Universal Interaction Systems with Only Two Agents

Denis Bechet

LIPN - UPRES-A 7030
Institut Galilée
Université Paris 13 & CNRS
99, avenue J.-B. Clément
93430 VILLETANEUSE
France
Denis.Bechet@lipn.univ-paris13.fr

Abstract. In the framework of interaction nets [6], Yves Lafont has proved [8] that every interaction system can be simulated by a system composed of 3 symbols named γ, δ and ϵ. One may wonder if it is possible to find a similar universal system with less symbols. In this paper, we show a way to simulate every interaction system with a specific interaction system constituted of only 2 symbols. By transitivity, we prove that we can find a universal interaction system with only 2 agents. Moreover, we show how to find such a system where agents have no more than 3 auxiliary ports.
Keywords: proof net, interaction net, combinator, universal system.

1 Introduction

In [6], Yves Lafont introduces *interaction nets*, a programming paradigm inspired by Girard's proof nets for *linear logic* [3]. Some translations from λ-calculus into interaction nets [9, 4, 5] or from proof nets [7, 10, 2, 1, 11] show that universal interaction systems are interesting for computation. We can explain this interest for these translations by the fact that computation with interaction nets is purely local and naturally confluent. Reductions can be made in parallel. Moreover, the number of steps that are necessary to reduce completely a net is independent of the way one may choose. From the point of view of λ-calculus, translations used in [4, 5] captures optimal reduction.

In [8], Lafont introduces a universal interaction system with only three different symbols γ, δ and ϵ. δ and ϵ are respectively a duplicator and an eraser and γ is a constructor. This system preserves the complexity of computation for a particular system. The number of steps that are necessary to reduce a simulated interaction net is just (at most) multiplied by a constant (which depends only on the simulated system and not on the size of the simulated net).

One may wonder if it is possible to find a simpler universal interaction system with only 2 symbols. This paper answers yes to this question. In fact, we prove that we can simulate a particular interaction system with only two symbols. By simulating a universal system, we prove that a universal system constituted of only two symbols exists.

Using the universal system in [8], the resulting universal system has two symbols, one is an eraser and the second is a constructor/rule encoder. The eraser has no auxiliary port but the second one has 16 auxiliary ports! In fact, we show a way to reduce this number to only three auxiliary ports.

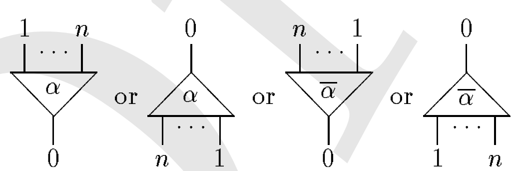
This paper is organized as follows: after an introduction to interaction nets and interaction systems, the notions of translations, simulations and universal interaction systems are presented. Section 4 is the heart of this article. It shows how to reduce a system to a system with only two agents. Section 5 reduces the number of auxiliary ports of the agents in this system to 0 and 3.

2 Interaction system

This model of computing is introduced in [6]. We briefly recall what interaction nets and interaction systems are.

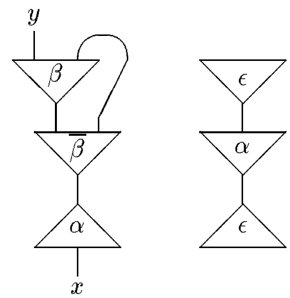
2.1 Agents and nets

An interaction net is a set of agents linked together through their ports. An individual agent is an instance of a particular symbol which is characterized by its *name* α and its *arity* $n \geq 0$. The arity defines the number of auxiliary ports associated to each agent. In addition to auxiliary ports, an agent owns a *principal port*. Graphically, an agent is represented by a triangle :



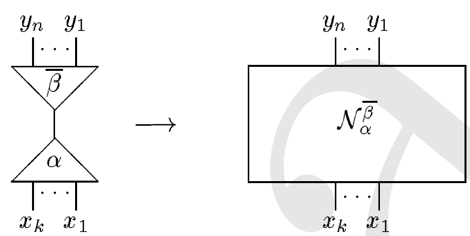
With α, auxiliary ports go clockwise from 1 to n but with $\overline{\alpha}$ it goes in the other direction (an agent is obtained by symmetry from α to $\overline{\alpha}$). Here, the principal port is noted 0, the auxiliary ports $1 \ldots n$.

An interaction net is a set of agents where the ports are connected two by two. The ports that are not connected to another one are the *free ports* of the net and are distinguished by a unique symbol. The set of the symbols of the free ports of a net consists the *interface* of this net. Below, the interface is $\{y, x\}$. α has one auxiliary port, β has two and ϵ has none.



2.2 Interaction rule and interaction system

An interaction net can evolve when two agents are connected through their principal port. An *interaction rule* is a rewriting rule where the left member is constituted of only two agents connected through their principal ports and the right member is any interaction net with the same interface.



An interaction net that does not contain two agents connected by their principal port is *irreducible* (we say also *reduced*). A net is reduced by applying an interaction rule to a couple of agents connected through their principal port. This step substitutes the couple by the right member of the rule. A reduction can be repeated several times.

An *interaction system* $\mathcal{I} = (\Sigma, \mathcal{R})$ is a set of symbols Σ and a set of interaction rules $\mathcal{R}$ where agents in the left and right members are instances of the symbols of Σ.

An interaction system $\mathcal{I}$ is *deterministic* when (1) there exists at most one interaction rule for each couple of different agent and (2) there exists at most one interaction rule for the interaction of an agent with itself. In this case, the right member of this rule must be symmetric from a center point. An interaction system $\mathcal{I}$ is *complete* when there is at least one rule for each couple of agent. In this a paper we consider deterministic and complete systems. With these systems, we can prove that reduction is strongly confluent. In fact, this property is true whenever the system is deterministic. Moreover, it is assumed that right member of every rule has no *deadlock* and do not introduce an infinite recursive computation or a computation that creates a deadlock. Thus, we can always erase every part of the right member of a rule with eraser agents noted ϵ.

3 Universal interaction systems

Universality means that every interaction system can be *simulated* by a universal interaction system. Here, we use a very simple notion of simulation that is based on *translation*.

3.1 Translation

Let Σ and Σ' be two sets of symbols. A *translation* Φ from Σ to Σ' is a map that associates to each symbol in Σ an interaction net of agents of Σ' with the same interface. This translation is naturally extended to interaction nets of agents of Σ.

3.2 Simulation

We say that a translation Φ from Σ to Σ' defines a *simulation* of an interaction system $\mathcal{I} = (\Sigma, \mathcal{R})$ by an interaction system $\mathcal{I}' = (\Sigma', \mathcal{R}')$ if the reduction mechanism on interaction nets of $\mathcal{I}$ and $\mathcal{I}'$ are *compatible*. More precisely, that means that, if $\mathcal{N}$ is an interaction net of $\mathcal{I}$ then:

1. $\mathcal{N}$ is irreducible if and only if $\Phi(\mathcal{N})$ is irreducible;
2. if $\mathcal{N}$ reduces to $\mathcal{M}$ then $\Phi(\mathcal{N})$ can be reduced to $\Phi(\mathcal{M})$.

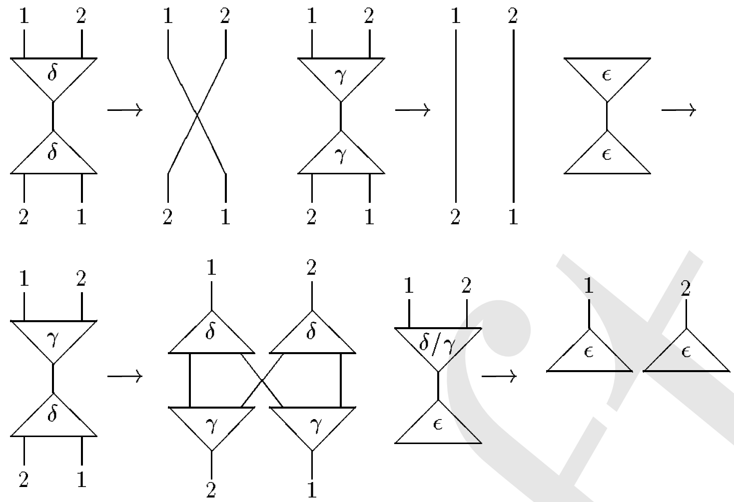
This definition brings some properties with complete and deterministic interaction systems:

- the interaction net corresponding to an agent must be reduced;
- it has at most one agent which principal port belongs to the interface and the symbol of this interface is the same as the symbol of the principal port of the initial agent;
- a translation is a simulation if and only if each rule of $\mathcal{R}$ is compatible with Φ;
- the simulation relation is transitive and symmetric.

In this paper, we have an approach that is not exactly the same as in [8]. Here, we work only with complete interaction systems but right members of interaction rules need not be reduced. They just need to be erasable by ϵ agents. However, it has a very small influence on the properties studied here.

3.3 Universal interaction system

An interaction system $\mathcal{U}$ is said to be *universal* if for any interaction system $\mathcal{I}$, there exists a simulation $\Phi^{\mathcal{I}}$ of $\mathcal{I}$ by $\mathcal{U}$. In [8], Lafont introduces a system of 3 combinators γ, δ and ϵ defined by 6 rules and he proves that this system is universal:



4 Universal system with only 2 agents

In this section, we show how to simulate a particular interaction system $\mathcal{I}$ with an interaction system composed of only two symbols ϵ and $\Pi_{\mathcal{I}}$. This system has exactly 3 rules.

ϵ agents have no auxiliary port. They erase everything even another ϵ agent (you can see for instance the above ϵ-rules). The other symbol $\Pi_{\mathcal{I}}$ has more auxiliary ports depending on the complexity of the system which is simulating. The rule between ϵ and itself gives the empty net. The rule between ϵ and $\Pi_{\mathcal{I}}$ gives m ϵ agents where m is the number of auxiliary ports of $\Pi_{\mathcal{I}}$. Finally, the rule between $\Pi_{\mathcal{I}}$ and itself is complex and depends of $\mathcal{I}$. This rule must be symmetric because it is a rule between an agent and itself.

4.1 Normalizing the number of auxiliary ports in $\mathcal{I}$

Before giving directly the simulation of a system $\mathcal{I}$, we simulate it by a system $\mathcal{I}'$ where all the agents have the same number of auxiliary ports except one ϵ that has a null arity.

For $\mathcal{I} = (\Sigma, \mathcal{R})$, let $n \geq 0$ be the maximum arity of the symbols. If $n = 0$, we set $n = 1$. We define $\Sigma' = \{(\alpha', n), (\alpha, i) \in \Sigma\} \cup \{(\epsilon, 0)\}$. ϵ has a null arity and erases everything that it meets. The other ones have the same arity n.

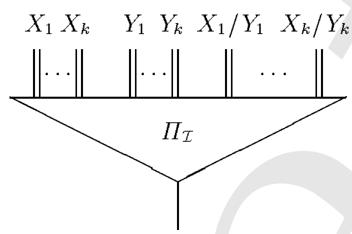
We translate an agent α of arity i in $\mathcal{I}$ by an agent α' where the $n - i$ ports number $i + 1, \ldots, n$ are connected to $n - i$ agents ϵ. The rule between agents α' and β' is derived from the rule between α and β by substituting in the right member each agent by its translation and by adding ϵ agents to the symbols in the interface that do not correspond to the interface of the rule between α and β. If the arity of α is i and the arity of β is j, there are $(n - i) + (n - j)$ ϵ agents added to complete the interface of the rule between α' and β'

A short proof shows that $\mathcal{I}$ is simulated by $\mathcal{I}'$.

4.2 Simulation with ϵ and $\Pi_{\mathcal{I}}$

We can assume that our interaction system $\mathcal{I}$ is composed of k symbols all of arity n and ϵ agents (which erase everything). This system has $\frac{k\times(k+1)}{2}$ proper rules between the k agents of arity n, n rules between these agents and ϵ which are the same (except for the symbol of the agent) and create n ϵ agents and a rule for ϵ and itself which gives the empty net.

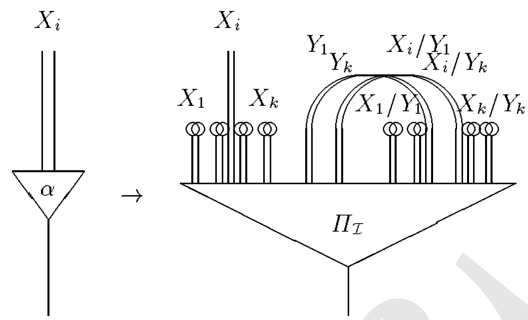
$\Pi_{\mathcal{I}}$ agents The interaction system $\mathcal{I}$ has k symbols which arity is n and the ϵ symbol. It is simulated by a system composed of two symbols: ϵ and $\Pi_{\mathcal{I}}$. $\Pi_{\mathcal{I}}$ has exactly $n \times k \times (k+2)$ auxiliary ports.



The auxiliary ports are grouped together by n. In the picture, they correspond to two vertical lines. Each group of n auxiliary ports are put in three different partitions. $X_1, \ldots, X_k$ are the inputs of the agent. One of this group corresponds to the n auxiliary ports of the initial agent of $\mathcal{I}$. The other input ports are connected to ϵ agents. The second group of auxiliary ports $Y_1, \ldots, Y_k$ are the inputs of the other agent when this agent interacts with another $\Pi_{\mathcal{I}}$ agent. The agent connects all the auxiliary ports in this partition to the auxiliary ports of the third group of ports. Finally, $X_1/Y_1, \ldots, X_k/Y_k$ are the interface of the right members of the rules generated by the interaction of this agent and another $\Pi_{\mathcal{I}}$ agent:

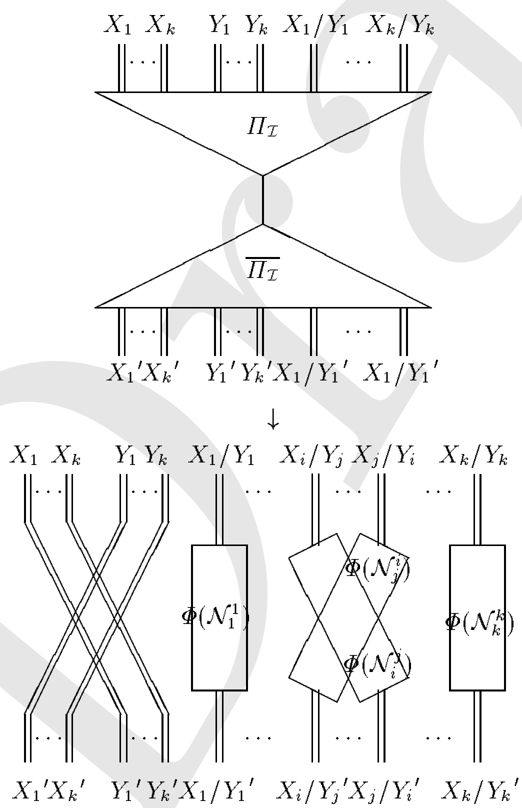
- $n \times k$ input ports X_i^t, $1 \leq t \leq n$ and $1 \leq i \leq k$;
- $n \times k$ intermediate ports Y_j^t, $1 \leq t \leq n$ and $1 \leq j \leq k$;
- $n^2 \times k$ rule ports $(X_i/Y_j)^t$, $1 \leq t \leq n$ and $1 \leq i, j \leq k$.

The ϵ agent of $\mathcal{I}$ is directly translated into itself. An agent α of $\mathcal{I}$, different from ϵ is simulated by an agent $\Pi_{\mathcal{I}}$, $n \times k$ links between auxiliary ports of this agent and $n \times (k^2 - 1)$ ϵ agents. In fact, the symbols of $\mathcal{I}$ that are different from ϵ are numbered from 1 to k. Thus, if the number associated to α is i, its translation is the net:



In this figure, the small circles are ϵ agents. The n input ports $X_i^1, \ldots, X_i^n$ correspond to the auxiliary ports of α. The other input ports are connected to ϵ agents. The $n \times k$ intermediate ports are connected to rule ports: for $1 \leq t \leq n$ and $1 \leq j \leq k$, Y_j^t is connected to $(X_i/Y_j)^t$. The other rule ports are connected to ϵ agents.

The rule between $\boldsymbol{\Pi_{\mathcal{I}}}$ and itself We need to define the interaction rule between two agents $\Pi_{\mathcal{I}}$. To simplify this presentation, we use the symmetric agent $\overline{\Pi_{\mathcal{I}}}$ for bottom agent. The rule is as follow:



In this rule, variables without $'$ are from top $\Pi_{\mathcal{I}}$ agent and variables with $'$ are from bottom $\overline{\Pi_{\mathcal{I}}}$ agent. In the right member of the rule, input port X_i^t of top agent is connected to the intermediate port $Y_i^{t'}$ of bottom agent. Symmetrically, $X_i^{t'}$ is connected to Y_i^t.

The groups of n rule ports $(X_i/Y_i)^1, \dots, (X_i/Y_i)^n$ and $(X_i/Y_i)^{1'}, \dots, (X_i/Y_i)^{n'}$ are connected to the translation of the rule between the agent number i of $\mathcal{I}$ and itself. This interaction net $\mathcal{N}_i^i$ is by hypothesis symmetric, thus $\Phi(\mathcal{N}_i^i)$ is also symmetric and this part of the rule between $\Pi_{\mathcal{I}}$ and itself is symmetric.

For $i \neq j$, the two groups $(X_i/Y_j)^1, \dots, (X_i/Y_j)^n$ and $(X_j/Y_i)^{1'}, \dots, (X_j/Y_i)^{n'}$ are connected to the translation $\Phi(\mathcal{N}_i^j)$ of the rule between the agent number i of $\mathcal{I}$ and the agent number j. Because there is an inversion of ports, the auxiliary ports corresponding to the side of the agent number i (the top agent on the figure) are connected to $(X_i/Y_j)^1, \dots, (X_i/Y_j)^n$ and the auxiliary ports corresponding to the side of the agent number j (the bottom agent on the figure) are connected to $(X_j/Y_i)^{1'}, \dots, (X_j/Y_i)^{n'}$. In the same way, the two groups of n rule ports $(X_j/Y_i)^1, \dots, (X_j/Y_i)^n$ and $(X_i/Y_j)^{1'}, \dots, (X_i/Y_j)^{n'}$ are connected to the translation $\Phi(\mathcal{N}_j^i)$ of the rule between the agent number i of $\mathcal{I}$ and the agent number j but in the opposite direction (upside-down). The 4 groups of rule ports $(X_i/Y_j)^1, \dots, (X_i/Y_j)^n$, $(X_j/Y_i)^1, \dots, (X_j/Y_i)^n$, $(X_i/Y_j)^{1'}, \dots, (X_i/Y_j)^{n'}$ and $(X_j/Y_i)^{1'}, \dots, (X_j/Y_i)^{n'}$ and the two translations $\Phi(\mathcal{N}_i^j)$ and $\Phi(\mathcal{N}_j^i)$ is symmetric.

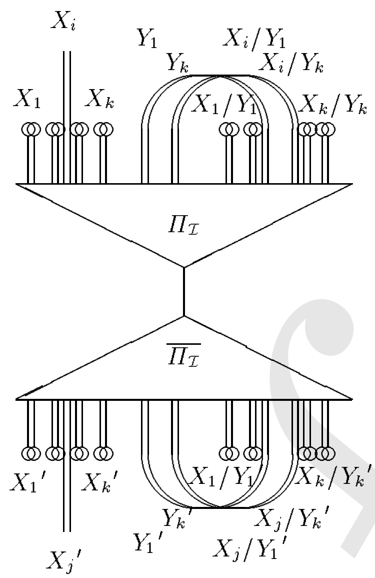
Thus the rule is completely symmetric. On the figure, because we write the rule between $\Pi_{\mathcal{I}}$ and $\overline{\Pi_{\mathcal{I}}}$, the right member of the rule has to be horizontally symmetric.

This rule and the erase rules with ϵ define an interaction system $\Pi_{\mathcal{I}}$ with 2 agents and 3 rules.

$\Pi_{\mathcal{I}}$ simulates $\mathcal{I}$ This interaction system $\Pi_{\mathcal{I}}$ simulates the interaction system $\mathcal{I}$. In fact, we just have to show that the translation of the interaction net constituted by two agents of $\mathcal{I}$ connected by their principal ports reduces to the translation of the right member of the rule between these two agents. This task is not so difficult to check.

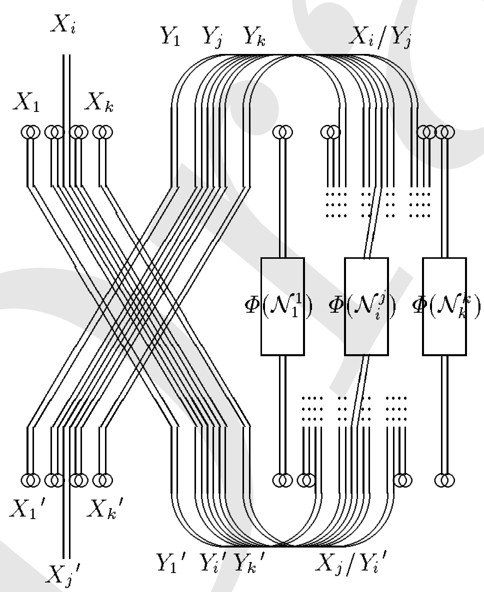
It is obvious for ϵ rules. For an agent α of $\mathcal{N}$ and another one β, the translation gives two agents $\Pi_{\mathcal{I}}$, several links and ϵ agents. In a first step, the two agents $\Pi_{\mathcal{I}}$ are reduced. This step replaces the two agents by a set of links and a set of translations of right members of the interaction rules from $\mathcal{I}$. A second step erases the right members of these rules that do not correspond to the interaction between α and β.

Below is a simulation of the interaction of the translation of an agent number i and an agent number j, $i \neq j$:



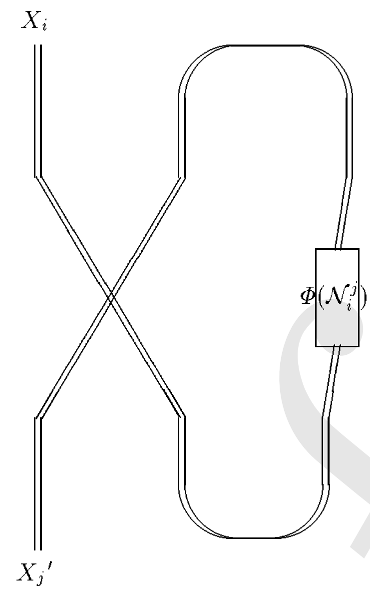
X_i
Y_1
X_i/Y_1
Y_k
X_i/Y_k
X_1
X_k
X_1/Y_1
X_k/Y_k
$\Pi_\mathcal{I}$
$\overline{\Pi_\mathcal{I}}$
$X_1{}'$
$X_k{}'$
X_1/Y_1
$X_k/Y_k{}'$
$Y_k{}'$
$X_j/Y_k{}'$
$Y_1{}'$
$X_j/Y_1{}'$
$X_j{}'$

↓



X_i
Y_1 Y_j Y_k
X_i/Y_j
X_1
X_k
$\Phi(\mathcal{N}_1^1)$
$\Phi(\mathcal{N}_i^j)$
$\Phi(\mathcal{N}_k^k)$
$X_1{}'$
$X_k{}'$
$X_j{}'$
$Y_1{}'$ $Y_i{}'$ $Y_k{}'$
$X_j/Y_i{}'$

↓



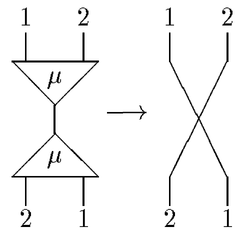
Universal system with 2 agents : first version Starting with a universal system $\mathcal{I}$, we obtain a universal system composed of only two agents $\Pi_{\mathcal{I}}$ and ϵ. For instance, with Lafont's combinators γ, δ and ϵ, we obtain a universal system with ϵ and an agent $\Pi_{\gamma,\delta}$ which has $2 \times 2 \times (2+2) = 16$ auxiliary ports.

5 Universal system with 2 agents and a minimum of ports

We have seen that the number of ports of $\Pi_{\mathcal{I}}$ agent is generally very big. For Lafont's universal system, the agent $\Pi_{\gamma,\delta}$ has 16 auxiliary ports. We can reduce this system to an interaction system with 2 agents, one with no auxiliary port and the other with only 3.

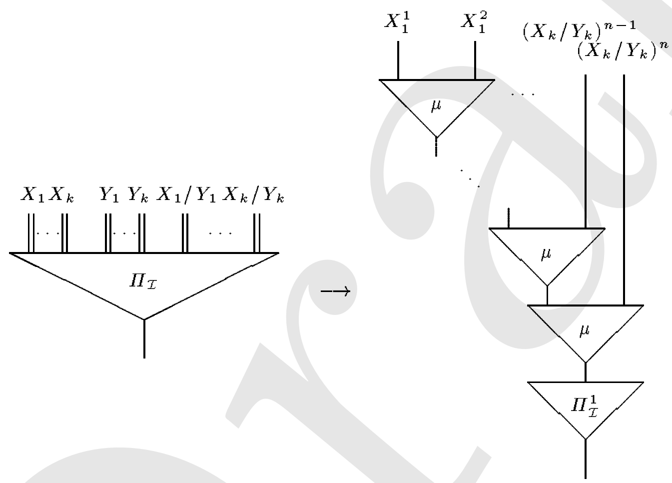
This transformation is done in three steps. The first step adds a multiplexor agent μ. The second step reduces the number of auxiliary ports of $\Pi_{\mathcal{I}}$ using the multiplexor. It leads to a new agent $\Pi^1_{\mathcal{I}}$ with only one auxiliary port. Finally, a last step merges together μ and $\Pi^1_{\mathcal{I}}$ in a single agent with three auxiliary ports.

Adding μ agents μ agents have 2 auxiliary ports. Their construction is the same as the multiplexor introduced in [8]. In fact, we can use either Lafont's combinators δ or γ. The rule between μ and itself is as follows (δ version):



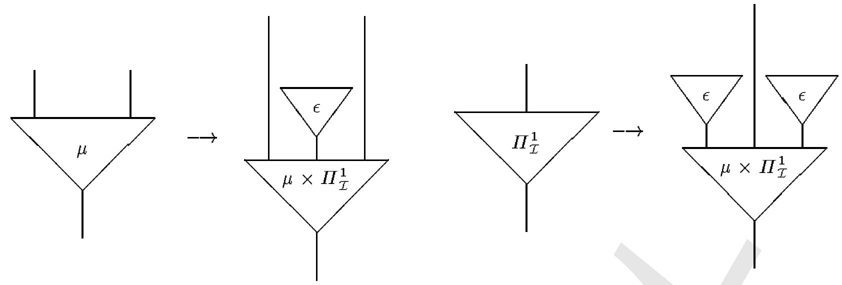
The multiplexor with 2 auxiliary ports can obviously be extended to multiplexors with $n \times k \times (k+2)$ auxiliary ports (the number of auxiliary ports of $\Pi_{\mathcal{I}}$) using $n \times k \times (k+2) - 1$ μ agents.

Reducing the ports of $\boldsymbol{\Pi_{\mathcal{I}}}$ to 1 auxiliary ports Then, we can transform $\Pi_{\mathcal{I}}$ into an agent $\Pi^1_{\mathcal{I}}$ with only 1 auxiliary port followed by a $n \times k \times (k+2)$ multiplexer:

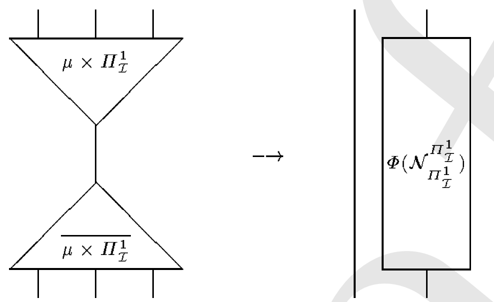


The rule between $\Pi^1_{\mathcal{I}}$ and itself is deduced from the rule between $\Pi_{\mathcal{I}}$ and itself by replacing in the right member of this rule each occurrence of $\Pi_{\mathcal{I}}$ by its translation and by folding both group of $n \times k \times (k+2)$ auxiliary ports with a $n \times k \times (k+2)$ multiplexor.

Merging $\boldsymbol{\mu}$ and $\boldsymbol{\Pi^1_{\mathcal{I}}}$ The functionalities of μ and $\Pi^1_{\mathcal{I}}$ do not overlap. In fact we only use the rule between μ and itself and $\Pi^1_{\mathcal{I}}$ and itself but never μ with $\Pi^1_{\mathcal{I}}$. As a consequence, we can put together these two agents into a single agent $\mu \times \Pi^1_{\mathcal{I}}$. The two translations from μ to $\mu \times \Pi^1_{\mathcal{I}}$ and from $\Pi^1_{\mathcal{I}}$ to $\mu \times \Pi^1_{\mathcal{I}}$ are as follows:



The rule between two $\mu \times \Pi^1_{\mathcal{I}}$ is as follows:



Universal system with 2 agents : second version Starting with a universal system $\mathcal{I}$, we obtain a universal system composed of only two agents $\mu \times \Pi^1_{\mathcal{I}}$ and ϵ. The first one has three auxiliary ports and the second one has no auxiliary port.

6 Conclusion

Three results are given in this paper. The first one gives a way to simulate every interaction system with a system composed of only two symbols. A corollary is that there exists a universal interaction system with only two symbols (in [8] the universal system has 3 symbols). In the last part of this article, it is shown how to reduce the number of auxiliary ports of one of the symbols to only 3. This leads to a universal system with two symbols, one without auxiliary port and the other with only 3.

We have succeeded in finding a simpler universal system than Lafont's one. However, the price is that the right member of the rule between $\Pi^1_{\mathcal{I}}$ or $\mu \times \Pi^1_{\mathcal{I}}$ and itself is big and not very pleasant (like Lafont's system). One may wonder if it is possible to find a system that has simpler right members for the rules. Moreover, an open question remains: is it possible to find a universal system with 2 agents one with 0 auxiliary ports and the other with 2 (less is obviously impossible)?

References

[1] D. Bechet. Proof transfornations in linear logic. In *Colloquium* PREUVES, RÉSEAUX ET TYPES. CIRM, Marseille, France, 1994.

[2] S. Gay. Combinators for interaction nets. In I. C. Mackie & R. Nagarajan C. L. Hankin, editor, *Proceedings of the Second Imperial College Department of Computing Workshop on Theory and Formal Methods*. Imperial College Press, 1995.

[3] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[4] G. Gonthier, M. Abadi, and J.-J. Levy. The geometry of optimal lambda reduction. In *Proceedings of the Nineteenth Annual Symposium on Principles of Programming Languages (POPL '90)*, pages 15–26, Albuquerque, New Mexico, January 1992. ACM Press.

[5] G. Gonthier, M. Abadi, and J.-J. Levy. Linear logic without boxes. In *Seventh Annual Symposium on Logic in Computer Science*, pages 223–234, Santa Cruz, California, June 1992. IEEE Computer Society Press.

[6] Y. Lafont. Interaction nets. In *Seventeenth Annual Symposium on Principles of Programming Languages*, pages 95–108, San Francisco, California, 1990. ACM Press.

[7] Y. Lafont. From proof nets to interaction nets. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*, pages 225–247. Cambridge University Press, 1995. Proceedings of the Workshop on Linear Logic, Ithaca, New York, June 1993.

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

[9] J. Lamping. An algorithm for optimal lambda calculus reduction. In *Seventeenth Annual Symposium on Principles of Programming Languages (POPL '90)*, pages 16–46, San Francisco, California, 1990. ACM Press.

[10] I. Mackie. *The Geometry of Implementation (an investigation into using the Geometry of Interaction for language implemetation)*. PhD thesis, Departement of Computing, Imperial College of Science, Technology and Medecine, 1994.

[11] I. Mackie. Interaction nets for linear logic. *Theoretical Computer Science*, 247:83–140, 2000.