nets. We say that μ is *observationally equivalent* to μ', notation $\mu \simeq^\circ \mu'$, iff for any (μ, μ')-finite test θ, $\theta[\mu]$ is an ε-package iff $\theta[\mu']$ is.

Proposition 3.4 (Maximality) *$\simeq^\circ$ is the greatest non-trivial congruence on total nets containing $\simeq_\beta$, i.e., if $\approx$ is a congruence such that $\simeq_\beta \subseteq \approx$, then either $\approx \subseteq \simeq^\circ$, or $\mu \approx \mu'$ for all μ, μ'.*

The above proposition can be proved independently of what we shall do in the sequel, but the proof is rather tedious, so we prefer to state it as a corollary of Theorem 3.7, Sect. 3.2. As a matter of fact, we shall prove that $\simeq^\circ$ can actually be defined in a much more concrete way, indeed as an equivalence relation generated by $\simeq_\beta$ plus the following equalities:



The top-right and bottom equations were already considered by Lafont [4]; in particular, the top-right rule states the equivalence of all ε-packages to the ε combinator. On the other hand, the top-left equations can be found in the work of Fernández and Mackie as part of a larger study on operational equivalence for interaction nets [2].

Definition 3.5 (Equivalence) Two nets μ, μ' are *η-equivalent*, notation $\mu \simeq_\eta \mu'$, iff they can be rewritten one into the other by means of the above equalities. We write $\simeq$ for the transitive closure of $\simeq_\beta \cup \simeq_\eta$, and if $\mu \simeq \mu'$, we say that μ and μ' are *equivalent*.

[4] Totality is obviously semi-decidable; this will not be a concern for us here.

We point out that there is no reasonable orientation for the equations defining η-equivalence, so there are no *canonical representatives* for the equivalence classes of $\simeq$ (as opposed to $\beta\eta$-normal forms in the λ-calculus).

3.1 Adequacy

To prove that $\simeq \subseteq \simeq^\circ$, we need the following lemma, which tells us that η-equivalence is a weak bisimilarity (the proof is omitted):

Lemma 3.6 *Let μ_1, μ_2 be two η-equivalent nets such that $\mu_1 \to \mu_1'$. Then, there exists a net μ_2' such that $\mu_2 \to^* \mu_2'$ and $\mu_2' \simeq_\eta \mu_1'$.*

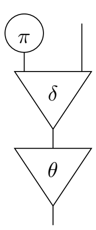
Now, if μ is a total net, ν its cut-free form, and θ a test, by confluence we have $\theta[\mu] = \theta[\nu]$, hence it is enough to consider cut-free nets. So let ν, ν' be two equivalent cut-free nets, and let $\widetilde{\nu}, \widetilde{\nu}'$ be the nets obtained by plugging them into a generic (ν, ν')-finite test θ. By hypothesis, $\widetilde{\nu} \simeq_\eta \widetilde{\nu}'$, and by Lemma 3.6 we conclude that $\theta[\nu] \simeq_\eta \theta[\nu']$. It is easy to see that the rules defining η-equivalence are such that a package π and an ε-package are η-equivalent iff π is itself an ε-package, so $\nu \simeq^\circ \nu'$.

3.2 Full-abstraction

In what follows, we write ε for the package consisting of the sole ε combinator, and δ for the package consisting of a δ combinator whose two auxiliary ports are connected by a wire.

Theorem 3.7 (Separation) *Let μ, μ' be two total nets such that $\mu \not\simeq \mu'$. Then, there exists a test θ such that $\theta[\mu] = \varepsilon$ and $\theta[\mu'] = \delta$,* or vice versa*.*

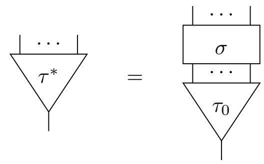
The above result proves in particular that $\simeq^\circ \subseteq \simeq$, and gives us a quick argument to prove Proposition 3.4. In fact, if $\approx$ is a congruence such that $\simeq_\beta \subseteq \approx$, and if $\mu \approx \mu'$ for two nets such that $\mu \not\simeq^\circ \mu'$, by adequacy and by Theorem 3.7 we have a test θ such that, for example, $\theta[\mu] = \varepsilon$ and $\theta[\mu'] = \delta$. Now put

$\theta_\pi =$ 

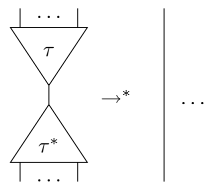
where π is any package. It is not hard to verify that $\theta_\pi[\mu] = \varepsilon$, while $\theta_\pi[\mu'] = \pi$. But $\approx$ is preserved through reduction, so $\mu \approx \mu'$ implies $\theta_\pi[\mu] \approx \theta_\pi[\mu']$, which means that $\varepsilon \approx \pi$, for all π.

We shall now sketch the proof of Theorem 3.7. First of all, as already observed in Sect. 3.1, it is enough to prove our result for packages, because we always consider total nets. We need three fundamental results, which we state in the sequel. All proofs are omitted.

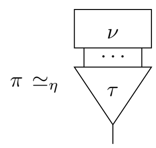
Lemma 3.8 (Anti-tree) *For any tree τ, there exists a net τ^*, called the* anti-tree *of τ, of the form*



such that



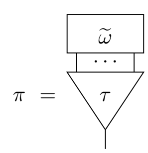
Lemma 3.9 (Equivalence lemma) *Let π be a package. Then, for any tree τ with n leaves, there exists a cut-free net ν with n free ports such that*



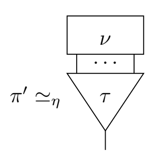
In the following, ε_2 is a net with two free ports consisting of two ε combinators, and we use the notation $(\theta_1, \theta_2)[\nu]$ to denote the result of the application of the tests θ_1, θ_2 to a net ν with 2 free ports.

Lemma 3.10 (Wire separation) *Let ω be a single wire, and ν a cut-free net with 2 free ports, such that $\nu \not\simeq \omega$. Then, there exist two tests θ_1, θ_2 such that $(\theta_1, \theta_2)[\omega] = \omega$ and $(\theta_1, \theta_2)[\nu] = \varepsilon_2$,* or viceversa.

Let now π, π' be two packages such that $\pi \not\simeq_\eta \pi'$. Suppose that



By the Equivalence Lemma 3.9, there exists a cut-free net ν such that



so it does not harm to assume that π and π' "end" with the same tree.

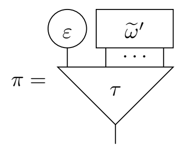
Now, at least one of the following two situations must apply:

(a) there exists a leaf of τ which is connected to an ε combinator of $\widetilde{\omega}$, but is connected to something not equivalent to ε in ν;

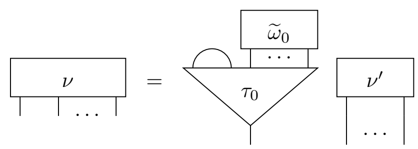
(b) there exist two leaves of τ which are connected by a wire of $\widetilde{\omega}$, whereas in ν the same two leaves are either not connected, or their connection is not equivalent to a wire.

As a matter of fact, if neither (a) nor (b) applied, we would have proved that $\pi \simeq_\eta \pi'$, against our hypothesis.

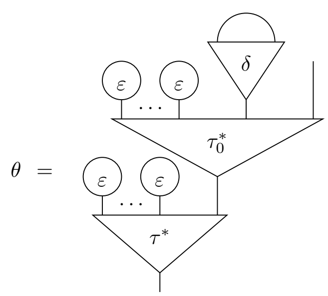
Suppose that situation (a) applies, and suppose w.l.o.g. that the leaf in question is the "leftmost" one, i.e., we have



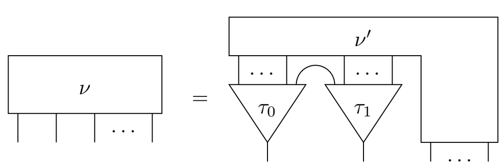
By hypothesis, the "leftmost" free port of ν, let us call it x, is connected to something not equivalent to ε; this means that if we "go up" the tree rooted at x in ν, let us call it τ_0, we must find a leaf of τ_0 connected by a wire to some other tree of ν. It may happen that all connections are within τ_0 itself, i.e., we have



(for graphical convenience, we have assumed w.l.o.g. that there is a direct connection between the "leftmost" two leaves of τ_0). Under such assumptions, using Lemma 3.8 one can verify that the test

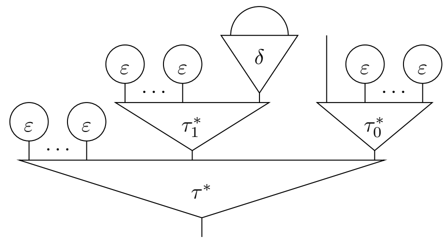


is such that $\theta[\pi'] = \delta$, whereas $\theta[\pi] = \varepsilon$. Suppose instead that τ_0 is connected to some other tree of ν, and suppose w.l.o.g. that this tree is the one immediately "to the right" of τ_0, let us call it τ_1:



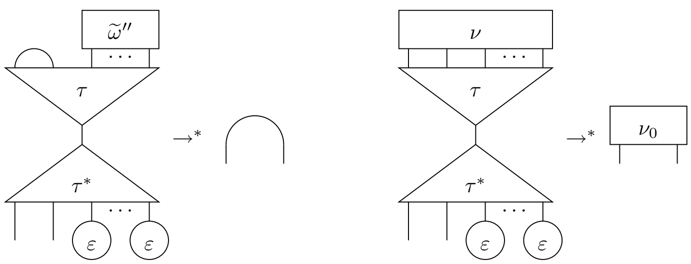
(again, in the picture we have made a convenient assumption about the connection between τ_0 and τ_1, without affecting the generality of our argument). Always using Lemma 3.8, and the fact that cut-free nets can be erased by ε cells, one may check that the test

$\theta =$



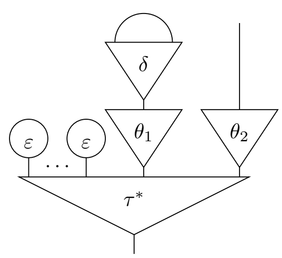
is such that $\theta[\pi'] = \delta$ and $\theta[\pi] = \varepsilon$.

Let us now consider situation (b), i.e., π has a direct connection for τ which π' is missing. Then, we can use the anti-tree τ^* and isolate the two leaves involved in the connection:



(as usual, for graphical purposes we have supposed w.l.o.g. that the two leaves in question are the "leftmost" ones). There is no room for the details here, but it is not hard to show that, under the hypotheses we have, ν_0 cannot be equivalent to a wire. Hence, the Wire Separation Lemma 3.10 applies, giving us two packages θ_1 and θ_2 which are able to distinguish between the wire and ν_0. Therefore, if we define

$\theta =$



we have $\theta[\pi] = \varepsilon$ and $\theta[\pi'] = \delta$, or viceversa, which completes the proof.

References

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

[2] Fernández, M. and I. Mackie, *Operational equivalence for interaction nets*, Theoretical Computer Science **297** (2003), pp. 157–181.

[3] Lafont, Y., *Interaction nets*, in: *Conference Record of POPL'90*, ACM SIGACT and SIGPLAN (1990), pp. 95–108.

[4] Lafont, Y., *Interaction combinators*, Information and Computation **137** (1997), pp. 69–101.