Logic Colloquium '88
Ferro, Bonotto, Valentini and Zanardo (Editors)
© Elsevier Science Publishers B.V. (North-Holland), 1989

GEOMETRY OF INTERACTION 1 : INTERPRETATION OF SYSTEM F

jean–yves girard
équipe de logique, UA 753 du CNRS
mathématiques, Université Paris VII
t.45–55, 5$^{\bullet}$ étage
2, place Jussieu 75251 PARIS cedex 05

ABSTRACT : this paper develops for the first time a semantics of computation free from the twin drawbacks of *reductionism* (which leads to static modelisation) and *subjectivism* (which leads to syntactical abuses, in other terms, bureaucracy). The new approach initiated there rests on the use of a specific $\mathbb{C}^*$–algebra Λ^*, which has the distinguished property of bearing a (non associative) inner tensor product. To each proof–as–program Π of system $\mathbf{F}$ (or second order linear logic) is associated a pair $(\Pi^{\bullet},\sigma)$ of partial symmetries, σ representing the cuts of the proof, i.e. the dynamics. The dynamics is introduced as a way of eliminating σ, namely forming

$$\mathrm{EX}(\Pi^{\bullet},\sigma) = (1 - \sigma^2).\Pi^{\bullet}.(1 - \sigma\Pi^{\bullet})^{-1}.(1-\sigma^2).$$

a formula which makes sense only if $\sigma\Pi^{\bullet}$ is nilpotent. The nilpotency of $\sigma\Pi^{\bullet}$ is the mathematical way of expressing strong normalisation. In fact it is possible to prove that $\sigma\Pi^{\bullet}$ is always nilpotent, and the proof strongly relies on (the first mathematical attempt to give) the definition *ex nihilo* of the concept of type : a and b are orthogonal when ab is nilpotent, a type is a set of operators equal to its biorthogonal. As to relation with normalisation, the formula written above is faithful only w.r.t. a limited class of types ; this is due to a different conception of execution, where actions at distance (*global time*) are forbidden, but what is found is always consistent with syntax, if not always the same. This difference with syntax must be seen as the disparition of any kind of central unit in a network of parallel computers trying to compute a sequential algorithm. Coming back to Λ^*, it turns out that this algebra manipulates very simple finite electronic circuits, and it is therefore likely that this modelisation could work on a concrete *ad hoc* machine.

In a previous paper [6], we proposed the program of *geometry of interaction*. Its essential aim is to provide a semantics for computation free from the twin drawbacks of *reductionism* (which leads to static modelisation) and *subjectivism* (which leads to syntactical abuses, i.e. bureaucracy). As to its achievement, the program is divided into several steps :

A) show that a representative class of algorithms can be modelised by means of standard mathematics.

B) study the model for itself, without reference to the representative class chosen in i), so to answer abstractly to some questions, e.g. *what is a type* ?

C) study the *feasibility* of the execution involved in A–B, so to define an abstract computer that should not be too far from concrete ones. As we explained in [6], the geometrical analysis should essentially free us from *ad hoc* time, i.e. artificial causalities, overcontrol, and this means that we are clearly seeking a *parallel* computer.

This paper fulfills step A– by showing that one of the most standard typed λ–calculi, namely system $\mathbb{F}$ of the author [2], can be modelised by means of operator algebras. In fact it is more natural (and rewarding in view of step B to come) to first translate $\mathbb{F}$ into (second order) *linear logic* [3] (without additives), and then to interpret linear logic (without additives). This is what is achieved here. Let us remind the reader that the expressive power of system $\mathbb{F}$ is so big that any numerical algorithm which can be proved to terminate within second–order arithmetic, i.e. by means of current mathematics, can be represented in it : hence our dynamics is fairly general, and it is clearly impossible to find any real computable function which does not obey to it. Actually, an interpretation of pure λ–calculus seems at hand, since our interpretation "forgets types".

To each proof–as–program of system $\mathbb{F}$ (or linear logic), we associate an operator u of a certain C*–algebra Λ^*. Furthermore, u is given together with a partial symmetry σ, which is responsible for the dynamics : if σ is *absent* (i.e. $\sigma = 0$), then u is a *datum*, i.e. an algorithm with no dynamics, whereas, in the general case, something has to be done to transform the general *algorithm* (u,σ) into a datum, namely

$$\mathrm{EX}(u,\sigma) = (1-\sigma^2).u.(1-\sigma u)^{-1}.(1-\sigma^2) \qquad (\mathrm{EX})$$

In order to show that such a description is reasonable, we have only to relate it to the familiar syntactical approach, which involves *rewriting* : a datum will correspond to a cut–free proof, whereas a general algorithm will correspond to a proof with cuts, the cuts of the proof being described by σ. Our formula (EX) therefore expresses the normalisation process at work in Gentzen's *Hauptsatz* (or typed λ–calculi). This formula makes sense only because σu is *nilpotent* ; it is surprising to discover that the outstanding results of Gentzen, can receive, more than 50 years later, a purely mathematical formulation : the *Hauptsatz* states the nilpotency of some operator and the result of cut–elimination is the operator of (EX). In particular, proof–theory is not this kind of *forensic medecine* which is at work in too many papers on the subject. The fact that we eventually reach a formulation free from logic (in terms of functional analysis) does not invalidate the use of logical tools : in fact the simplest way to prove termination (i.e. nilpotency) will always be standard logical manipulation, "symbol pushing" ; but the existence of a model for this symbol pushing obviously brings in new methods, insights etc.

In fact (EX) corresponds to syntactic normalisation only in sufficiently simple cases (when the type is free from ?, $\exists$) : there is a distance between (EX) and syntactic manipulation in the general case. However, there is a principle of *associativity of cut* which basically asserts that two executions can be done as one (i.e. $\mathrm{EX}(u,\sigma+\tau) = \mathrm{EX}(\mathrm{EX}(u,\sigma),\tau)$), and which has the following consequence : if we first make an execution which is "wrong" from the syntactic viewpoint, but then we use the "wrong" result in a computation whose type is free from ?, $\exists$, then the final result is correct. The distance between syntactical normalisation and (EX) comes from the fact that our modelisation is without *global external time*, i.e. without any kind of synchronisation. Due to this lack of synchronisation, some "actions at distance" familiar to the syntax cannot be performed, at least cannot be performed before some input has been given, but eventually everything gets fine. We have therefore gained a new freedom w.r.t. time, and this should be essential in view of parallel execution.

In order to prove nilpotency, since there is a distance from syntax, it is necessary to give a semantic proof, namely to attempt to solve abstractly the questions "what is a type, an algorithm", etc. The answer given here to these abstract questions is particularly simple : two operators a and b are said to be *orthogonal* when the product ab is nilpotent ; a *type* is a set of operators equal to its biorthogonal. This gives the flavour of things to come in part B– of the program, but that's not it, since we cannot accept (prejudice of logician) that 0 belongs to all types. Also, fixing this defect would presumably give room to the neutral elements of linear logic, which are absent from our modelisation.

In pedantic terms, Λ^* is not a C*–algebra, since it contains another binary operation, namely a (non–associative) inner tensor product. Very crucial is the *message space*, which is a Boolean algebra $\mathcal{B}$, whose atoms are binary trees whose leaves are labelled by two objects, G and D. The operators which are used in the modelisation are the simple operations on such trees, basically changing the tree and permuting leaves, adding new leaves. These operators are finitary operations, but need the abstract framework of partial isometries to find their conceptualisation. The fact that all operations at work have this kind of concrete meaning is surely an encouragement for step C– to come.

So to speak, Λ^* internalises the fact that, due to the essential unicity of the Hilbert space, $\mathbb{H}$ is isomorphic to $\mathbb{H} \oplus \mathbb{H}$ and to $\mathbb{H} \otimes \mathbb{H}$. The internalisation of the first isomorphism is easy, and made in analogy with Cuntz' C*–algebra O_2 [1]. The internalisation of the second one is much more problematic (at least to us), and it is only recently that we found a way to do it, in analogy with the literature on monoidal categories, e.g. [7] : it is necessary to add isometries which compensate for the lack of associativity, etc., of the inner tensor product. The internalisation of direct sum is needed to interpret binary connectives such as multiplicatives, since it is then possible to reduce the size of a matrix in a way compatible with (EX). The inner tensor product is needed because duplication instructions have to be put on specific tensorial factors, so to avoid problems of mismatch, i.e. defects of commutation. Hence the algebra allows potential additive ($\oplus$) or multiplicative ($\otimes$) decomposition *ad libitum*.

The work was started two years ago, after the modelisation [6] of the multiplicative fragment of linear logic by means of permutations, with a formula corresponding to (EX). But the dynamical power of this fragment is ridiculous, and it was therefore necessary to adapt everything to the context of general linear logic. After a while, matrices with boolean coefficients were introduced, leaving then room for partial isometries. Then the theory remained for a long time a collection of recipes, since, due to the absence of the tensorial aspects, it was stumbling on impossible commutation problems. It is only very recently (September 1988) that the introduction of the inner tensor product immediately solved all these questions. The author is indebted to Loic Colson, Vincent Danos and Laurent Regnier, which have currently been working on proof–net–like execution of λ–terms, and whose work has been an encouragement to the author. Thanks are also due to Marouan Ajlani and Philippe Mathieu for many stimulating discussions.

I. the C*-algebra B(H)

I.1. C*–algebras

There is no hope of explaining what is a C*–algebra within this restricted space. We just give some milestones :

Take the Hilbert space $\mathbb{H} = \ell^2$, of square summable sequences of complex numbers, i.e. sequence $(z_n;n \in \mathbb{N})$ such that $\|(z_n)\|^2 = \Sigma\, z_n.\bar{z}_n < +\infty$ The space is equipped with the (sesquilinear) *scalar product*

$<(x_n),(y_n)> = \Sigma\, x_n.\bar{y}_n$.

A (bounded) *operator* u on $\mathbb{H}$ is a linear map from $\mathbb{H}$ to itself such that

$N(u) = \sup(\|u(z)\|;\|z\| = 1) < +\infty$; N(u) is the *norm* of u.

The space $\mathcal{B}(\mathbb{H})$ of all operators on $\mathbb{H}$ has the following properties :

i) it is a complex algebra (addition, composition, scalar multiplication) with unit 1, the identity operator.

ii) it is a Banach algebra (complete normed algebra).

iii) it is equipped with an involution : to u we can associate a unique u* (called the *adjoint* of u) defined by :

$<u(x),y> = <x,u^*(y)>$.

The following properties of the adjoint are easily checked :

$1^* = 1$

$u^{**} = u$

$(u + v)^* = u^* + v^*$

$(\lambda.u)^* = \bar{\lambda}u^*$

$(uv)^* = v^*.u^*$

$N(u) = N(u^*)$

$N(uu^*) = N(u)^2$

The last equality is the crucial one, and has a lot of consequences. For instance, if $vv^* = 0$, then $N(vv^*) = 0 = N(v)^2$, so $N(v) = 0$, hence $v = 0$.

A C*–algebra is a complex Banach algebra together with an involution * enjoying the properties just listed. Standard examples are :

the algebra $C(X,\mathbb{C})$ of continuous maps from a compact space X into $\mathbb{C}$; the involution is just pointwise complex conjugation. This example is known as the general form of a commutative C*–algebra.

the algebra $\mathcal{M}_n(C)$ of $n \times n$ square matrices ; the involution is just transconjugation.

norm– and *–closed subalgebras of $\mathcal{B}(\mathbb{H})$ where $\mathbb{H}$ is a Hilbert space. This is known to be the general form of an abstract C*–algebra. If we keep in mind that a Hilbert space is determined up to isomorphism by the cardinality of its Hilbertian basis, then ℓ^2 is so to speak "the" Hilbert space (denumerable basis), the other ones being either too small or too big for current needs : so our starting example is very general indeed, and we can always keep in mind that our operations in a C*–algebra should be easily understood in terms of ℓ^2 (or any other isomorphic space : the proviso is not trivial, since the isomorphism between two spaces may be complicated).

I.2. zoology of operators

It is useful to distinguish among operators of a C*–algebra $\mathcal{K}$, certain classes :

i) u is said to be *unitary* when $uu^* = u^*u = 1$. In $\mathcal{B}(\mathbb{H})$ this means that u is an isometry of $\mathbb{H}$ onto itself. If u is unitary, then the map $v \mapsto u^*vu$ of $\mathcal{K}$ into itself is an automorphism of $\mathcal{K}$.
The unitaries of the trivial C*–algebra $\mathbb{C}$ are just the complex numbers of modulus 1, i.e. the unit circle.

ii) u is said to be *hermitian* when $u = u^*$; expressions of the form vv^*, $v+v^*$ are always hermitian. In $\mathbb{C}$, the hermitians are just the real numbers.

iii) u is said to be a *projector* when u is hermitian and $u^2 = u$. In $\mathbb{H}$, given a closed subspace E of $\mathbb{H}$ and its orthogonal complement F, there is a unique decomposition of any vector x as $x = x' + x''$, $x' \in E$, $x'' \in F$, and the map $u(x) = x'$ is clearly a projector of $\mathcal{B}(\mathbb{H})$. Conversely, every projector of $\mathcal{B}(\mathbb{H})$ is of this form, which means that, in C*–algebras, we have an internal way of speaking of closed subspaces. When two projectors commute, then their product is easily seen to be a projector ; similarly, 1–u is a projector when u is, and $(1-u)(x) = x''$. This shows, that commuting projectors behave like a boolean algebra... but it is not in this way that we want to connect C*–algebras with logic.

iv) when u is both hermitian and unitary, u is said to be a *symmetry* : $u^2 = 1$, $u = u^*$. Symmetries are deeply related to projectors : in the case considered in iii), $u(x) = x' - x''$ is easily seen to be a symmetry, and every symmetry of $\mathcal{B}(\mathbb{H})$ is obtained in this way. In fact, when p is a projector, then $2p - 1$ is a symmetry ($(2p-1)^2 = 4p^2 - 4p + 1 = 1$, etc.), and conversely, given a symmetry s, $1/2.(1 + s)$ is a projector, which means that symmetries are just another way of internalising subspaces in a C*–algebra.

v) things become more serious with so–called *partial isometries* : u is a partial isometry when both uu^* and u^*u are projectors ; in fact it is enough to require that one of these products is a projector, because of the

LEMMA 1 :

If uu^* is a projector, so is u^*u.

PROOF : let $v = uu^*u - u$; then $vv^* = (uu^*)^3 - 2(uu^*)^2 + uu^* = 0$; it has been remarked that this forces $v = 0$, hence $uu^*u = u$, so $(u^*u)^2 = u^*u$. QED

In particular, if $u^*u = 1$, then uu^* is a projector, that nothing forces to be equal to 1. The following example is very basic : for this it is convenient to introduce the canonical base (b^n) of ℓ^2, by $b^n(m) = 0$ if $n \neq m$, $= 1$ if $n = m$. The sequence (z_n) of ℓ^2 can therefore be written as $\Sigma\, z_n.b^n$; define operators p and q by :

$p(\Sigma\, z_n.b^n) = \Sigma\, z_n.b^{2n}$ $\qquad$ $q(\Sigma\, z_n.b^n) = \Sigma\, z_n.b^{2n+1}$

it is easily checked that

$p^*(\Sigma\, z_n.b^n) = \Sigma\, z_{2n}.b^n$ $\qquad$ $q^*(\Sigma\, z_n.b^n) = \Sigma\, z_{2n+1}.b^n$

Obviously, p and q do not erase information, they display it on a smaller space, whereas p^* and q^* erase information. It is also easily checked that

$p^*p = q^*q = 1$, and that

$pp^*(\Sigma\, z_n.b^n) = \Sigma\, z_{2n}.b^{2n}$ $\qquad$ $qq^*(\Sigma\, z_n.b^n) = \Sigma\, z_{2n+1}.b^{2n+1}$

so that

$pp^* + qq^* = 1$.

Such operators u are called partial isometries because, when they act on concrete Hilbert spaces, u sends isometrically the subspace defined by the projector u^*u (*initial* projector) onto the subspace defined by the projector uu^* (*final* projector). Typically, p sends isometrically ℓ^2 onto the space of sequences "without" odd coefficients.

It might be of interest to give an alternative description of p and q : consider the Cantor space $X = \{0,1\}^{\mathbb{N}}$ of denumerable sequences of 0 and 1. This space is equipped with a measure μ, and one can form the Hilbert space $L^2(\mu)$ of square integrable complex functions on X. The scalar product is the familiar $\langle f,g \rangle = \int \bar{f}g d\mu$; define

$p(f)(0 \# s) = f(s).\sqrt{2} = q(f)(1 \# s)$

$p(f)(1 \# s) = 0 = q(f)(0 \# s)$

where $0\#s$ denotes the result of prefixing 0 to the sequence s, etc. ; it is immediate that

$p^*(f)(s) = f(0 \# s).\sqrt{2}/2$ $\qquad$ $q^*(f)(s) = f(1 \# s).\sqrt{2}/2$

the coefficients $\sqrt{2}$, $\sqrt{2}/2$ are needed to keep the norm constant : for instance, f is send by p on g which is "the same", but on a space twice smaller, which means that the integrals have to be renormalised, and the square root comes from the fact that the norm is defined via a quadratic expression.

This example can be used to demonstrate how close C^*–algebras can be w.r.t. computation : imagine an infinite stack of 0 and 1 : p has the meaning of "push 0", q means "push 1" ; the meaning of p^* is "pop 0", q^* means "pop 1". However one would first argue that "pop 0" means *nothing*, since we don't know what is the stack. But "pop 0" and "pop 1" together mean "pop", i.e. by combining p^* and q^* it will be possible to describe an action depending on the last value on the stack. A much more serious objection is that a concrete stack may be empty, and that this modelisation has the same defects as the algebraisation of school arithmetic, which may yield a negative answer for the capacity of a bottle. Here we stumble on the most fascinating question raised by this kind of approach : *when p^* is to be applied to an empty stack, the operation cannot be performed, until the stack is filled.* One has only to make sure that there is no deadlock, namely that there is somewhere another operation that can be performed. The study of this clearly belongs to part C of our program, and should contribute to answer the question "*what is time in computation*". Now, in order to make our idea work it will be necessary to replace the condition $pp^*+qq^* = 1$ by the weaker $p^*q = 0$.

To end with partial isometries, they have an unpleasant feature, namely that the product uv of two partial isometries is *not* always a partial isometry (strong difference with unitaries !) ; but when the initial projector u^*u of u commutes with the final projector vv^* of v, then uv is a partial isometry.

I.3. direct sums

The direct sum of Hilbert spaces (e.g. $\mathbb{H}$ and $\mathbb{H}'$) is defined by taking their direct sum as vector spaces, i.e. formal expressions $x \oplus x'$, with the scalar product

$$<x \oplus x',y \oplus y'> = <x,x'> + <y,y'> ;$$

the Hilbert space $\ell^2 \oplus \ell^2$ has a natural basis formed of the vectors $b^n \oplus 0$ and $0 \oplus b^m$; since this basis is denumerable, there are isomorphisms between ℓ^2 and $\ell^2 \oplus \ell^2$, typically those induced by a bijection between $\mathbb{N}$ and $\mathbb{N} + \mathbb{N}$. If α is such an isomorphism, it is immediate that α can be written $\alpha(x) = p^*(x) \oplus q^*(x)$, for some operators p and q on ℓ^2. Now it is immediate that

(1) $pp^* + qq^* = 1$

(2) $p^*p = q^*q = 1$.

So to speak, p and q internalise α, i.e. enable us to use α, while staying within endomorphisms of $\mathbb{H} = \ell^2$. For obvious reasons, combinations of p and q will be enough to speak of isometries between $\mathbb{H}$ and any finite direct sum of copies of $\mathbb{H}$, $\mathbb{H}^n$, $n \neq 0$. We basically need to define $p_1,\ldots,p_n$, such that

(1') $\Sigma\, p_i p_i^* = 1$

(2') $p_i^* p_i = 1$ for $i = 1,\ldots,n$

There are a lot of solutions, e.g. for $n = 5$, randomly :

$p_1 = p^2$, $p_2 = pqp$, $p_3 = pq^2$, $p_4 = q^2$, $p_5 = qp$, or

$p_1 = q$, $p_2 = pq$, $p_3 = p^2q$, $p_4 = p^3q$, $p_5 = p^4$; the equations (1) and (2) are enough to establish their generalisations (1') and (2') to 5 partners in both cases.

Once we have access to the p_i we can speak of operators on $\mathbb{H}^n$, but stay within operators of $\mathbb{H}$. If we denote by $\mathbb{H}_i$ the i–th factor of the direct sum $\mathbb{H}^n$, then any operator u on $\mathbb{H}^n$ is faithfully described by a matrix (u_{ij}) made of operators of $\mathbb{H}$, u_{ij} being obtained from u by restricting the domain of u to $\mathbb{H}_j$ and its codomain to $\mathbb{H}_i$. In fact it is easy to see that if $w = u + v$, (resp. $\lambda.u$, uv, u^*), then $w_{ij} = u_{ij} + v_{ij}$ (resp. $\lambda.u_{ij}$, $\Sigma\, u_{ik}v_{kj}$, $u_{ji}{}^*$), i.e. operators on $\mathbb{H}^n$ are the same thing as n × n matrices with as coefficients operators on $\mathbb{H}$. It follows from the isomorphism between $\mathbb{H}$ and $\mathbb{H}^n$ induced by the p_i that the algebra $\mathcal{M}_n(\mathcal{B}(\mathbb{H}))$ of operators on $\mathbb{H}^n$ is isomorphic to $\mathcal{B}(\mathbb{H})$, namely to a matrix (u_{ij}) associate the operator

$\Phi((u_{ij})) = \Sigma\, p_i u_{ij} p_j{}^*$.

The inverse of Φ is defined by :

$\Psi(u) = (p_i{}^* u p_j)$.

The fact that Φ, Ψ are isomorphisms is obvious from the way we found these formulas. It may be of interest to try to verify some properties of Φ and Ψ directly by algebraic manipulations ; for instance, one will need the property $p_j{}^*p_i = 0$ for $i \neq j$ (proof : first observe that, if u+v, u, v are projectors, then $uv = 0$, since $(u+v)^2 = u+v+2uv$. From this, it is easy to get a n–ary analogue : if $u_1,\ldots,u_n$ are projectors and $\Sigma\, u_i$ is a projector, then $u_iu_j = 0$ for $i \neq j$. Applying this to $u_i = p_ip_i{}^*$, we get for $i \neq j$ $p_jp_j{}^*p_ip_i{}^* = 0$, hence $p_j{}^*p_i = p_j{}^*(p_jp_j{}^*p_ip_i{}^*)p_j = 0$. QED)

In fact, the situation is quite general : as soon as a C*–algebra $\mathcal{K}$ contains operators p and q enjoying (1) and (2), then one easily defines isomorphisms between the algebras $\mathcal{M}_n(\mathcal{K})$ and $\mathcal{K}$; this can also be used to define isomorphisms between $\mathcal{M}_n(\mathcal{K})$ and $\mathcal{M}_p(\mathcal{K})$. In the sequel we shall meet the following construction : *contract the last two rows/columns of a matrix*, which amounts to define an isomorphism between $\mathcal{M}_{n+1}(\mathcal{K})$ and $\mathcal{M}_n(\mathcal{K})$: in fact to a n+1 × n+1 matrix (u_{ij}) we associate a n × n matrix (v_{ij}) as follows :

$v_{ij} = u_{ij}$ for $i,j \neq n$

$v_{in} = u_{in}p^* + u_{in+1}q^*$ for $i \neq n$

$v_{nj} = pu_{nj} + qu_{nj}$ for $j \neq n$

$v_{nn} = pu_{nn}p^* + pu_{nn+1}q^* + qu_{n+1n}p^* + qu_{n+1n+1}q^*$

Conversely, the process of *splitting an row/column*, i.e. passing from (v_{ij}) to (u_{ij}) is defined by :

$u_{ij} = v_{ij}$ for $i,j \neq n, n+1$

$u_{in} = v_{in}p$ $\quad$ $u_{in+1} = v_{in}q$ for $i \neq n,n+1$

$u_{nj} = p^*v_{nj}$ $\quad$ $u_{n+1j} = q^*v_{nj}$ for $j \neq n,n+1$

$u_{nn} = p^*v_{nn}p$ $\quad$ $u_{nn+1} = p^*v_{nn}q$ $\quad$ $u_{n+1n} = q^*v_{nn}p$ $\quad$ $u_{n+1n+1} = q^*v_{nn}q$.

In fact, in our algebra Λ^*, we shall prefer to work with the weaker condition $p^*q = 0$. When we no longer have $pp^* + qq^* = 1$, it is still true that there is an isomorphism which contract rows, but this isomorphism is not onto ; moreover, it does not send the unit of $\mathcal{M}_{n+1}(\mathcal{K})$ on the unit of $\mathcal{M}_n(\mathcal{K})$, but only on a projector of $\mathcal{M}_n(\mathcal{K})$. But this is enough for our purposes. There is no longer any kind of isomorphism which splits rows.

II.4. tensor product

The tensor product of Hilbert spaces $\mathbb{H}$ and $\mathbb{H}'$ is defined in a more complex way : we first form the algebraic tensor product of $\mathbb{H}$ and $\mathbb{H}'$, i.e. the space of all linear combinations of expressions $x \otimes x'$ *with coefficients in* $\mathbb{C}$, quotiented by the relations :

$$x \otimes (x' + y') = x \otimes x' + x \otimes y' \qquad (x + y) \otimes x' = x \otimes x' + y \otimes x'$$
$$(\lambda.x) \otimes x' = x \otimes (\lambda.x') = \lambda.(x \otimes x')$$

that we equip with a scalar product defined by

$$<x \otimes x',y \otimes y'> = <x,x'>.<y,y'>$$

the space obtained is not complete in general, and one completes it w.r.t. the scalar product norm so that to get a Hilbert space $\mathbb{H} \otimes \mathbb{H}'$.

Given operators u on $\mathbb{H}$ and u' on $\mathbb{H}'$, it is possible to define a unique operator v on their algebraic tensor product, such that

$v(x \otimes x') = u(x) \otimes u'(x')$; v is then extended by continuity to an operator $u \otimes u'$ on $\mathbb{H} \otimes \mathbb{H}'$.

The following are easy to verify :

$$1 \otimes 1 = 1$$
$$(u + v) \otimes u' = u \otimes u' + v \otimes u' \qquad u \otimes (u' + v') = u \otimes u' + u \otimes v'$$
$$(\lambda.u) \otimes v = u \otimes (\lambda.v) = \lambda.(u \otimes v)$$
$$(u \otimes u')(v \otimes v') = (uu') \otimes (vv')$$
$$(u \otimes v)^* = u^* \otimes v^*.$$

The most basic claim about the tensor product construction is that (up to a completely straightforward isomorphism), $\mathbb{H} \otimes (\mathbb{H}' \otimes \mathbb{H}'')$ is equal to $(\mathbb{H} \otimes \mathbb{H}') \otimes \mathbb{H}''$, i.e. the construction is associative. In the same way, there is an *isomorphism* between $\mathbb{H} \otimes \mathbb{H}'$ and $\mathbb{H}' \otimes \mathbb{H}$.

We also need to internalise the tensor product, i.e. we want to be able to speak of operators on some tensor power of $\mathbb{H}$, while staying within operators on $\mathbb{H}$; the answer is definitely much more complicated than the one corresponding to the direct sum. W.l.o.g. we will assume that $\mathbb{H}$ is ℓ^2. In that case, observe that $\ell^2 \otimes \ell^2$ is the space of square summable sequences of complex numbers, indexed by $\mathbb{N} \times \mathbb{N}$. Since this space has clearly a Hilbertian basis (c^{nm}) defined by $c^{nm}(nm) = 1$, $c^{nm}(ij) = 0$ if $ij \neq nm$, and $\mathbb{N}$ can be put in bijection with $\mathbb{N} \times \mathbb{N}$, there is an isomorphism β between $\mathbb{H}$ and $\mathbb{H} \otimes \mathbb{H}$. In particular, given any two operators u and v on $\mathbb{H}$, we can form a new operator $w = \beta^{-1}(u \otimes v)\beta$. which is still an operator on $\mathbb{H}$. We shall use the notation $u \otimes v$ for w, which does not harm, since our aim is to internalise everything and that therefore $\mathbb{H} \otimes \mathbb{H}$ will never actually occur !

The inner tensor product enjoys the following :

(3) $1 \otimes 1 = 1$

(4) $(u \otimes v)(u' \otimes v') = uu' \otimes vv'$

(5) $(u + u') \otimes v = (u \otimes v) + (u' \otimes v)$ $\qquad u \otimes (v + v') = (u \otimes v) + (u \otimes v')$

(6) $(\lambda.u) \otimes v = u \otimes (\lambda.v) = \lambda.(u \otimes v)$

(7) $(u \otimes v)^* = u^* \otimes v^*$

However, in order to go on, we would like to express the associativity of the inner tensor product ; however, a principle like $u \otimes (v \otimes w) = (u \otimes v) \otimes w$ is obviously wrong : if $n,m \mapsto <n,m>$ is a bijection between $\mathbb{N}$ and $\mathbb{N} \times \mathbb{N}$, it is impossible to have $<<n,m>,p> = <n,<m,p>>$ for all n,m,p. But there is however a bijection between $\mathbb{N}$ and $\mathbb{N}$, which answers for this idea of associativity :

$<n,<m,p>> \mapsto <<n,m>,p>$,

this bijection induces a unitary operator t on $\mathbb{H}$; this operator enjoys

(8) $tt^* = t^*t = 1$

(9) $t(u \otimes (v \otimes w)) = ((u \otimes v) \otimes w)t$

Furthermore, t enjoys certain relations (which are inspired from the standard Mac Lane–Kelly [7] coherence relations for monoidal categories), and which have the following meaning : t enables us to change any expression involving $\otimes$ into an isomorphic one, by changing the bracketing ; in case two combinations do the same job then they are equal. In fact it suffices to write the analogue of the standard Mac Lane "pentagon" :

(10) $t^2 = (t \otimes 1)t(1 \otimes t)$, which is enough to generate all these relations.

To answer for the idea of commutativity, we introduce another unitary operator, induced by the function

$<n,m> \quad \mapsto <m,n>$

We obviously get the following :

(11) $ss^* = s^*s = 1$

(12) $s(u \otimes v) = (u \otimes v)s$

together with the coherence relations :

(13) $s = s^*$

(14) $tst = (s \otimes 1)t(1 \otimes s)$

It remains to think at a possible "neutral" element for the inner tensor product. In fact the situation is hopeless, since for instance $1 \otimes u$ is not even isomorphic with u. However, it makes sense to relate $1 \otimes u$ with u by means of a partial isometry r : $(1 \otimes u)r = ru$, $r^*r = 1$. For instance, if r is defined from the injective function $\quad n \mapsto <0,n>$; in fact since p can be defined from an arbitrary injective function whose codomain has an infinite complement, there is no obstacle to assume that $p = r$, reducing thus our primitives (in fact we shall take for q the operator induced by the function $n \mapsto <1,n>$). We therefore get

(15) $(1 \otimes u)p = pu$ $\qquad$ $(1 \otimes u)q = qu$

together with new coherence relations

(16) $tp = p \otimes 1$ $\qquad$ $tq = q \otimes 1$.

I.5. The C*-algebra $\mathcal{B}(\mathbb{H})$

For the moment on, it will be more convenient to modelise things by use of $\mathcal{B}(\mathbb{H})$; in fact in $\mathcal{B}(\mathbb{H})$, we shall only retain p,q,s,t and the binary operation $\otimes$. Later on (part III), we shall investigate what has been actually used, and slight modifications will be made, and an abstract C*-algebra Λ^* will be proposed. The introduction of Λ^* at this early moment would spoil the intuition. Remember that we no longer have $pp^* + qq^* = 1$, but only the weaker form $p^*q = 0$. However, since the stronger form is much more intuitive, one may read part II with this stronger form in mind. Some proofs are slightly complicated by the fact that we cannot use the strong form.

II. interpretation of linear proofs in B(H)

II.1. interpretation of proofs

We shall consider second order linear logic without additives or neutral elements (these features are not needed to interpret system $\mathbb{F}$, so please don't weep), whose formulas are built, using the connectives $\otimes$, ⅋, ! and ? and the quantifiers $\forall\alpha$ and $\exists\alpha$, from literals α, $\alpha^\perp$, β, $\beta^\perp$,..., i.e. propositional variables and their negations. Keep in mind that (linear) negation is *defined* by De Morgan–like formulas, i.e. that there is no connective for negation. We shall keep the sequent calculus as it is formulated in [3], with only a minor difference : we shall memorize in the sequent the cuts already made, i.e. a sequent will look like $\vdash [\Delta]$, Γ, where Δ lists the cuts that have been made during the proof of $\vdash \Gamma$. In general if Γ consists of n formulas, and Δ of m formulas, we shall represent a proof Π of $\vdash [\Gamma]$, Δ by means of an operator $\Pi^\bullet$ in the C*-algebra $\mathcal{M}_{2m+n}(\mathcal{B}(\mathbb{H}))$. Do not forget that this algebra is isomorphic to a subalgebra of $\mathcal{B}(\mathbb{H})$, but the isomorphism depends on the choice of partial isometries $p_1,\ldots,p_{2m+n}$, so it is slightly more natural to work with matrices, especially we get more manageable expressions... but all this is inessential bureaucracy. Together with $\Pi^\bullet$, we shall consider the matrix (σ_{ij}) defined by

$\sigma_{2i,2i-1} = \sigma_{2i-1,2i} = 1$ for $i = 1,\ldots,m$, all other coefficients being 0.

σ is hermitian and satisfies $\sigma^3 = \sigma$: one can say that σ is a *partial symmetry*. (From $\sigma^3 = \sigma$, it follows that $\sigma^2 = \sigma\sigma^* = \sigma^*\sigma$ is a projector, hence σ is a partial isometry with σ^2 as its *support*, i.e. both its domain and codomain ; on this subspace, σ acts like a symmetry.) $1 - \sigma^2$ is the projector whose matrix is defined by

$\pi_{2m+i} = 1$ for $i = 1,\dots,n$, all other coefficients being 0.

In the case of a cut–free proof, we have $m = 0$, then $\sigma = 0$ and $\pi = 1$.

We shall now define $\Pi^\bullet$ by following the rules of sequent calculus :

i) Π is an axiom $\vdash A, A^\perp$; then $m = 0$, $n = 2$; $\Pi^\bullet$ is defined to be the antidiagonal matrix :

$\Pi^\bullet_{11} = \Pi^\bullet_{22} = 0 \qquad \Pi^\bullet_{12} = \Pi^\bullet_{21} = 1$

ii) Π is obtained by means of the cut–rule applied to subproofs Π' and Π'' ; assume that Π' ends with $\vdash [\Delta'], A, \Gamma'$ and Π'' ends with $\vdash [\Delta''], A^\perp, \Gamma''$, so that Π ends with $\vdash [\Delta, \Delta', A], \Gamma', \Gamma''$; the integers at work in Π', Π'', Π are respectively (m',n'), (m'',n'') and $(m'+m''+1, n'+n'')$. We have already by hypothesis the matrices $\Pi'^\bullet$ and $\Pi''^\bullet$; it is convenient to assume that the indices for these matrices have been chosen as follows :

$1,\dots,2m',2(m'+m'')+1,2(m'+m''+1)+1,\dots,2(m'+m''+1)+n'$ for $\Pi'^\bullet$,

$2m'+1,\dots,2(m'+m''),2(m'+m''+1),2(m'+m''+1)+n'+1,\dots,2(m'+m''+1)+n''$ for $\Pi''^\bullet$.

$\Pi^\bullet$ is defined by

$\Pi^\bullet_{ij} = \Pi'^\bullet_{ij}$ when $\Pi'^\bullet_{ij}$ is defined

$\Pi^\bullet_{ij} = \Pi''^\bullet_{ij}$ when $\Pi''^\bullet_{ij}$ is defined

$\Pi^\bullet_{ij} = 0$ otherwise.

So to speak, $\Pi'^\bullet$ and $\Pi''^\bullet$ have been put together so that to respect the orderings of $[\Delta]$ and Γ.

iii) Π is obtained from Π' by means of an exchange rule : Π' ends with $\vdash [\Delta], \Gamma'$, whereas Π ends with $\vdash [\Delta], \Gamma$, where Γ is obtained by exchanging two consecutive formulas A_i and A_{i+1} of Γ'. Then $\Pi^\bullet$ is clearly defined from $\Pi'^\bullet$ by exchanging the rows $2m+i$ and $2m+i+1$.

iv) Π is obtained from Π' by means of a "par" rule : Π' ends with $\vdash [\Delta], \Gamma', A, B$, whereas Π ends with $\vdash [\Delta], \Gamma', A \parr B$. Then $\Pi^\bullet$ will be obtained from $\Pi'^\bullet$ by contracting the last two indices into one, by means of p and q :

$\Pi^\bullet_{ij} = \Pi'^\bullet_{ij}$ for $i, j \neq 2m+n$,

$\Pi^\bullet_{i,2m+n} = \Pi'^\bullet_{i,2m+n}p^* + \Pi'^\bullet_{i,2m+n+1}q^*$ for $i \neq 2m+n$

$\Pi^\bullet_{2m+n,j} = p\Pi'^\bullet_{2m+n,j} + q\Pi'^\bullet_{2m+n+1,j}$ for $j \neq 2m+n$

$$\Pi^\bullet_{2m+n,2m+n} = p\Pi'^\bullet_{2m+n,2m+n}p^* + p\Pi'^\bullet_{2m+n,2m+n+1}q^* + q\Pi'^\bullet_{2m+n+1,2m+n}p^* + q\Pi'^\bullet_{2m+n+1,2m+n+1}q^*$$

v) if Π is obtained from Π' and Π'' by means of a "times" rule, let's assume that Π', Π'', Π respectively end with $\vdash [\Delta'], \Gamma', A$, $\vdash [\Delta''], \Gamma'', B$, $\vdash [\Delta', \Delta''], \Gamma', \Gamma'', A \otimes B$, so that the integers involved are respectively $(m', n'+1)$, $(m'', n''+1)$, $(m'+m'', n'+n''+1)$:

in a first step we put together $\Pi'^\bullet$ and $\Pi''^\bullet$ so that to form a matrix of dimension $2(m'+m'')+n'+n''+2$, the last two rows/columns corresponding to the formulas A and B ; we know how to do this from the case of the cut rule ii).

using p and q as in case iv) the last two rows are contracted into one

the result of this process if $\Pi^\bullet$.

vi) if Π is obtained from Π' by means of an "of course" rule, then let's assume that Π', Π respectively end with $\vdash [\Delta], ?\Gamma', A$ and $\vdash [\Delta], ?\Gamma', !A$, so that in both cases the integers involved are $(m,n'+1)$, hence $n = n'+1$; we define $\Pi^\bullet$ from $\Pi'^\bullet$ as follows :

$\Pi^\bullet_{ij} = t(1 \otimes \Pi'^\bullet_{ij})t^*$ for $i,j \neq 2m+n$

$\Pi^\bullet_{i,2m+n} = t(1 \otimes \Pi'^\bullet_{i,2m+n})$ for $i \neq 2m+n$

$\Pi^\bullet_{2m+n,j} = (1 \otimes \Pi'^\bullet_{2m+n,j})t^*$ for $j \neq 2m+n$

$\Pi^\bullet_{2m+n,2m+n} = 1 \otimes \Pi'^\bullet_{2m+n,2m+n}$

vii) if Π is obtained from Π' by means of a "dereliction" rule, then assume that Π' and Π respectively end with $\vdash [\Delta], \Gamma', A$ and $\vdash [\Delta], \Gamma', ,?A$, so that in both cases the integers involved are $(m,n'+1)$, hence $n = n'+1$; we define $\Pi^\bullet$ from $\Pi'^\bullet$ as follows :

$\Pi^\bullet_{ij} = \Pi'^\bullet_{ij}$ for $i,j \neq 2m+n$

$\Pi^\bullet_{i,2m+n} = \Pi'^\bullet_{i,2m+n}p^*$ for $i \neq 2m+n$

$\Pi^\bullet_{2m+n,j} = p\Pi'^\bullet_{2m+n,j}$ for $j \neq 2m+n$

$\Pi^\bullet_{2m+n,2m+n} = p\Pi'^\bullet_{2m+n,2m+n}p^*$

viii)if Π is obtained from Π' by means of a "weakening" rule, then assume that Π' and Π respectively end with $\vdash [\Delta], \Gamma'$ and $\vdash [\Delta], \Gamma',?A$, so that the integers involved are (m,n') and $(m,n'+1)$, so that $n = n'+1$; we define $\Pi^\bullet$ from $\Pi'^\bullet$ as follows :

$\Pi^\bullet_{ij} = \Pi'^\bullet_{ij}$ for $i,j \neq 2m+n$

$\Pi^\bullet_{ij} = 0$ in all other cases

ix) if Π is obtained from Π' by means of a "contraction" rule, then assume that Π' and Π respectively end with $\vdash [\Delta], \Gamma', ?A, ?A$ and $\vdash [\Delta], \Gamma',?A$, so that the integers involved are $(m,n+1)$ and (m,n) ; we define $\Pi^\bullet$ from $\Pi'^\bullet$ as in the case of the "par" rule (case (iv) above) with *only one difference* : we do not use p and q, but $p\otimes 1$ and $q\otimes 1$. In fact we have here a lot of possible choices of the form $r\otimes 1$, $s\otimes 1$, where r and s are such that $r^*r = s^*s = 1$, $r^*s = 0$.

x) if Π is obtained from Π' by means of a "for all" rule, then assume that Π' and Π respectively end with $\vdash [\Delta], \Gamma', A$ and $\vdash [\Delta], \Gamma', \forall\alpha A$, so that the integers involved are (m,n) in both cases. Let $\Pi^\bullet = \Pi'^\bullet$.

xi) if Π is obtained from Π' by means of a "there is" rule, then assume that Π' and Π respectively end with $\vdash [\Delta], \Gamma', A[B/\alpha]$ and $\vdash [\Delta], \Gamma', \exists\alpha A$, so that the integers involved are (m,n) in both cases. Let $\Pi^\bullet = \Pi'^\bullet$.

II.2. the dynamics : some crucial cases

What follows is the heart of the matter : we introduce the expression

$$EX(\Pi^\bullet,\sigma) = (1-\sigma^2).\Pi^\bullet.(1 - \sigma\Pi^\bullet)^{-1}.(1-\sigma^2)$$

This matrix M is of size $2m+n$, but its only non-zero coefficients are among the $M_{2m+i,2m+j}$ for $i,j = 1,\dots,n$. Hence M could in fact be seen as a matrix of size n. If Π ends with $\vdash [\Delta], \Gamma$, then M (seen as a $n \times n$ matrix) could be the matrix of a proof of $\vdash \Gamma$, i.e. could be the matrix of the proof obtained from Π by normalisation, or something close to it. We shall now consider some cases, and see what is the matrix M ; this will not completely settle the question, but the essential ideas of the proof to come will have been met.

i) assume that Π ends with a cut, one premise of which is an axiom, e.g.

$$\dfrac{\begin{array}{c}\Pi' \\ \vdash A,\ \Gamma\end{array} \qquad \vdash A^{\perp},\ A}{\vdash [A],\ \Gamma,\ A}$$

i.e. Π is a cut on the formula A, the premise proved by Π' being cut–free, the other one being an axiom. Then $m = 1$, $n = n'+1$, and the matrix $\Pi^{\bullet}$ is $2+n \times 2+n$

$\Pi^{\bullet}_{11} = \Pi'^{\bullet}_{11}$ $\qquad$ $\Pi^{\bullet}_{1,2+j} = \Pi'^{\bullet}_{1,1+j}$ $(j = 1,\ldots,n')$

$\Pi^{\bullet}_{2+i,1} = \Pi'^{\bullet}_{1+i,1}$ $(i = 1,\ldots,n')$ $\qquad$ $\Pi^{\bullet}_{2+i,2+j} = \Pi'^{\bullet}_{1+i,1+j}$ $(i,j = 1,\ldots,n')$

$\Pi^{\bullet}_{2,2+n} = \Pi^{\bullet}_{2+n,2} = 1$ $\qquad$ all other coefficients null.

σ is the partial symmetry whose only non zero coefficients are $\sigma_{12} = \sigma_{21} = 0$.

Let us compute $EX(\Pi^{\bullet},\sigma)$: if we set $\pi = 1 - \sigma^2$, then it is of the form

$$\pi EX'(\Pi^{\bullet},\sigma)\pi,$$

where $EX'(\Pi^{\bullet},\sigma)$ is defined via an infinitary expansion

$$\Pi^{\bullet} + \Pi^{\bullet}\sigma\Pi^{\bullet} + \Pi^{\bullet}\sigma\Pi^{\bullet}\sigma\Pi^{\bullet} + \ldots$$

the only hope to define an operator in this way (since σ and Π have norm 1) is the nilpotency of $\sigma\Pi^{\bullet}$. In the case we are facing, the developemt will be cut after the first three monomials.

computation of $v = \Pi^{\bullet}\sigma\Pi^{\bullet}$:

v_{ij} is a sum of monomials $\Pi^{\bullet}_{ip}\sigma_{pq}\Pi^{\bullet}_{pj}$; the only non zero ones are of the form $\Pi^{\bullet}_{i1}\Pi^{\bullet}_{2j}$ and $\Pi_{i2}\Pi_{1j}$. We eventually find $v_{1,2+n} = \Pi'^{\bullet}_{11} = v_{2+n,1}$, $v_{2+i,2+n} = \Pi'^{\bullet}_{1+i,n}$, $v_{2+n,2+j} = \Pi'^{\bullet}_{n,1+j}$, for $i,j = 1,\ldots,n'$, all other coefficients of v being null

computation of $w = \Pi^{\bullet}\sigma\Pi^{\bullet}\sigma\Pi^{\bullet} = \Pi^{\bullet}\sigma v$:

w_{ij} is a sum of monomials $\Pi^{\bullet}_{ip}\sigma_{pq}v_{qj}$; the only non zero ones are of the form $\Pi^{\bullet}_{i1}v_{2j}$ and $\Pi^{\bullet}_{i2}v_{1j}$. We eventually find $w_{2+n,2+n} = \Pi'^{\bullet}_{11}$ as the only non–zero coefficient of w. It is immediate that the fourth term of the expansion is null. Now the result $EX(\Pi^{\bullet},\sigma)$ is obtained by summing up $\Pi^{\bullet} + v + w$, then removing (or replacing by zero) the coefficients whose indices are not both > 2. We thus obtain

$M_{2+i,2+j} = \Pi'^{\bullet}_{1+i,2+j}$ $\qquad$ $M_{2+i,2+n} = \Pi'^{\bullet}_{1+i,n}$ $\qquad$ $M_{2+n,2+j} = \Pi'^{\bullet}_{n,1}$

$M_{2+n,2+n} = \Pi'^{\bullet}_{11}$ $\qquad$ all other coefficients null. If we consider M as a $n \times n$ matrix, by removing the indices 1,2, and renumbering $3,\ldots,2+n$ into $1,\ldots,n$, then we get exactly the matrix corresponding to the proof that would come from Π' by *ad hoc* exchanges. But, this proof is what is currently taken as the normalised version of Π. Hence, as long as this case (which is by the way the essential one) is considered, then our formula corresponds exactly to cut–elimination. Good start !

From now on, we shall concentate on simple cases of cuts : namely that of

$$\dfrac{\begin{array}{c}\Pi' \\ \vdash A,\ \Gamma\end{array} \qquad\qquad \begin{array}{c}\Pi'' \\ \vdash A^{\perp},\ \Delta\end{array}}{\vdash [A],\ \Gamma,\ \Delta}$$

a cut between two sequents, each of them being proved in a cut–free way ; we shall assume that the last rules (R') and (R") applied to Π' and Π'' are (up to exchange) logical rules for A or $A^{\perp}$. In that case, we have a way to replace the cut by other ones, and this process is the heart of Gentzen's proof, and of all its variants, such as our proofs in [2] or [3]. If we denote by Ξ the proof obtained by this process, our goal is to relate $EX(\Pi^{\bullet},\sigma)$ with $EX(\Xi^{\bullet},\tau)$ where τ is the partial symmetry expressing the new cuts of Ξ. We shall meet 5 cases

ii) $A = B \otimes C$, so that $A^{\perp} = B^{\perp} ⅋ C^{\perp}$; hence (up to exchanges, that we once for all ignore), Π' comes from proofs Π_1 and Π_2 of sequents $\vdash B, \Gamma_1$ and $\vdash C, \Gamma_2$, (with $\Gamma = \Gamma_1, \Gamma_2$) by means of a $\otimes$–rule, whereas Π'' comes from a proof Π_3 of $\vdash B^{\perp}, C^{\perp}, \Delta$, by a ⅋–rule. Ξ is defined by making a cut between Π_1 and Π_3, which yields Π_0, proof of $\vdash [B], C^{\perp}, \Gamma_1, \Delta$, and a second cut between Π_2 and Π_0 yields a proof Ξ of $\vdash [B,C], \Gamma, \Delta$. To see what happens, we shall assume that Γ_1, Γ_2 and Δ all consist of one formula, so that we can write a matrix, which is much more visual than indices : the matrices $\Pi_1{}^{\bullet}$, $\Pi_2{}^{\bullet}$ and $\Pi_3{}^{\bullet}$ (2 × 2, 2 × 2, 3 × 3) are given :

$$\begin{matrix} a & b \\ c & d \end{matrix} \qquad \begin{matrix} e & f \\ g & h \end{matrix} \qquad\qquad \begin{matrix} i & j & k \\ l & m & n \\ x & y & z \end{matrix}$$

Now $\Pi^{\bullet}$ is 5 × 5 :

$$\begin{matrix} pap^*+qeq^* & 0 & pb & qf & 0 \\ 0 & \begin{array}{c} pip^*+pjq^* \\ +qlp^*+qmq^* \end{array} & 0 & 0 & pk+qn \\ cp^* & 0 & d & 0 & 0 \\ gq^* & 0 & 0 & h & 0 \\ 0 & xp^*+yq^* & 0 & 0 & z \end{matrix}$$

whereas $\Xi^{\bullet}$ is 7 × 7 :

$$\begin{matrix} a & 0 & 0 & 0 & b & 0 & 0 \\ 0 & i & 0 & j & 0 & 0 & k \\ 0 & 0 & e & 0 & 0 & f & 0 \\ 0 & l & 0 & m & 0 & 0 & n \\ c & 0 & 0 & 0 & d & 0 & 0 \\ 0 & 0 & g & 0 & 0 & h & 0 \\ 0 & x & 0 & y & 0 & 0 & z \end{matrix}$$

Moreover, whereas σ exchanges 1 with 2, τ exchanges 1 with 2, and 3 with 4. Define an isomorphism Ψ from $\mathcal{M}_7(\mathcal{B}(\mathbb{H}))$ into $\mathcal{M}_5(\mathcal{B}(\mathbb{H}))$ by contracting indices 1,3 into 1 and indices 2, 4 into 2 by means of p, q in both cases, the indices 5,6,7 being renamed 3,4,5 ; then, $\Psi(\Xi^\bullet) = \Pi^\bullet$, $\Psi(\tau) = \sigma.(pp^* + qq^*)$. So $\sigma\Pi^\bullet = \sigma(pp^* + qq^*)\Pi^\bullet$ is nilpotent exactly if $\tau\Xi^\bullet$ is and in that case $EX(\Pi^\bullet,\sigma) = \Psi(EX(\Xi^\bullet,\tau))$. Now if we restrict our attention to the last 3 × 3 squares of both matrices, since Ψ is identical on this square, it makes sense to say that $EX(\Xi^\bullet,\tau) = EX(\Pi^\bullet,\sigma)$.

iii) assume that $A = \forall\alpha B$, so that $A^\perp = \exists\alpha B^\perp$; this means that Π' is obtained from a proof Π_1 of $\vdash B, \Gamma$ by means of a $\forall$–rule (so α is not free in Γ), whereas Π'' is obtained from a proof Π_2 of $\vdash B^\perp[C/\alpha], \Delta$ by means of a $\exists$–rule ; in that case, $\Pi'^\bullet = \Pi_1{}^\bullet$, $\Pi''^\bullet = \Pi_2{}^\bullet$. Ξ is defined as follows : we first form Π_3, proof of $\vdash B[C/\alpha], \Gamma$, then a cut with Π_2 yields a proof Ξ of $\vdash [B[C/\alpha]], \Gamma, \Delta$. Now there is no change in the size of matrices involved and $\sigma = \tau$; moreover, an immediate induction (which is uniformative, since all steps are trivial) shows that $\Pi_3{}^\bullet = \Pi_1{}^\bullet$, hence $EX(\Pi^\bullet,\sigma)$ exists iff $EX(\Xi^\bullet,\tau)$ does and in that case they are equal.

iv) assume that $A = {!B}$, so that $A^\perp = {?B^\perp}$; then Π' comes from a proof Π_1 of $\vdash A, \Gamma$ (with Γ of the form $?\Gamma_1$), by means of a !–rule. Assume moreover that (R'') is the contraction rule, so that Π'' comes from a proof Π_2 of $\vdash ?B^\perp, ?B^\perp, \Delta$. Ξ is obtained by first making a cut between Π_1 and Π_2, so to get rid of (the first occurence of) $?B^\perp$, yielding thus a proof Π_3 of $\vdash [!B], ?B^\perp, \Gamma, \Delta$, then another cut between Π_1 and Π_3 yields a proof Π_0 of $\vdash [!B,!B], \Gamma, \Gamma, \Delta$; finally a sequence of contractions yields a proof Ξ of $\vdash [!B,!B], \Gamma, \Delta$. In fact our formula holds only when Γ is empty (this is not as bad as it looks !). To see what happens, let us assume that Γ is empty and that Δ consists of exactly one formula, so that we can use a matricial representation : by hypothesis $\Pi_1{}^\bullet$ is a 1 × 1 matrix, and $\Pi_2{}^\bullet$ is 3 X 3 :

$$a \qquad\qquad \begin{matrix} b & c & d \\ e & f & g \\ h & i & j \end{matrix}$$

and $\Pi^\bullet$ is therefore

$$\begin{matrix} 1\otimes a & 0 & 0 \\ 0 & \begin{matrix} p'bq'^*+p'cq'^* \\ +q'ep'^*+q'fq'^* \end{matrix} & p'd+q'g \\ 0 & hp'^*+iq'^* & j \end{matrix}$$

with $p' = p \otimes 1$, $q' = q \otimes 1$; on the other hand, $\Xi^\bullet$ is the matrix :

$$\begin{matrix} 1\otimes a & 0 & 0 & 0 & 0 \\ 0 & b & 0 & c & d \\ 0 & 0 & 1\otimes a & 0 & 0 \\ 0 & e & 0 & f & g \\ 0 & h & 0 & i & j \end{matrix}$$

Consider the isomorphism Ψ from $\mathcal{M}_5(\mathcal{B}(\mathbb{H}))$ to $\mathcal{M}_3(\mathcal{B}(\mathbb{H}))$ described informally as follows : the index 5 is renamed 3, and the indices 1,3 and 2,4 are respectively contracted into 1 and 2, by means of p' and q'. It is immediate that $\Psi(\tau) = \sigma.(p'p'^* + q'q'^*)$. Now $\Psi(\Xi^\bullet)$ is almost $\Pi^\bullet$; the only difference lies in its first diagonal coefficient, which is now $(pp^* + qq^*) \otimes a$. But remark that $\sigma\Pi^\bullet$ is nilpotent iff $(1 \otimes a)(p'bp'^* + p'cq'^* + q'ep'^* + q'fq'^*)$ is, and this last *expression is nilpotent iff* $((p'p'^* + q'q'^*) \otimes a)(p'bp'^* + p'cq'^* + q'ep'^* + q'fq'^*)$ is in turn nilpotent, which is another way to say that $\sigma(p'p'^* + q'q'^*)\Psi(\Xi^\bullet)$ is nilpotent. It is also easy to see that, in case of nilpotency, $EX(\Pi^\bullet,\sigma) = EX(\Psi(\Xi^\bullet),\sigma.(p'p'^* + q'q'^*))$, so $EX(\Pi^\bullet,\sigma) = \Psi(EX(\Xi^\bullet,\tau))$, but if we restrict to the last 2×2 squares on which Ψ is identical, we get $EX(\Pi^\bullet,\sigma) = EX(\Xi^\bullet,\tau)$.

v) as in iv), but assume that Π' comes by dereliction from a proof Π_2 of $\vdash B^\perp, \Delta$; in that case, Π is defined as the result of cutting Π_1 with Π_2, so that to get a proof of $\vdash [B], \Gamma, \Delta$. Here again we shall work with the extra hypothesis that Γ is empty, and illustrate the proof in the particular case where Δ consists of one formula. Assume that $\Pi_1{}^\bullet$ and $\Pi_2{}^\bullet$ are respectively

$$a \qquad\qquad\qquad \begin{matrix} b & c \\ d & e \end{matrix}$$

Then $\Pi^\bullet$ is obviously

$$\begin{matrix} 1 \otimes a & 0 & 0 \\ 0 & pbp^* & pc \\ 0 & dp^* & e \end{matrix}$$

whereas $\Xi^\bullet$ is

$$\begin{matrix} a & 0 & 0 \\ 0 & b & c \\ 0 & d & e \end{matrix}$$

and $\sigma = \tau$. The nilpotency of $\tau\Xi^\bullet$ means that ba is nilpotent. on the other hand, the nilpotency of $\Pi^\bullet$ is the same as the nilpotency of $(pbp^*)(1 \otimes a) = pbap^*$, using the fact that $p^*(1 \otimes a) = ap^*$. But $p(ba)p^*$ is nilpotent iff ba is nilpotent. Then $\tau\Xi^\bullet$ and $\sigma\Pi^\bullet$ are silmutaneously nilpotent. If one of them is nilpotent, then the unique coefficient of $EX(\Xi^\bullet,\tau)$ is $e + dac^* + dabac^* + dababac^* + \ldots$, whereas the unique coefficient of $EX(\Pi^\bullet,\sigma)$ is $e + dp^*(1 \otimes a)pc^* + dp^*(1 \otimes a)pbp^*(1 \otimes a)pc^* + \ldots$ which is equal, using $p^*(1 \otimes a)p = a$, to $e + dac^* + dabac^* + \ldots$, i.e. once more $EX(\Xi^\bullet,\tau) = EX(\Pi^\bullet,\sigma)$.

vi) as in iv) but Π'' is obtained from a proof Π_2 of $\vdash \Delta$ by means of a weakening. Ξ is defined as follows : since all formulas of Γ begin with ?, simply apply weakenings to Π_2, so that to get a cut-free proof of $\vdash \Gamma, \Delta$. Here again we shall assume that Γ is empty and that Δ consists of one formula. Hence $\Pi_1{}^\bullet$ and $\Xi^\bullet$ have both dimension 1, and $\tau = 0$. Let a and b be their respective coefficients. Then $\Xi^\bullet$ is a 1×1 matrix consisting of b, whereas $\Pi^\bullet$ is 3×3 :

$$\begin{array}{ccc} a & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & b \end{array}$$

It is immediate (school computation) that $\Pi^\bullet\sigma\Pi^\bullet = 0$, hence if we ignore the first two rows/columns, $\mathrm{EX}(\Pi^\bullet,\sigma)$ is equal to b, hence to $\Xi^\bullet$; but since $\tau = 0$, $\mathrm{EX}(\Xi^\bullet,\tau) = \Xi^\bullet$, and the property holds in that case too.

vii) there is yet another interesting test case : consider a proof Π of $\vdash [!B], \Delta, !C$ ending with a cut between $\vdash !B$ (proved by Π', which comes from a proof Π_1 of $\vdash B$ by a !–rule) and $\vdash ?B^\perp, ?\Delta, !C$ (proved by Π'', which comes from a proof Π_2 of $\vdash ?B^\perp, ?\Delta, C$ by a !–rule). Here Ξ is classically defined as the result of first cutting Π' with Π_2 so that to get a proof Π_3 of $\vdash [!B], ?\Delta, C$ to which a !–rule is then applied so that to get a proof Ξ of $\vdash [!B], \Delta, !C$. As usual we shall assume that Δ consists of one formula, so that we start with the following matrices for $\Pi_1{}^\bullet$ and $\Pi_2{}^\bullet$:

$$a \qquad\qquad \begin{array}{ccc} b & c & d \\ e & f & g \\ h & i & j \end{array}$$

so that $\Pi^\bullet$ is :

$$\begin{array}{cccc} 1\otimes a & 0 & 0 & 0 \\ 0 & t(1\otimes b)t^* & t(1\otimes c)t^* & t(1\otimes d) \\ 0 & t(1\otimes e)t^* & t(1\otimes f)t^* & t(1\otimes g) \\ 0 & (1\otimes h)t^* & (1\otimes i)t^* & 1\otimes j \end{array}$$

and $\Xi^\bullet$ is :

$$\begin{array}{cccc} t(1\otimes(1\otimes a))t^* & 0 & 0 & 0 \\ 0 & t(1\otimes b)t^* & t(1\otimes c)t^* & t(1\otimes d) \\ 0 & t(1\otimes e)t^* & t(1\otimes f)t^* & t(1\otimes g) \\ 0 & (1\otimes h)t^* & (1\otimes i)t^* & 1\otimes j \end{array}$$

moreover, $\sigma = \tau$. But $t(1\otimes(1\otimes a))t^* = (1\otimes 1)\otimes a = 1\otimes a$, and so $\Pi^\bullet = \Xi^\bullet$.

II.3. the main theorem : statement and discussion

THEOREM 1 :

i) if $(\Pi^\bullet,\sigma)$ is the interpretation of a proof Π of a sequent $\vdash [\Delta], \Gamma$, then $\sigma\Pi^\bullet$ is nilpotent.

ii) if Γ does not use the symbols "?" or "$\exists$", and Ξ is any cut–free proof of $\vdash \Gamma$ obtained from Π by using standard Gentzen reduction steps *in any order*, then $\mathrm{EX}(\Pi^\bullet,\sigma) = \Xi^\bullet$. (As usual this makes sense with the abuse consisting in removing from $\mathrm{EX}(\Pi^\bullet,\sigma)$ the rows/columns corresponding to Δ, which are filled with null coefficients).

The theorem must be discussed, since there is an important restriction as to the form of Γ. This restriction is due to the fact that in many of the cases yet considered, we had to require that the context of a !–rule is void. However let us remark the following points :

1– The nilpotency of $\sigma\Pi^\bullet$ is established without restriction, hence $\mathrm{EX}(\Pi^\bullet,\sigma)$ always makes sense, although it may be very far from $\Xi^\bullet$.

2– Typical formulas involving "?" are the usual data types, e.g. tally integers, whose type, in linear logic is $int = \forall\alpha(?(\alpha\otimes\alpha^\perp) ⅋ (\alpha^\perp ⅋ \alpha))$. Therefore, any program ending with an output of type *int* is not covered by the theorem ! First observe that this limitation does not apply when the result is an intermediate one, since we can only meet problems with final results. But we have here to remember that the result has to be displayed somewhere, by means of a *side effect* (screen, printer, noise, etc.). These side effects are not part of logic, which does not mean that they cannot be modelised via functional analysis (but the operators involved may lose some property, e.g. maybe not longer hermitian). Here we shall content ourselves with a very primitive method, which is enough for the crude purpose of showing that integer computations are covered by our theorem.

Define a new boolean type $truth = \forall\alpha((\alpha\otimes\alpha) \multimap (\alpha\otimes\alpha))$; the truth values *true* and *false* are respectively defined as the two basic cut–free proofs of *truth* whose associated operators (1×1 matrices) are respectively : $p^2p^*q^* + qp(p^*)^2 + pq(q^*)^2 + q^2q^*p^*$, and $p^2(q^*)^2 + q^2(p^*)^2 + pqp^*q^* + qpq^*p^*$. Moreover, there is a function of definition by cases of type $\forall\alpha(!\alpha \multimap (!\alpha \multimap (truth \multimap \alpha)))$, coming form the proof of the sequent $\vdash ?\alpha^\perp, ?\alpha^\perp, truth^\perp, \alpha$ whose matrix is :

$$\begin{array}{cccc}
0 & 0 & (p^*)^2 & 0 \\
0 & 0 & q^*p^* & 0 \\
p^2 & pq & 0 & qp^2 \\
0 & 0 & (p^*)^2q^* & 0
\end{array}$$

and which is written in traditional syntax as :

$$\dfrac{\dfrac{\dfrac{\vdash ?\alpha^\perp,\ !\alpha \qquad \vdash ?\alpha^\perp,\ !\alpha}{\vdash ?\alpha^\perp,\ ?\alpha^\perp,\ !\alpha\otimes !\alpha} \qquad \dfrac{\dfrac{\dfrac{\vdash \alpha^\perp;\ \alpha}{\vdash ?\alpha^\perp,\ \alpha}}{\vdash ?\alpha^\perp,\ ?\alpha^\perp,\ \alpha}}{\vdash ?\alpha^\perp ⅋ ?\alpha^\perp,\ \alpha}}{\vdash ?\alpha^\perp;\ ?\alpha^\perp,\ (!\alpha\otimes !\alpha)\otimes(?\alpha^\perp ⅋ \alpha^\perp),\ \alpha}}{\vdash ?\alpha^\perp,\ ?\alpha^\perp,\ \exists\alpha((!\alpha\otimes !\alpha)\otimes(?\alpha^\perp ⅋ \alpha^\perp)),\ \alpha}$$

This functional proof corresponds to an IF...THEN...ELSE... instruction : the IF part is the 3rd row/column of the matrix, the parts THEN and ELSE occupy indices 1 and 2, whereas the result is given by index 4.

All this shows that *truth* is a perfectly legitimate boolean type. Now, everything that comes as a result can be eventually seen as a sequents of bits of fixed length, hence can be represented by a program of type some (maybe quite big) tensor power of *truth*, say *truth*$^{\mathrm{n}}$, and since such a type is free from "?", we are done. Theoretically speaking, it is possible to construct an object of type *int* ⊸ *int* transforming the (badly shaped) result of an execution of integer type into a well–shaved integer : see chapter IV.

3– To understand why our approach strongly differs from the standard syntactical "symbol pushing", remember that symbol pushing mainly rests on so–called *β–conversion*, $(\lambda xt)u \mapsto t[u/x]$, in which the global entity u is duplicated or erased, and moved. If we see this as a physical process, and if we imagine that such a term occupies a very big space, this operation can only be performed by some omnipotent God (or in more concrete terms, we postulate a global time for the operation). This is why traditional syntax has been of very little help for parallelism, since as soon as we need a global time, we need some synchronizing device, and such devices may be more costly than the improvement due to parallelisation. Our approach refuses any kind of global time, i.e. we can only make local moves, with no need for synchronisation. Sometimes these moves are not yet possible, but some other move will make them possible. Now, our conception of time is roughly the same as the one coming from relativity theory, namely causality : a move μ is *before* a move ν when ν has contributed to ν. So most of moves will be temporally unrelated, and a snapshot of our kind of execution will therefore show something quite far from what syntax usually yields. The difference is so big that we have not been able to state clearly what happens in the general case (restriction on "?"). In fact from a purely syntactical viewpoint, the execution makes "mistakes", but it is precisely because of these "mistakes" that we can free ourselves from the need of a universal time !

4– A last word to clarify what we mean by causality : a move p^* (or q^*) is a "pop" move, and is by definition impossible, unless the stack is not empty, i.e. p^* makes sense only in the contexts p^*p (= 1) or p^*q (= 0). When, in the development of $EX(\Pi^\bullet,\sigma)$, we find a monomial $\alpha p^*\beta$, the part β has *first* to be brought to the form $p\gamma$ or $q\gamma$ (one has to take this as a possible definition of *first*). The reader will argue that booleans are then not doable at all, since they involve a lot of p^* and q^* ; but we must use a side effect : to know the value of a boolean β, compute βq^2, and depending on the value (qp or p^2) found, deduce the truth value of β (*true* or *false*). By the way, the meaning of removing $pp^* + qq^* = 1$ is to prevent 1 to occur as a sum of monomials. Concretely, this equation says that if we pop and push again what we poped, then we get the same thing : what a physical nonsense ! (think of an empty stack). In fact we are simply refusing very strongly so–called η–conversion : when we remove $pp^* + qq^* = 1$, then the two proofs of $\vdash (A \otimes B)^\perp, A \otimes B$ given by the identity or given by the identies on the components and logical rules are distinct operators (both are antidiagonal 2 × 2 matrices) ; but the former has coefficients equal to 1, whereas the latter has coefficients equal to $pp^* + qq^*$.

II.4. some rudiments of theory

We first start with some lemmas concerning EX :

LEMMA 2 :

Let μ, μ' be two monomials of $\mathcal{B}(\mathbb{H})$ (using p,q,t,s,1,$\otimes$,*), without scalar coefficients ; then they are partial isometries, and their initial projectors commute.

PROOF : here we come back to the concrete interpretation of our operations in $\mathcal{B}(\mathbb{H})$: all the partial isometries which are our primitives are induced by partial functions from $\mathbb{N}$ to $\mathbb{N}$, namely, when s is one of these partial isometries, there is a partial function s^- from $\mathbb{N}$ to $\mathbb{N}$ such that $s(\Sigma\, \zeta_n.b^n) = \Sigma\, \zeta_n.b^{s^-(n)}$; moreover, we have that $(ss')^- = s^-s'^-$, and when s, s' both come from partial functions, so does $s \otimes s'$, namely $(s \otimes s')^-(<n,m>) = <s^-(n),s^-(m)>$. In fact the initial (and final) projectors of these isometries are all obtained from some idempotent partial function and this is why they commute. QED

LEMMA 3 :

Let u be any partial isometry of $\mathcal{B}(\mathbb{H})$ of the form $\Pi^\bullet$ or σ (here $\mathcal{M}_n(\mathcal{B}(\mathbb{H}))$ has been for a moment replaced by $\mathcal{B}(\mathbb{H})$, using monomials in p and q to contract the indices), and let π be a projector belonging to the Boolean algebra $\mathcal{B}$ generated by initial projectors of monomials (see lemma 2). Then there is in $\mathcal{B}$ a smallest projector $u[\pi]$ such that : $u[\pi].u = u\pi$.

PROOF : looking carefully at the construction of, say, $\Pi^\bullet$, we discover that it is a sum of monomial partial isometries θ_i with pairwise disjoint domains, pairwise disjoint codomains. This means that $\Pi^\bullet$ also comes from a partial function from $\mathbb{N}$ to $\mathbb{N}$. In particular, if $u[\pi]$ is defined as $u\pi u^*$, then $u\pi u^*$ is a projector in $\mathcal{B}$, and $u\pi u^* u = uu^* u\pi = u\pi$. QED

LEMMA 4 :

If $EX(\Pi^\bullet,\sigma)$ exists, then it is a partial symmetry, induced by a partial function from $\mathbb{N}$ to $\mathbb{N}$.

PROOF : let $\pi = 1-\sigma^2$, and consider the monomials $\mu_n = \pi\Pi^\bullet(\sigma\Pi^\bullet)^n\pi$. Then $\mu_n = \mu_n{}^*$, moreover, $\mu_n{}^2 = \pi\Pi^\bullet(\sigma\Pi^\bullet)^n\pi(\Pi^\bullet\sigma)^n\Pi^\bullet\pi = \pi\tau_n\Pi^\bullet(\sigma\Pi^\bullet)^n(\Pi^\bullet\sigma)^n\Pi^\bullet\pi = \pi\tau_n\pi = \pi\tau_n$ from some projector π' of $\mathcal{B}$, hence is a projector. So μ_n is a partial symmetry. The initial projector $\pi\tau_n$ of μ_n are such that $\pi\tau_n.\pi\tau_m = 0$ for $n \neq m$: if $m = n+p+1$, then $\tau_n = \Pi^\bullet(\sigma\Pi^\bullet)^m[(\sigma\Pi^\bullet)^{p+1}[\pi]]$, whereas $\tau_m = \Pi^\bullet(\sigma\Pi^\bullet)^m[\pi]$, but since $(\sigma\Pi^\bullet)^{p+1}[\pi]$ begins with σ, then its product with π is null, hence $\tau_n\tau_n = 0$. Finally the μ_i's are partial symmetries with pairwise disjoint supports, hence their sum is of the same nature. It is also induced by a partial function from $\mathbb{N}$ to $\mathbb{N}$. QED

LEMMA 5 (*associativity of cut*) :

we assume that Π is a proof of a sequent $\vdash [\Gamma,\Delta], \Lambda$, and that σ and τ are the partial symmetries corresponding to Γ and Δ ; then

i) $\Pi^\bullet(\sigma+\tau)$ is nilpotent iff $\sigma\Pi^\bullet$ and $\tau EX(\Pi^\bullet,\sigma)$ are nilpotent

ii) in that case $\mathrm{EX}(\Pi^\bullet,\sigma+\tau) = \mathrm{EX}(\mathrm{EX}(\Pi^\bullet,\sigma),\tau)$.

PROOF : let $\theta = \sigma+\tau$, $\pi = 1-\theta^2$, and introduce, for $n \in \mathbb{N}$ and $q \subset \{1,\dots,n\}$ the monomial $\mu_{n,q} = u\rho_1 u\rho_2 u\dots u\rho_n u$, with $u = \Pi^\bullet$ and $\rho_i = \tau$ for $i \in q$, $\rho_i = \sigma$ otherwise. When n is fixed, the 2^n monomials $\mu_{n,q}$ are partial isometries with pairwise disjoint domains, pairwise disjoint codomains : in particular, since $u(\theta u)^n$ is the sum of these 2^n monomials, θu is nilpotent iff for some n all $\mu_{n,q}$ are null. Since $u(\sigma u)^n = \mu_{n,0}$ (with 0 the void set), σu is nilpotent iff some $\mu_{n,0}$ is null. Now if σu is nilpotent $\mathrm{EX}(\Pi^\bullet,\sigma)(\tau\mathrm{EX}(\Pi^\bullet,\sigma))^n$ splits into a finite sum of monomials

$(1-\sigma^2).\mu_{n,i}.\tau(1-\sigma^2).\mu_{m,j}\tau(1-\sigma^2)\dots.\tau(1-\sigma^2)\mu_{p,k}.(1-\sigma^2) = \pi.\mu_{n,i}.\tau.\mu_{m,j}.\tau\dots.\tau.\mu_{p,k}.\pi$ (with n times τ) and these monomials are partial isometries of the form $\pi\mu_{a,b}\pi$. They have pairwise disjoint domains, pairwise disjoint codomains, hence if $\tau\mathrm{EX}(\Pi^\bullet,\sigma)$ is nilpotent too, there is an integer n such that all monomials $\pi.\mu_{n,q}.\pi$ are null : but then any monomial $u\rho_1\mu_{n,q}\rho_{n+2}u$ is null using $\rho_i\pi = \rho_i$, which means that for some m all monomials $\mu_{m,p}$ are null. Conversely the existence of such a m entails in a trivial way nilpotency of σu and $\tau\mathrm{EX}(u,\sigma)$. Putting things together, we just proved claim i). Now $\mathrm{EX}(u,\theta)$ is easily shown to be the sum of all monomials $\pi\mu_{n,q}\pi$, whereas $\mathrm{EX}(\mathrm{EX}(u,\sigma),\tau)$ is easily shown to be the sum of all monomials

$(1-\tau^2).\pi.\mu_{n,i}.\tau.\mu_{m,j}.\tau\dots.\tau.\mu_{p,k}.\pi.(1-\tau^2)$; but these monomials are exactly the monomials $\pi\mu_{n,q}\pi$, using $(1-\tau^2)\pi = \pi$, and we established ii). QED

We shall now develop in a C*–algebraic framework the exact analogue of our proof of normalisation of [2] (more precisely of its adaptation to the case of linear logic), as worked out in [3].

DEFINITION 1 :

The *message space* $\mathcal{B}$ is the Boolean algebra generated by initial (or final) projectors of monomials as in lemma 2. An *observable* operator is a partial isometry a such that : for all π in the Boolean algebra $\mathcal{B}$ used in the previous lemmas, $a^*\pi a$ and $a\pi a^*$ belong to $\mathcal{B}$. $\mathcal{K}$ will denote the set of observable operators. Composition of two observables is an observable, sum of two observables with disjoint domains and disjoint codomains, is an observable.

Let a and b be two observable operators ; then a is said to be *orthogonal* to b exactly when ab is nilpotent (notation $a \perp b$).

Orthogonality is obviously symmetric, moreover 0 is orthogonal to everything.

DEFINITION 2 :

Given a subset X of $\mathcal{K}$, define $X^\perp = \{a\ ;\ \forall b\ (b \in X \mapsto a \perp b)\}$. A *type* is any subset X of $\mathcal{K}$ equal to its biorthogonal, i.e. $X = X^{\perp\perp}$. Since $Y^\perp = Y^{\perp\perp\perp}$ for any Y, X is a type iff X is equal to $Y^\perp$ for some $X \subset K$; clearly 0 belongs to any type.

DEFINITION 3 :

Let A be a formula of the language of linear logic under study ; let α be a sequence of free variables including all free variables of A, and let X be sequence of types of the same length. We define the type $\theta A[X/\alpha]$ by induction on A as follows :

i) if A is α_i (one of the variables of α), then $\theta A[X/\alpha]$ is X_i (the ith type in X)

ii) if A is $\alpha_i{}^\perp$, then $\theta A[X/\alpha]$ is $X_i{}^\perp$, (the ith type in X)

iii) if A is $B \otimes C$, then consider the set Y made with all operators $pap^* + qbq^*$, when a and b vary through $\theta B[X/\alpha]$ and $\theta C[X/\alpha]$ respectively. Define $\theta A[X/\alpha] = Y^{\perp\perp}$.

iv) if A is B ⅋ C, then consider the set Y made with all operators $pap^* + qbq^*$, when a and b vary through $(\theta B[X/\alpha])^\perp$ and $(\theta C[X/\alpha])^\perp$ respectively. Define $\theta A[X/\alpha] = Y^\perp$.

v) if A is !B, then consider the set Y made with all operators $1 \otimes a$ when a varies through $\theta B[X/\alpha]$. Define $\theta A[X/\alpha] = Y^{\perp\perp}$.

vi) if A is ?B, then consider the set Y made with all operators $1 \otimes a$ when a varies through $(\theta B[X/\alpha])^\perp$. Define $\theta A[X/\alpha] = Y^\perp$.

vii) if A is $\forall\beta B$, then consider the set Y which is the intersection of all types $\theta B[X,T/\alpha,\beta]$, when T varies through all types. Define $\theta A[X/\alpha] = Y$. (Here $Y^{\perp\perp} = Y$, since $(.)^\perp$ commutes with intersection.)

viii) if A is $\exists\beta B$, then consider the set Y which is the intersection of all types $(\theta B[X,T/\alpha,\beta])^\perp$, when T varies through all types. Define $\theta A[X/\alpha] = Y^\perp$.

LEMMA 6 : (*substitution lemma*, see e.g. [2])

$$\theta A[X,\theta B[X/\alpha]/\alpha,\beta] = \theta(A[B/\beta])[X/\alpha]$$

PROOF : the lemma states that, if we compute θA with $\theta B[X/\alpha]$ as the type associated with β, or if we compute directly θC, where C is $A[B/\beta]$, we find the same result. The lemma is proved by a straightforward induction on A. It uses strongly the fact that $\theta(B[X/\alpha]^\perp) = (\theta B[X/\alpha])^\perp$, which is an obvious consequence of the definitions. QED

DEFINITION 4 :

Let $\vdash \Gamma = \vdash A_1,\ldots,A_n$ be a sequent, let α be a sequence of free variables including all free variables of A, and let X be sequence of types of the same length ; a *datum* of type $\theta\Gamma[X/\alpha]$ is a n × n matrix $M = (\alpha_{ij})$ such that :

i) it is a partial isometry

ii) all α_{ij} are in $\mathcal{K}$ (so to speak, M is an *observable matrix*)

iii) for any $\beta_1 \in \theta A_1[X/\alpha]^\perp, \ldots, \beta_n \in \theta A_n[X/\alpha]^\perp$, the matrix $(\beta_i\alpha_{ij})$ is nilpotent.

An *algorithm* of type $\theta\Gamma[X/\alpha]$ is a matrix $M = (\alpha_{ij})$ of dimension 2m+n × 2m+n, for some integer m, enjoying conditions i) and ii) above, and such that, if σ denotes the partial symmetry exchanging indices 1 and 2, 3 and 4, ..., 2m−1 and 2m, then

iv) σM is nilpotent

v) the n × n matrix obtained from $EX(M,\sigma) = (1-\sigma^2)M(1-\sigma M)^{-1}(1-\sigma^2)$ by removing the first 2m rows/columns, is a datum of type $\theta\Gamma[X/\alpha]$.

THEOREM 2 :

If Π is a proof of Γ, then $\Pi^\bullet$ is an algorithm of type $\theta\Gamma[X/\alpha]$.

PROOF : by induction on the proof Π : to simplify notations, we shall make the proof in the case α is empty

i) if Π is an axiom, so that $\Gamma = A, A^\perp$; then $\Pi^\bullet$ is the antidiagonal matrix ; take any a, b respectively in θA and $\theta A^\perp$; we want to show that the matrix M :

$$\begin{matrix} 0 & a \\ b & 0 \end{matrix} \quad \text{is nilpotent. But its square is} \quad \begin{matrix} ab & 0 \\ 0 & ba \end{matrix}$$

and ab and ba are nilpotent, so $M^{2n} = 0$ for some n.

Before going on, let us prove a very useful lemma :

LEMMA 7 :

Given a $n \times n$ matrix $M = (\alpha_{ij})$, with $n \neq 0$, and $a \in \mathcal{B}(\mathbb{H})$, define CUT(a,M) as the matrix (β_{ij}) : $\beta_{1j} = a\alpha_{1j}$, $\beta_{ij} = \alpha_{ij}$ when $i \neq 1$ and let π_1 be the projector of the first row/column. Then an observable matrix M belongs to $\theta(A,\Gamma)$ iff for any a in $\theta A^\perp$, $a\alpha_{11}$ is nilpotent, and the $n \times n$ matrix (extracted from) $ex(CUT(a,M)) = (1-\pi_1).M.(1-\pi_1 CUT(a,M))^{-1}.(1-\pi_1)$, is in $\theta\Gamma$.

PROOF : assume for simplicity that n = 2, so that Γ is a formula B and M is

$$\begin{matrix} \alpha & \beta \\ \gamma & \delta \end{matrix}$$

$M \in \theta(A,B)$ iff for any a and b in $\theta A^\perp$ and $\theta B^\perp$, the matrix P[a,b]

$$\begin{matrix} a\alpha & a\beta \\ b\gamma & b\delta \end{matrix}$$

is nilpotent. But introduce N[a,b] as

$$\begin{matrix} a & 0 & 0 & 0 \\ 0 & \alpha & 0 & \beta \\ 0 & 0 & b & 0 \\ 0 & \gamma & 0 & \delta \end{matrix}$$

and the partial symmetries σ (which exchanges 1 and 2) and τ (which exchanges 3 and 4). It is immediate that P[a,b] is nilpotent iff $(\sigma+\tau)N[a,b]$ is nilpotent. Now, assume that $M \in \theta(A,B)$. Then $(\sigma+\tau)N[a,b]$ is nilpotent, and by associativity of cut, $\sigma N[a,b]$ and $\tau.EX(N(a,b),\sigma)$ are nilpotent. Now, if we forget the first 2 indices, $EX(N[a,b],\sigma)$ can be written as

$$\begin{matrix} b & 0 \\ 0 & \delta \end{matrix}$$

for a certain δ independant of b. The fact that $\tau.EX(N(a,b),\sigma)$ is nilpotent is the same as the nilpotency of $b\delta$, hence $\delta \in \theta B$. (The fact that $\theta B^\perp$ is non void has been very heavily used). But this can be restated as the nilpotency of the first diagonal coefficient of CUT(a,M), and δ

is easily shown to be ex(CUT(a,M)). The other direction is proved in the same way. QED

ii) assume that Π is proof of $\vdash [\Delta', \Delta'', A], \Gamma', \Gamma''$ coming from two proofs Π' and Π'' of respectively $\vdash [\Delta'], \Gamma', A$ and $\vdash [\Delta''], A^{\perp}, \Gamma''$, using a cut–rule. By hypothesis $\Pi'^{\bullet}$ and $\Pi''^{\bullet}$ respectively belong to $\theta(A,\Gamma')$ and $\theta(A^{\perp},\Gamma'')$. We assume for simplicity of notations that both Γ' and Γ'' consist of single formulas, B' and B''. We first investigate the case where both Δ' and Δ'' are void. By lemma 7, given any b' in $\theta B^{\perp}$, then $\mathrm{CUT}(b',\Pi'^{\bullet})$ is nilpotent and $a = \mathrm{ex}(\mathrm{CUT}(b',\Pi'^{\bullet})) \in \theta A$. Then $\mathrm{CUT}(a,\Pi''^{\bullet})$ is nilpotent and $\mathrm{ex}(\mathrm{CUT}(a,\Pi''^{\bullet})) \in \theta B''$. Now an easy use of associativity of cut yield that $\sigma\Pi^{\bullet}$ is nilpotent and the first diagonal coefficient of $\mathrm{CUT}(b,\mathrm{EX}(\Pi^{\bullet},\sigma))$ nilpotent, and $\mathrm{ex}(\mathrm{CUT}(b,\mathrm{EX}(\Pi^{\bullet},\sigma)) = \mathrm{ex}(\mathrm{CUT}(a,\Pi''^{\bullet})) \in \theta B''$. Using lemma 7 once more, we get that $\mathrm{EX}(\Pi^{\bullet},\sigma) \in \theta B''$.

It remains to consider the case where Δ' and/or Δ'' is non void. But by an easy use of associativity of cut, it can be reduced to the case just treated. By the way, from now on, we shall ignore the cuts in the proofs, since it is always possible to first eliminate them by EX, then prove the statement and then apply associativity of cut

iii) if Π ends with an exchange rule, then there is very little to do.

iv) if Π ends with a $\otimes$–rule, applied to two subproofs Π' of $\vdash \Gamma', A$ and Π'' of $\vdash \Gamma'', B$. Then, we can easily, by uses of lemmas 5 and 7, reduce the problem to the problem of showing that, given a in θA, b in θB, $pap^* + qaq^* \in \theta(A \otimes B)$, which is immediate.

v) if Π ends with a $\parr$–rule, applied to a subproof Π' of $\vdash \Gamma, A, B$, then the problem is easily reduced to showing that, whenever

$$\begin{matrix}\alpha & \beta \\ \gamma & \delta\end{matrix}$$

belongs to $\theta(A,B)$, then $\epsilon = p\alpha p^* + p\beta q^* + q\gamma p^* + q\delta q^* \in \theta(A \parr B)$. But by hypothesis we have that, for any a in $\theta A^{\perp}$, any b in $\theta B^{\perp}$,

$$\begin{matrix}a\alpha & a\beta \\ b\gamma & b\delta\end{matrix}$$

is nilpotent, hence for any such a and b $(pap^* + qbq^*).\epsilon$ is nilpotent, which shows that $\epsilon \in \theta(A \parr B)$.

vi) if Π ends with a $\forall$–rule, applied to a subproof Π' of $\vdash \Gamma, A$, then the problem is easily reduced to showing that, whenever α belongs to $\theta A[X/\alpha]$ for all X, then it belongs also to $\theta\forall\alpha A$. This is immediate.

vii) if Π ends with a $\exists$–rule, applied to a subproof Π' of $\vdash \Gamma, A[B/\alpha]$, then the problem is easily reduced to showing that, whenever α belongs to $\theta A[B/\alpha]$, then it belongs to $\theta\exists\alpha A$. But (lemma 6) $\theta A[B/\alpha]$ is of the form $\theta A[X/\alpha]$, hence the property reduces to the trivial fact that the union of all $\theta A[X/\alpha]$ is included in $\theta\exists\alpha A$.

viii) if Π ends with a !–rule, applied to a subproof Π' of $\vdash \Gamma, A$, with Γ of the form $?\Gamma'$, then we must be cautious ; for simplicity we consider the case where Γ consists of the only formula B. Assume that $\Pi'^{\bullet}$ and $\Pi^{\bullet}$ are respectively

$$\begin{matrix} \alpha & \beta \\ \gamma & \delta \end{matrix} \qquad\qquad \begin{matrix} t(1 \otimes \alpha)t^* & t(1 \otimes \alpha) \\ (1 \otimes \gamma)t^* & 1 \otimes \delta \end{matrix}$$

take any b in $\theta B^{\perp}$, hence $1 \otimes b \in \theta!(B^{\perp})$. The hypothesis says that M

$$\begin{matrix} (1 \otimes b)\alpha & (1 \otimes b)\beta \\ \gamma & \delta \end{matrix}$$

is nilpotent and that $ex(M) \in \theta A$. Consider N

$$\begin{matrix} (1 \otimes b)t(1 \otimes \alpha)t^* & (1 \otimes b)t(1 \otimes \beta) \\ (1 \otimes \gamma)t^* & 1 \otimes \delta \end{matrix}$$

since $1 \otimes b = (1 \otimes 1) \otimes b$, we get $1 \otimes b = t(1 \otimes (1 \otimes b))t^*$, so N is equal to

$$\begin{matrix} t(1 \otimes ((1 \otimes b)\alpha))t^* & t(1 \otimes ((1 \otimes b)\beta)) \\ (1 \otimes \gamma)t^* & 1 \otimes \delta \end{matrix}$$

Consider the isomorphism Ψ of $\mathcal{M}_2(\mathcal{B}(\mathbb{H}))$ into iself which transforms $\Pi'^{\bullet}$ into $\Pi^{\bullet}$; then it is clear that $\Psi(M) = N$, so N is nilpotent, and $ex(N) = \Psi(ex(M)) = ex(M) \in \theta A$: the hypothesis of lemma 7 holds for any element $1 \otimes b$ of $\theta!(B^{\perp})$, and those are dense in this type w.r.t. biorthogonality... so the property holds (easy analogue of lemma 7) for all objects of $\theta!(B^{\perp})$.

ix) if Π ends with a weakening rule, applied to a subproof Π' of $\vdash \Gamma$, then the problem is easily reduced to showing that $0 \in \theta?A$, which is obvious.

x) if Π ends with a dereliction rule, applied to a subproof Π' of $\vdash \Gamma$, A, then the problem is easily reduced to showing that, whenever α belongs to θA, then $p\alpha p^*$ belongs to $\theta?A$. For this take a in $\theta A^{\perp}$, so that $a\alpha$ is nilpotent. Now $(1 \otimes a)p\alpha p^*$ behaves like $p^*(1 \otimes a)p\alpha$ w.r.t. nilpotency, but $p^*(1 \otimes a)p = a$, hence $p\alpha p^* \perp (1 \otimes a)$, and we are done.

xi) if Π ends with a contraction rule, applied to a subproof Π' of $\vdash \Gamma$, ?A, ?A then the problem is easily reduced to showing that, whenever a matrix

$$\begin{matrix} \alpha & \beta \\ \gamma & \delta \end{matrix}$$

belongs to $\theta(?A,?A)$, then

$\epsilon = (p \otimes 1)\alpha(p^* \otimes 1) + (p \otimes 1)\beta(q^* \otimes 1) + (q \otimes 1)\gamma(p^* \otimes 1) + (q \otimes 1)\delta(q^* \otimes 1) \in \theta(?A)$. The hypothesis yields that

$$\begin{matrix} (1 \otimes a)\alpha & (1 \otimes a)\beta \\ (1 \otimes a)\gamma & (1 \otimes a)\delta \end{matrix}$$

is nilpotent for any a in $\theta A^{\perp}$, and this can be rewritten, using contraction of indices as $0 = \epsilon.((p \otimes 1)(1 \otimes a)(p^* \otimes 1) + (q \otimes 1)(1 \otimes a)(q^* \otimes 1)) = \epsilon.(pp^*+qq^*) \otimes a =$ $\epsilon.((pp^* + qq^*) \otimes 1).(1 \otimes a)$, but $\epsilon.((pp^* + qq^*) \otimes 1) = \epsilon$, so $\epsilon.(1 \otimes a) = 0$, and we are done.

This ends the proof of theorem 2.

II.5. the main theorem : proof

First observe that theorem 2 contains part i) of theorem 1, namely the nilpotency of $\sigma\Pi^{\bullet}$. We have to prove part ii), and we shall indeed content ourselves with a sketch. First we shall use the *proof–nets* of [3], and the basic result is that, if two proofs yield the same multiplicative proof–net, then their interpretations are the same. We shall not prove it, since, so to speak, the functional analysis model is a generalisation of proof–nets, and the proof can only be boring and uninformative. Then, proof–nets have been generalised in [5] so that boxes for universal quantifiers are removed. In that case, the fact that our interpretation depends only on the underlying proof–net is even more obvious, since the interpretation of quantifiers is particularly trivial. So we are left with a proof containing only !–boxes. Now, due to the form of the result (without $\exists$ or ?), we know that all possible normalisation strategies lead to the same thing, since the result does not contain any rule for "?". Now, it is an easy exercise to show that the following normalisation strategy can be followed : use only the contractions (in the terminology of [3], chapter 4) (AC), ($\otimes$/⅋–SC), (!/W?–SC), (!/D?–SC), (!/C?–SC), ($\forall/\exists$–SC), (!–CC) ; (by ($\forall/\exists$–SC), we mean its obvious adaptation to the case where there is no longer $\forall$–boxes). Moreover in all the !–contractions, the !–box is without context. The justification is as follows : due to strong normalisation, it is possible to use all contractions which are not !–contractions, up to the moment where all cuts are on formulas !A, $?A^{\perp}$, the part !A being the main door of a !–box. If the result is not cut–free, then among all these boxes, there is one which is without context (this comes from the hypothesis on the shape of the conclusion), let say the one leading to $!A_0$; then we look at the last rule yielding $?A_0^{\perp}$; this rule is not an axiom (because of (AC)), hence must be either a weakening, a dereliction, a contraction, or a !–box : in this case, $?A_0$ is a side door of this other box. All these cases are handled respectively by (!/W?–SC), (!/D?–SC), (!/C?–SC) and (!–CC). Hence we can make a new normalisation step, and we got one step closer to the final result (strong normalisation). Now, the rules we have been considering have been examined in II.2. as respectively cases i), ii), vi), v), iv), iii) and vii), and in each case, we were able to show that EX is invariant. However, the sketch is not rigourous, since in the cases we were considering, the cut under elimination was the last rule. But observe that, so to speak, EX commutes with logical rules. The typical example is when our proof–net Π is a !–box proving $\vdash [\Delta], \Gamma$, coming from a proof–net Π' proving $\vdash [\Delta], \Gamma', A$, with Γ' beginning with "?", and Γ is Γ', !A. Now, if we assume for simplicity that both Δ and Γ consist of one formula, $\Pi'^{\bullet}$ has a matrix :

$$\begin{matrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ w & x & y & z \end{matrix}$$

whereas $\Pi^\bullet$ has the matrix

$$\begin{matrix} t(1\otimes a)t^* & t(1\otimes b)t^* & t(1\otimes c)t^* & t(1\otimes d) \\ t(1\otimes e)t^* & t(1\otimes f)t^* & t(1\otimes g)t^* & t(1\otimes h) \\ t(1\otimes i)t^* & t(1\otimes j)t^* & t(1\otimes k)t^* & t(1\otimes l) \\ (1\otimes w)t^* & (1\otimes x)t^* & (1\otimes y)t^* & 1\otimes z \end{matrix}$$

Consider the isomorphism Ψ of $\mathcal{M}_4$ into itself which precisely changes matrices in this way ; it is immediate that $\Psi(\sigma) = \sigma$, hence $\sigma\Pi^\bullet$ is nilpotent iff $\sigma\Pi'^\bullet$ is, and in that case

$\mathrm{EX}(\Pi^\bullet,\sigma) = \Psi(\mathrm{EX}(\Pi'^\bullet,\sigma))$. But $\Psi(\mathrm{EX}(\Pi'^\bullet,\sigma))$ is precisely the result of the interpretation of the box "!" applied to $\mathrm{EX}(\Pi'^\bullet,\sigma)$ and we are done. There is still another difficulty, namely that in II.2., we have only considered situations where the premises of the cut are cut–free. But this is an easy application of associativity of cut, and we are done.

This ends our proof, or rather our sketch of proof.

III. the C*-algebra Λ*

Our first intention was to introduce Λ* by means of an axiomatic description ; but this has two essential drawbacks : first we shall have to worry about completeness, namely making sure that some equation is not missing, and if this kind of work is difficult, it is of very little interest ; second, we would like to find a very concrete physical meaning to our operations, since we have not in mind that the execution could run through something as ugly as implementing the syntax of Λ*. The difficulty is that in traditional set–theoretic terms, partial isometries are monsters. But this is maybe because the primitives of set–theory are wrong... In fact, quite surprisingly, we can modelise our operations in a nice finitary way. The idea is that is to use (continuous) step functions on the Cantor continuum, the values being G, D, or I. To do that, we have just to push the dichotomy to a certain point. Once the values are given, dichotomising further will change nothing. Our primitives will be the basic moves on such step functions. In set–theory, any space in which such moves make sense is infinite ; but can we dream of something more finite than that ? The Cantor continuum can of course be alternatively described by means of finite binary trees, but then we have to spent a lot of energy on changing these trees and this is why we stick to the very basic intuition of continuity.

DEFINITION 5 :

i) the *diods* are defined as the two formal objects G, D and I, equipped with a partial composition relation : $G^2 = G$, $D^2 = D$, $I^2 = I$, $GI = IG = G$, $DI = ID = D$, GD and DG undefined. So to speak G is the left diod, D is the right one, and I is neutral.

ii) the Cantor space is the space Ω of infinite sequences of 0 and 1, i.e. $\{0,1\}^{\mathbb{N}}$; concatenation of sequences is denoted by #, and we shall write $0 \# s$ instead of $(0) \# s$ etc. ; equipped with the product topology, the Cantor space is a compact space. Its basic open sets are of the form $O_q = \{q \# s \,;\, s \in \Omega\}$, where q is any finite sequence ; moreover, due to compactness, every clopen set is a finite union of sets O_q.

iii) a *pure message* is a continuous map from Ω to the (discrete) diod space. A pure message can therefore be seen as a finite tree of 0 and 1 whose leaves are labelled with diods. Such a representation is not unique, but among all possible representation of the same pure message, there is one with the smallest possible tree.

iv) if M and M' are pure messages, one may (try to) define their product as their pointwise product, which, due to the fact that GD and DG are not defined, is not always defined.

v) if M and M' are pure messages, one may define their tensorisation $M'' = M \otimes M'$ by $M''(0 \# s) = M(s)$, $M''(1 \# s) = M'(s)$.

vi) an *atomic* message is a message M such that $M(s) \neq I$ for all $s \in \Omega$. To say that M is atomic means that for any M', either MM' = M or MM' is undefined.

DEFINITION 6 :

i) a *pure observable* (M',φ,M) consists in the following data

pure messages M (source message) and M' (target message)

a continuous function φ which is a bijection between iM and iM', where iM is defined as $\{s \,;\, M(s) = i\}$, etc., enjoying the property :

(P) there exist finite sequences $a_1,\ldots,a_n$ (resp. $b_1,\ldots,b_n$) of 0 and 1, such that the domain (resp. the codomain) of φ is the disjoint union of the O_{a_i} (resp. O_{b_i}) and for $i = 1,\ldots,n$ and $s \in \Omega$, we have $\varphi(a_i \# s) = b_i \# s$.

ii) another way to represent pure observables is to consider 3–tuples (U,φ,T), where T and U are finite binary trees made of 0 and 1, with leaves labelled with G, D or I, and φ is a bijection between the set of I–leaves of T and the set of I–leaves of U.

Of course such a representation is never unique.

iii) let (M',φ,M) be a pure observable, and let N be a pure message ; we (try to) define the product $(M',\varphi,M)N$ as follows :

if MN is undefined, then $(M',\varphi,M)N$ is undefined

otherwise, define φ' as the restriction of φ to i(MN), and the message M'' by $M''(s) = M'(s)$ except for $s \in iM' - rg(\varphi')$, in which case, $M''(s) = M(\varphi^{-1}(s))$.

iv) in particular, when N is atomic, then φ' is always void, and M'' is atomic ; to each pure observable (M',φ',M) is therefore associated a partial function f from atomic messages to atomic messages, which satifisfies : f(N) defined iff $(M',\varphi,M)N$ defined and in this case

$(M',\varphi,M)N = (f(N),\emptyset,MN)$. Two pure observables with the same induced function on atomic messages must obviously be equal.
v) the product of two pure observables $(M',\varphi,M)(N',\psi,N)$ is defined exactly when MN' is ; in that case, we form the products $(M',\varphi,M)N' = (M'',\varphi',MN')$ and $M(N',\psi,N) = (MN',\psi',N'')$, and we define $(M',\varphi,M)(N',\psi,N) = (M'',\varphi'\psi',N'')$. The definition used $M(N',\psi,N)$, whose definition can be imagined easily. The product is defined in such a way that it corresponds to to the composition of the associated partial functions. The product is undefined exactly when the composition of the associated partial functions would we nowhere defined. We can think of pure messages as a particular case of pure observables, mamely by representing M by (M,φ,M), where φ is the identity on the I–leaves of M. This identification is compatible with the two definitions of product involving messages. As an observable, a message has an underlying partial function, which is idempotent, i.e. can be seen as a set of atomic messages.
vi) the tensor product $(M',\varphi,M) \otimes (N',\psi,N)$ of pure observables is defined as $(M' \otimes N',\varphi \otimes \psi,M \otimes N)$, with $(\varphi \otimes \psi)(0 \# s) = 0 \# \varphi(s)$, $(\varphi \otimes \psi)(1 \# s) = 1 \# \psi(s)$. This definition extends the one already given for messages. As to the induced partial functions on atomic messages, it is immediate that if f, g are associated with two pure observables, then h defined by $h(M \otimes M') = f(M) \otimes g(M')$ is associated with their tensor product.
vii) given a pure observable (M',φ,M), one can define its adjoint as (M,φ^{-1},M'), so that pure messages are self–adjoint. In terms of induced functions, adjunction is just inversion. Here we have to remark that the partial function associated with a pure observable is a bijection between its domain and its codomain.

It would now be the room to check endless properties ; the best is to remark that we came as close as possible to our intuitions of chapter 1. Consider a bijection $n, m \mapsto \langle n,m\rangle$ between $\mathbb{N} \times \mathbb{N}$ and $\mathbb{N}$; it does not cost much to assume that $\langle 0,0\rangle = 0$, $\langle 1,1\rangle = 1$ and that all integers are generated from 0 and 1 by means of $\langle .,.\rangle$. Then to each atomic message M, we can associate a unique integer $\$M$, namely $\$G = 0$, $\$D = 1$, $\$(M \otimes N) = \langle \$M,\$N\rangle$ (by G we mean the message constantly equal to G etc.). So atomic messages are in bijection with $\mathbb{N}$, and we can say that to any pure observable is associated with a partial bijection from $\mathbb{N}$ to $\mathbb{N}$, *hence with a partial isometry of ℓ^2*. When we translate our definitions of product, tensor, adjoint into these partial isometries, we just find the notions we introduced in chapter 1 ; when the product is not defined, then the associated partial isometry is 0. This remark is enough to understand what has been done so far.

Among pure observables, let us mention
i) $t = (I,\varphi,I)$: the initial and the final message are both I (constant function with value I) and $\varphi(0 \# s) = (0,0) \# s$, $\varphi((1,0) \# s) = (0,1) \# s$, $\varphi((1,1) \# s) = 1 \# s$.
ii) $s = (I,\psi,I)$: $\psi(0 \# s) = 1 \# s$, $\psi(1 \# s) = 0 \# s$ (sorry for the mismatch of letters ; more

seriously, remember that we never used s, and that its eventual removal from Λ^* is not excluded).

iii) $p = (G \otimes I, \theta, I)$, $q = (D \otimes I, \theta, I)$ with $\theta(s) = 1 \# s$.

For instance it is immediate that , if M is an atomic message, then $pM = G \otimes M$, $qM = D \otimes M$, hence p and q induce the functions $n \mapsto \langle 0,n \rangle$ and $n \mapsto \langle 1,n \rangle$, as ewpected.

DEFINITION 7 :

i) the space $p\Lambda^*$ (p for "pre") is defined as the set of all formal finite linear combinations $\Sigma\, \lambda_i . P_i$, where the P_i are pure observables and the λ_i are complex coefficents. If by convention, we decide that $PP' = 0$ (the null linear combination), when their product was not defined, there is no difficulty in extending by (bi) linearity the operations of product, tensor and adjoint, so that, together with the obvious notion of sum we get a C*–algebra, without yet a norm. The object I plays the role of the unit.

ii) since there is an obvious interpretation of our linear combinations as operators of ℓ^2, there is at least one possible C*–norm on $p\Lambda^*$, namely the norm of the associated concrete operators. We didn't look serously at the marginal question of determining whether or no this norm is the only posible one, but there are two simple facts, namely first that there is, as usual a greatest C*–norm, and second, that there is a smallest one, namely the norm N coming from the ℓ^2 representation. (see below) Then we norm $p\Lambda^*$ by means of N, and complete so to get Λ^*.

LEMMA 8 :

N is the smallest C*–semi–norm on $p\Lambda^*$.

PROOF : let (m_n) be the enumeration of atomic messages which has just been introduced ; if we define the pure observables b^n as $b^n = (m_n, \emptyset, m_0)$, and if N' is any C*–semi–norm on $p\Lambda^*$, then $N'(b^n) = \sqrt{N'(b^0)}$, hence with $n = 0$, $N'(b^0) = 0$ or 1 ; but $m_0 = 0$ would induce $m_0 \otimes I = 0$, and then we would get $pp^* = 0$, hence $p^*p = 0$, but $p^*p = I = 1$. So $N'(b^n) = N'(b^0) = 1$. In particular, $N'(\Sigma\, \lambda_i . b^i)^2 = N'(\Sigma\, \lambda_i . b^i . \Sigma\, \overline{\lambda}_i . b^{i*}) = \Sigma\, \lambda_i . \overline{\lambda}_i$. Now, if u is any operator of $p\Lambda^*$, then $u(\Sigma\, \lambda_i . b^i) = \Sigma\, \mu_i . b^i$, and we pass from the sequence (λ_i) to the sequence (μ_i) by means of the operator of ℓ^2 that we associated with u. Hence $N(u) = \sup\{N(\Sigma\, \mu_i . b^i) \,;\, N(\Sigma\, \lambda_i . b^i) = 1\} = \sup\{N'(\Sigma\, \mu_i . b^i) \,;\, N'(\Sigma\, \lambda_i . b^i) = 1\}$
$\leq \sup(N'(uv) \,;\, N'(v) = 1) = N'(u)$. QED

The space Λ^* is incredibly concrete. To finish with our definitions,

DEFINITION 8 :

i) a message is any finite sum of pure messages whose pairwise products are null.

ii) an observable is any sum of pure observables whose domains are pairwise incompatible, and whose codomains are pairwise incompatible.

Observe that in the boolean algebra genrated by pure messages, we only consider those projectors which have positive coefficients ; in the same way, ii) is stronger than saying that an observable operates on messages. It is easy to see that we were in fact using this refinement in chapter 2. By the way, let us remind the reader that there is no way to say that 1 is the sum of all atomic messages (such a denumerable sum cannot converge in norm, and only makes sense w.r.t. some kind of weak topology, typical of so–called Von Neumann algebras, which are another world).

Finally, the only question at this moment is to decide whether or not our kind of computation is feasible. But the result is a sum of monomials ; we already discussed this issue in II.3. and reduced it morally (more refined studies should be made later) to the case of a boolean result, and we remarked that it was enough to compute the expression $\mathrm{EX}(\Pi^\bullet,\sigma)q^2$, and depending on the result pq or p^2, the answer was found. Now, our expression is a finite sum of observables, and therefore all summands but one are null. This is one of the main reasons for deleting the equation $pp^*+qq^* = 1$, which would have induced possible synthesis of observables. So one of the monomials is 1, whereas the others are zero. In particular, if we do the execution from right to left, starting with q^2, and then making the development :
q^2, $\Pi^\bullet(1-\sigma^2)q^2$, $(1-\sigma^2)\Pi^\bullet(1-\sigma^2)q^2 + \Pi^\bullet\sigma\Pi^\bullet(1-\sigma^2)q^2$,
$(1-\sigma^2)\Pi^\bullet(1-\sigma^2)q^2 + (1-\sigma^2)\Pi^\bullet\sigma\Pi^\bullet(1-\sigma^2)q^2 + \Pi^\bullet\sigma\Pi^\bullet\sigma\Pi^\bullet(1-\sigma^2)q^2$, etc., then each step consists of exactly one monomial. This is because everything is isometric, and in order to get two monomials, we should therefore have a choice between multiplying on the left by, say, αp^* and βq^* at some moment. But then the only way to recover a monomial pq or p^2 at the end is that the right part is of the form $p\gamma$ or $q\gamma$ etc. So there is at least a way to execute. This way is by no means the best, and part C– of the program should be concerned with the study of efficiency, as long as this remains a mathematical problem. From the moment on, we obtained what we were longing for, namely :

a finitary dynamics free from syntax.

IV. example : tally integers

Integers in tally representation are interesting from an abstract viewpoint because they behave to some extent like sequences, but are simpler. We shall introduce their representations in Λ^*, and demonstrate the dynamic power of (EX) on simple functions like iterators.

DEFINITION 9 :

i) the type *int* is defined as $\forall\alpha.(!(\alpha \multimap \alpha) \multimap (\alpha \multimap \alpha))$.

ii) the integer N_0 is defined as $qp(q^*)^2 + q^2p^*q^*$, for $k \neq 0$, the integer N_k is defined as $\Sigma\,(pp'_{i+1}pq(p^*)^2p'_i{}^*p^* + \Sigma\,pp'_ip^2q^*p^*p'_{i+1}{}^*p^* + + qp(p^*)^2p'_k{}^*p^* + pp'_kp^2p^*q^* + q^2q^*p^*p'_0{}^*p^* + pp'_0pq(q^*)^2$, where $p'_0,\dots,p'_k$ are pure observables (to be defined below) of the form $r_i \otimes 1$ such that $p'_i{}^*p'_j = 0$ for $i \neq j$, and the sums are taken oven $0,\dots,k-1$. The r_i's are not yet defined, since they come from contractions inside the k–th canonical proof of the formula *int*, and therefore, there are several possible choices, depending on the order of the contraction. Their actual values will be obtained by means of the successor function.

DEFINITION 10 :

The successor function is the 2 × 2 matrix coming from the proof of $\vdash int^\perp, int$, which precisely defines the successor. We therefore get :

$$\begin{array}{ll} 0 & p(p^* \otimes 1)p^* + q^2(p^*)^2(q^* \otimes 1)p^* + qpp^*q^* \\ p(p \otimes 1)p^* + p(q \otimes 1)p^2(q^*)^2 + qpp^*q^* & p(q \otimes 1)pq(q^*)^2 + q^2q^*p^*(q^* \otimes 1)p^* \end{array}$$

LEMMA 9 :

It is possible to chose the p'_i in such a way that the successor of N_k is N_{k+1}.

PROOF : let's write the successor as

$$\begin{array}{ll} 0 & B \\ B^* & C \end{array}$$

then the successor of N_k is $EX(M,\sigma)$, where σ exchanges the first two indices of the matrix M :

$$\begin{array}{lll} N_k & 0 & 0 \\ 0 & 0 & B \\ 0 & B^* & C \end{array}$$

and $EX(M,\sigma)$ is easily shown to be equal to $C + B^*N_kB$, i.e. to :
$p(q \otimes 1)pq(q^*)^2 + q^2q^*p^*(q^* \otimes 1)p^* + \Sigma\,p(p \otimes 1)p'_{i+1}pq(p^*)^2p'_i{}^*(p^* \otimes 1)p^* + \Sigma\,p(p \otimes 1)p'_ip^2q^*p^*p'_{i+1}{}^*(p^* \otimes 1)p^* + p(p \otimes 1)p'_0pq(p^*)^2(q^* \otimes 1)p^* + p(q \otimes 1)p^2q^*p^*p'_0{}^*(p^* \otimes 1)p^* + p(p \otimes 1)p'_kp^2p^*q^* + qp(p^*)^2p'_k{}^*(p^* \otimes 1)p^*$. Now define $r_0 = q$, $r_{i+1} = pr_i$. Then our expression rewrites as :
$pp'_0pq(q^*)^2 + q^2q^*p^*p'_0{}^*p^* + \Sigma\,pp'_{i+2}pq(p^*)^2p'_{i+1}{}^*p^* + \Sigma\,pp'_{i+1}p^2q^*p^*p'_{i+2}{}^*p^* + pp'_1pq(p^*)^2p'_0{}^*p^* + pp'_0p^2q^*p^*p'_1{}^*p^* + pp'_{k+1}p^2p^*q^* + qp(p^*)^2p'_{k+1}{}^*p^*$,
i.e. is equal to N_{k+1}. QED

DEFINITION 11 :

Take any type A, m an object of type A, M (a square matrix) of type $\vdash A^\perp, A$; then the iterator It(m,M) is a square matrix deduced from the logical proof corresponding to iteration (and which is of type $\vdash int^\perp, A$) whose coefficients are, provided M is

$$\begin{array}{cc} a & b \\ c & d \end{array}$$

$$\begin{array}{cc} p(1 \otimes (pap^* + pbq^* + qcp^* + qdq^*))p^* + qpmp^*q^* & q^2 \\ (q^*)^2 & 0 \end{array}$$

Typical examples are :

i) with $A = \mathit{int}$, $m = N_0$, $M =$ successor : It(m,M) is called the *shaving* functional

ii) with $A = !\mathit{int}$, $m = 1 \otimes N_0$, M being the matrix

$$\begin{array}{cc} 0 & t(1 \otimes B) \\ (1 \otimes B^*)t^* & 1 \otimes C \end{array}$$

deduced from the successor matrix by means of the !–rule : It(m,M) is called the *linearising* functional.

THEOREM 3 :

Let A, M and m be as in definition 11 ; then the result of applying It(m,M) to the integer N_k is exactly $M^k(m)$, if we use the notation M(m) to denote the result of applying M to m (i.e. making a cut on A, then executing), etc.

PROOF : add a new symbol A to the formulas of linear logic, together with axioms (i.e. syntactical boxes) $\vdash A$ (a denumerable family, the k–th being interpreted by $M^k(m)$) and $\vdash A^\perp, A$ (to be interpreted by M). Then we can adapt the machinery of theorem 1 to this case : syntactically speaking, a cut between the k–th axiom $\vdash A$ and our new axiom $\vdash A^\perp, A$ being reduced to the k+1–th axiom $\vdash A$. Then everything works, since the result is of atomic type. The result of cutting the k–th axiom with the new identity axiom is

$$\begin{array}{ccc} M^k(m) & 0 & 0 \\ 0 & a & b \\ 0 & c & d \end{array}$$

and after execution, the result is $M^{k+1}(m)$, i.e. the only new feature of the syntax is interpreted in the right way. QED

REMARKS :

i) the result still holds (associativity of cut) when we don't plug in N_k, but the operator obtained by execution of a program of type *int*, whose syntactical result is the k–th integer. This simply comes from the fact that semantically speaking, the final result of type A, must be $M^k(m)$.

ii) in particular, if we feed the shaving functional with a (semantic) integer coming from the execution of (the semantic translation of) a proof of type *int*, then we find a N_k.

iii) it is easily checked that the linearisation functional does two things : first shave the input into some N_k, then replace this N_k by $1 \otimes N_k$. The name of this functional comes from the fact that it can be used to replace a general function defined on integers (type $\vdash ?int^{\perp}$, B for some B) by a linear one (type $\vdash int$, B).

We shall here suggest an exercise, namely compute the shaving of N_k ; there is nothing new here, but toying a little with this objects, which, on the whole are very concrete ones, may be illuminating :

the question is to compute EX(P,σ), where σ exchanges the first two lines of P :

$$\begin{array}{ccc} N_k & 0 & 0 \\ 0 & p(1 \otimes (pBq^* + qB^*p^* + qCq^*))p^* + qpN_0p^*q^* & q^2 \\ 0 & (q^*)^2 & 0 \end{array}$$

The concrete computation must rather be done on a blackboard than inside a mathematical paper ; good luck !

V. two ideological themes

V.1. communication by nilpotency

Our basic claim is that nilpotency expresses the absence of loops inside the information flow. To illustrate this, we have the mathematical development already made, but also a toy example, namely the *paradigm of the dictionary* :

consider a (technical) dictionary, where a set of words, $w_1,\ldots,w_k$ is explained, by use of current language. Concretely, some terms are defined using other ones, e.g. $k = 4$, and w_1 is defined in terms of w_2 and w_3, w_2 is defined just by means of current language, the definition of w_3 involves w_2, whereas w_4 is defined by means of w_1 and w_3. Now, our dictionary works like a mini–program, namely, to understand w_4, we are reduced to w_1 and w_3, which are in turn reduced to w_2 and w_3, which are in turn reduced to w_2, and w_2 is reduced to nothing so that w_4 is eventually understood. Now we write a 4 × 4 matrix expressing the dependency ($\alpha_{ij} = 1$ when w_i is defined in terms of w_j, $= 0$ otherwise), then we obtain :

$$\begin{array}{cccc} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{array}$$

Now, to say that our dictionary is sound means that we never loop when trying to find the meaning of a word. In other terms, it is possible to relabel our words as w'_i, so that each w'_i is defined using only the previous ones. Typically, in our example, relabel 1,2,3,4 as 3,1,2,4, and the matrix becomes :

$$\begin{matrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{matrix}$$

and we have obtain a new matrix which is obviously strictly triangular. Such a matrix is nilpotent, and this nilpotency is independant of the numbering of rows/columns. Observe that very often the relabelling is far from being unique, hence the formulation of nilpotency which does not involve any sequentialization is much more manageable. To some extent, it is what we have been doing, except that instead of coefficients 1, we were using partial isometries.

To illustrate how these kind of considerations could be of possible logical interest, consider the old question of Henkin quantifiers. These quantifiers express, like all quantifiers, dependencies of existential variables over universal variables. But here $Qxx'yy'.A$ means that one can simultaneously find the value of y in terms of x and the value of y' in terms of x'. Traditional (sequential) quantifiers cannot express this kind of dependency, e.g. if we write $\forall x\exists y\forall x'\exists y'A$, then y' is supposed to depend on both x and x', and therefore, sequential logic (the only kind of extant logic) will therefore fail to express this subtle situation. We know by abstract model–theoretic considerations (essentially Lindström's theorem) that there is no way to fix this defect without destroying the main properties of classical logic. With linear logic, the situation becomes slightly better, since we are building the logic on a symmetry

proofs/counterproofs

and the arguments of model theory do not apply. The only problem is now to find out what could be the abstract dual form of dependency (coHenkin quantifier) $Q^{\perp}yy'xx'$ of y and y' over x and x' such that, when doing normalisation, the actual value of the terms is eventually found. Two quantifiers R and S (one expressing the dependency of the y_j over the x_i, the other expressing the dependency of the x_i over the y_j) will be said to be *orthogonal* when, on the sole basis of the dependencies, the values of the x_i and the y_j will eventually be found. A quantifier is a way to speak of functions, without carrying them, only remembering the dependencies : this is why the logical approach should consider the question independently of the concrete functions involved. Now, if we form a dictionary with the x_i and the y_j as entries, we are reduced to the previous problem, namely to look for a nilpotent dependency matrix. Coming back to Henkin, $Q^{\perp}$ will be a set of possible dependencies of the y over the x, which can be expressed by matrices

$$\begin{matrix} a & b \\ c & d \end{matrix}$$

(columns : y,y', rows : x,x') ; the Henkin dependency being expressed by

$$\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}$$

(columns : x,x', rows : y,y'), and we are trying to find out which possible values 0 and 1 for a,b,c,d will make the square matrix

$$\begin{matrix} 0 & 0 & a & b \\ 0 & 0 & c & d \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{matrix}$$

nilpotent. It is immediate that the only possibilities are :

$a = b = d = 0 \qquad c = 1$

$a = c = d = 0 \qquad b = 1$

$a = b = c = d = 0$

in other terms, $Q^{\perp}yy'xx'A$ says that y depends on x', y' depends on x, and one of these dependencies is fake. One may ask what is the biorthogonal of Q, and it is easy to verify that Q is its own biorthogonal. Then this indicates the way of handling parallel dependencies in logic. This should eventually be of interest for those questions related to parallel execution, where usual logics force us to stick to sequential dependencies.

V.2. the absence of understanding, or genericity

This has been one of the main underlying themes of our modelisation. The basic idea is that typing is not only a way of making sure that a program will eventually terminate, will produce the right answer : it is a measure of the *degree of genericity* of the unit we have typed. Second order typing possesses two ways of abstraction, the one coming from exponential connectives, the one coming from quantifiers. When we pass from a proof of $\vdash A[B/\alpha]$ to a proof of $\vdash \exists\alpha A$, B is irreversibly lost ; we can argue that, anyway, our formulation does nothing in that case, but observe that replacing B by an isomorphic type B' would change nothing to nilpotency (and nothing to the result of an execution with a cut on $\exists\alpha A$). In the same way, the rules of dereliction and contraction involve partial isometries, whose choice is random : typically, if we introduce $p_0 = p$, $p_1 = q$, $p_k = (p_i \otimes 1)p_j$ when $k = \langle i,j \rangle$, then we can use any p_k (not always the same) in any dereliction, and any two pairs $p_i \otimes 1$, $p_j \otimes 1$ (with $i \neq j$) in any contraction without any need to use always the same pair. This is because the dual operations ($\forall\alpha$, !) do not at all know in advance what kind of –say– duplication isometry will be used, and is prepared therefore to treat them generically, without understanding them. This genericity does not extend to the other operations : typically, we have decided once for all that p and q should be used in this order for all $\otimes$ and all ⅋, and if we were changing our mind, then we would have to make a new uniform choice. So to speak, the typing, which basically indicates to which depth we are analysing things, tells us that anything beyond this degree of analysis, is "up to isomorphism". This genericity is ultimately the greatest warranty for modularity : in a cut (communication) between A and $A^{\perp}$ there is a finite common language, namely the operations which correspond to the logical decomposition of the type A, and all other instructions are internal to the two participants. (In real life we communicate using words, but this limited interface does not fully expresses our thougths). Beyond the logical level, every protagonist is unable to do anything but generic operations on the instructions send by its opponent, and this is what we call

communication without understanding

This philosophical (or ideological) point has been too important during the genesis of our program not to have been developed here. In later works to come, this thesis will be formulated in precise mathematical terms. Just for the moment, let us remark the following, namely, that if we accept understanding, then there is a 1–step predecessor function on the integer type, namely the 2 × 2 matrix

$$\begin{matrix} 0 & B^* + q^2(q^*)^2 \\ B + q^2(q^{*2}) & 0 \end{matrix}$$

where B has been defined in lemma 9. Let us call it P ; if we compute in this way the predecessor of N_k, then the result is $(B + q^2(q^*)^2)N_k(B^* + q^2(q^*)^2)$; then, when $k = 0$, we obtain N_0 ; when $k = i + 1$, so that N_k is $C + B^*N_iB$, we find $BB^*N_iBB^*$, which is equal to N_i. The computation is 1–step since the development of the formula (EX) is cut after the second monomial. But this computation is possible only because we know the combinations (e.g. $p \otimes 1$, $q \otimes 1$) used in the contraction rules ; if we were making other choices, then the result would be inpredictible, for instance it might lead to 0, i.e. erase everything. However, one may argue that would be posssible to work with *int* as a new atomic type with two primitives, zero and successor, the successor function being no longer defined up to some isomorphism (i.e. $p \otimes 1$ and $q \otimes 1$ rigidified), and w.r.t. this new primitive type, there would be no possibility of mistake.

BIBLIOGRAPHY

[1] Cuntz, J. *Simple C*—algebras generated by isometries,* Comm. Math. Phys. 57, pp. 173–185, 1977.

[2] Girard, J.Y. *Une extension de l'interprétation de Gödel à l'analyse et son application à l'élimination des coupures dans l'analyse et la théorie des types*, Proc. 2nd Scand. Log. Symp., ed. Fenstad, pp. 63–92, North–Holland 1971.

[3] Girard, J.Y. *Linear Logic,* Theor. Comp. Sc. 50, pp. 1–102, 1987.

[4] Girard, J.Y. *Multiplicatives*, to appear in Rendic. Semin. Univ. Polit. Torino, 1988.

[5] Girard, J.Y. *Quantifiers in linear logic*, to appear in the proceedings of the Congress SILFS, held in Cesena, January 1987.

[6] Girard, J.Y. *Towards a geometry of interaction*, to appear in AMS volume dedicated to the congress "category theory and computer science", held in Boulder, June 1987.

[7] Kelly, G.M. *On Mac Lane's conditions for coherence of natural associativities*, Jour. Algebra 1, pp. 397–402, 1964.