Free comonoids in models of linear logic

Nicolas Tabareau

PPS (Preuve Programmation Système)
PARIS, FRANCE

Geocal, 2006

joint work with Paul-André Melliès

work in progress

Motivations : String diagrams

$$A \otimes B$$

- Physic : Tensor Algebra of particles
- Semantic : Tensor Algebra of resources

Hint :

- GoI and linear logic
- Knot theory, quantum groups
- Geometry of concurrency (homotopy and interferences)

Motivations (2)

Cooking up a game model of PCF (= λ-calculus + fixpoint)

Key ingredients :

- A model of linear logic (smcc + finite products + bang)
- A trace operator

Shortcut : starting from a compact closed category with free comonoids

Outline

1. *Trace and references*
2. Conway games
3. Duplication and free comonoid

Trace and references

From an object-programming point of view, a variable is essentially :

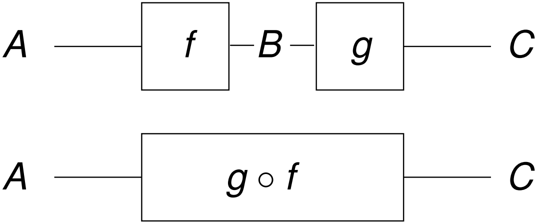
- an object in which you can write (input)
- an object that you can read (output)

How to merge these two notions ?

Idea : input = output

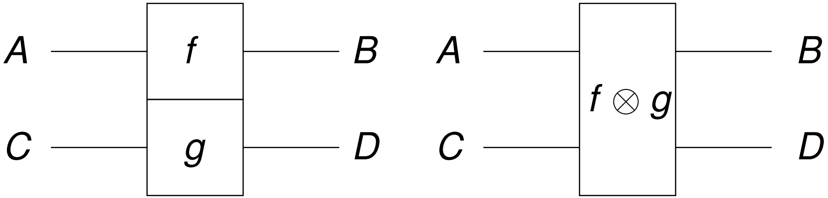
String diagrams in monoidal categories

Category : Graph whose arrows compose horizontally



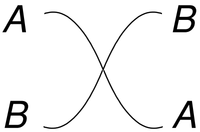
String diagrams in monoidal categories (2)

Monoidal category : Category whose arrows compose vertically

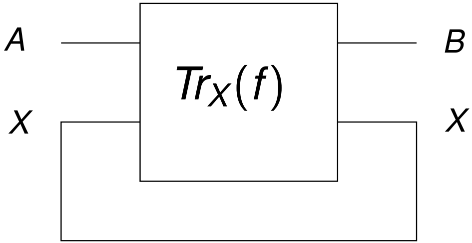


String diagrams in monoidal categories (3)

Symmetry :



A symmetric monoidal category with feedback



Traced Categories [Joyal et al., 1996]

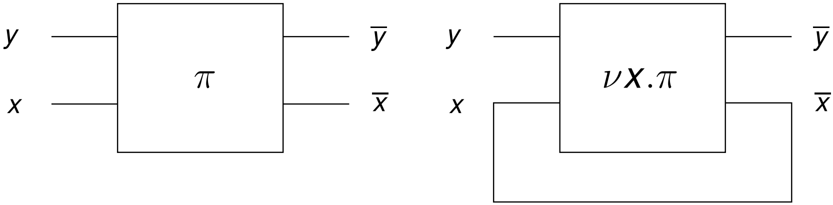
A symmetric monoidal category with feedback

$$Tr_X : \quad \frac{A \otimes X \quad \longrightarrow \quad B \otimes X}{A \quad \longrightarrow \quad B}$$

+ axioms

Trace and π-calculus

This notion has already been used by Milner to describe the operator of restriction of π-calculus, the so called ν



Model the scope extrusion of π-calculus

Axioms of a trace operator

A symmetric monoidal traced category [Joyal et al., 1996] is a SMC $(\mathcal{C}\text{at}, \otimes, I, s)$ together with a family of functions

$$Tr_X : \quad \frac{X \otimes A \longrightarrow X \otimes B}{A \longrightarrow B}$$

satisfying the following axioms :

- **Naturality :** $Tr_X(id_X \otimes g; f; id_X \otimes h) = g; Tr_X f; h$
- **Strength :** $Tr_X(f \otimes g) = Tr_X f \otimes g$
- **Symmetry sliding :** $Tr_X(Tr_Y(f; c_{XY} \otimes id_B)) = Tr_Y(Tr_X(c_{XY} \otimes id_A; f))$
- **Yanking :** $Tr_X(c_{XX}) = 1_X$

Axioms of a trace operator

A symmetric monoidal traced category [Joyal et al., 1996] is a SMC $(\mathcal{C}\text{at}, \otimes, I, s)$ together with a family of functions

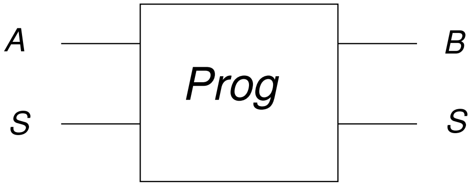
$$Tr_X : \quad \frac{X \otimes A \longrightarrow X \otimes B}{A \longrightarrow B}$$

satisfying the following axioms :

Strength $\sim$ Scope extrusion

$$Tr_X(f \otimes g) = Tr_X f \otimes g \quad \leftrightsquigarrow \quad \nu.x\ (f \mid g) \cong \nu.x\ f \mid g$$

Trace and linear state monad



$$S \otimes A \to S \otimes B$$

$$A \to S \multimap (S \otimes B)$$

restriction = localisation of global variable

Compact closed category

A compact closed category is a monoidal category where the tensor is autodual, ie. satisfies :

$$\frac{(A \otimes B) \to C}{B \to A^* \otimes C}$$

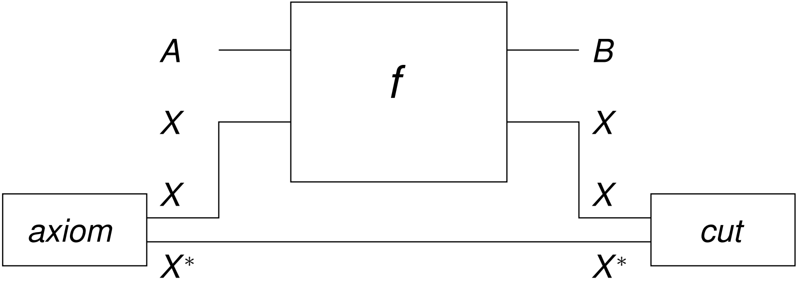
From a linear logic point of view, that's a logic where

$$\otimes = \parr$$

Central notion in knot theory (a category of ”rubber” maps)

Compact closed category and trace

Every compact closed category admits a canonical trace



$$\text{axiom} : 1 \longrightarrow X \otimes X^*$$
$$\text{cut} : X \otimes X^* \longrightarrow 1$$

Outline

Definition [Joyal, 1977]

A Conway game $A = (V_A, E_A, \lambda_A)$ is given by :

- a rooted oriented graph (V_A, E_A) with root $\star_A$
- a map $\lambda_A : E_A \to \{-1, +1\}$ specifying the polarity of a move

As usual -1 means opponent and $+1$ player

Plays and paths

We note $x \rightarrow y$ when $(x, y) \in E_A$

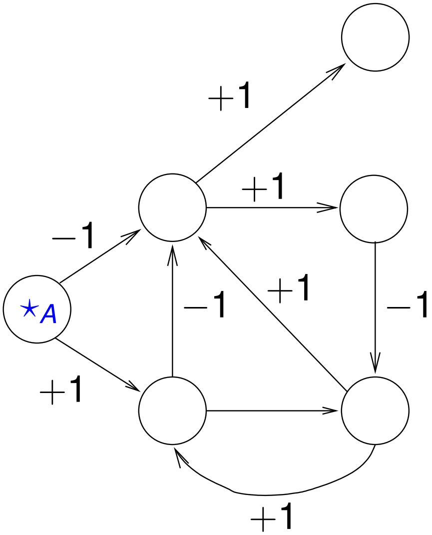
A path is a sequence of moves

$$x_0 \xrightarrow{m_1} x_1 \xrightarrow{m2} \dots \xrightarrow{m_{k-1}} x_{k-1} \xrightarrow{m_k} x_k$$

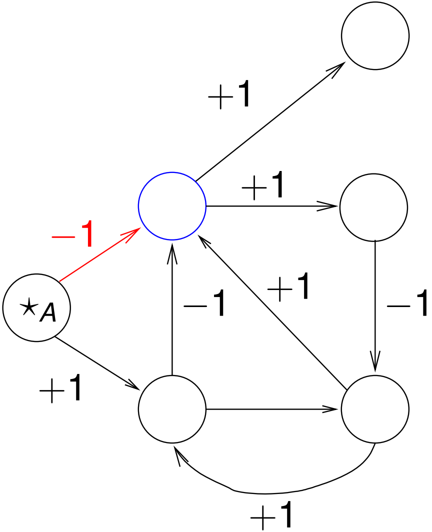
A play is a path starting from the root

$$\star_A \xrightarrow{m_1} x_1 \xrightarrow{m2} \dots \xrightarrow{m_{k-1}} x_{k-1} \xrightarrow{m_k} x_k$$

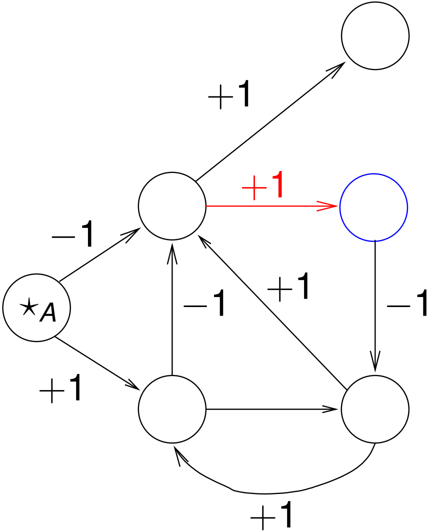
Strategies



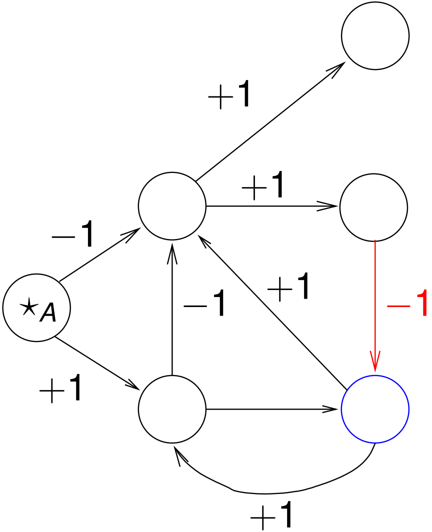
Strategies



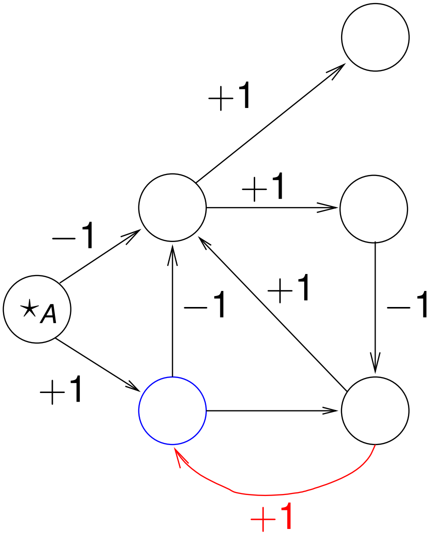
Strategies



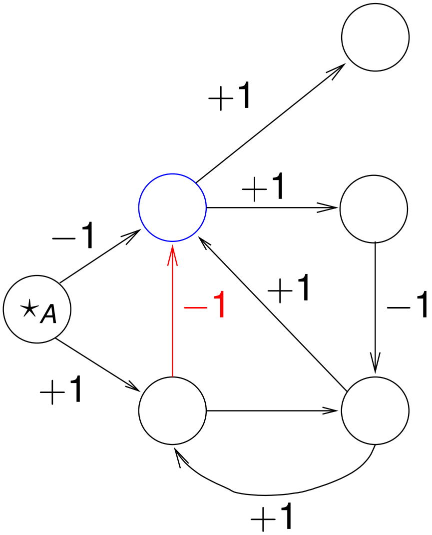
Strategies



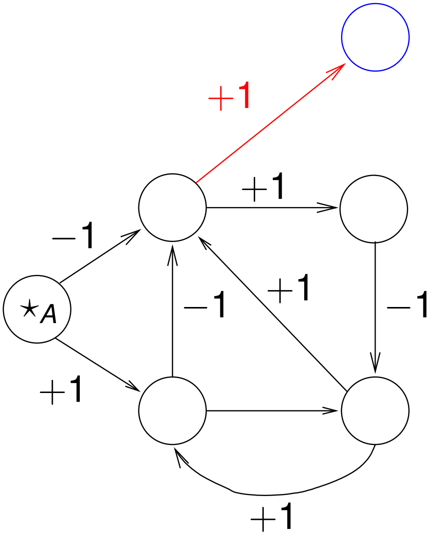
Strategies



Strategies



Strategies



The tensor language

The dual of a Conway game A is the mirror game A^*

$$A^{**} = A$$

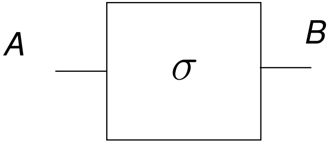
The tensor product of two games A and B, noted $A \otimes B$, is the asynchronous product of the graphs

$$\begin{array}{lll} x \otimes y \longrightarrow x' \otimes y & \text{if} & x \xrightarrow{A} x' \\ x \otimes y \longrightarrow x \otimes y' & \text{if} & y \xrightarrow{B} y' \end{array}$$

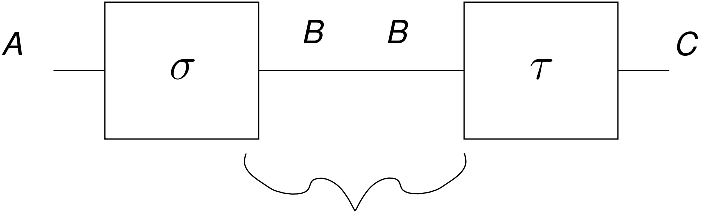
$$(A \otimes B)^* = A^* \otimes B^*$$

The category of Conway games

Morphism :



Composition :



parallel composition on B plus hiding

The category of Conway Games (2)

The category *CONG*, whose objects are Conway games and whose arrows are strategies between them is a

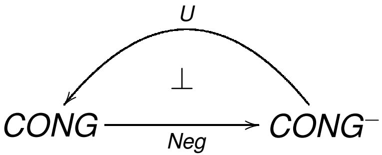
compact closed category

In particular,

$$\frac{(A \otimes B) \to C}{B \to A^* \otimes C}$$

Accommodating the additives

The category does not have binary products [Melliès, 2004]



Neg erases the plays that begin by a player move.

The category $CONG^{-}$ has all finite products
(constructed by pointed sums of the underlying graphs)

Transfer of closed structure : A folklore result

- $(\mathbb{C}, \otimes_{\mathbb{C}}, \multimap_{\mathbb{C}})$ and $(\mathbb{D}, \otimes_{\mathbb{D}})$
- $U : \mathbb{D} \to \mathbb{C}$ strong monoidal, full and faithfull
- $F : \mathbb{C} \to \mathbb{D}$ a right adjoint : $U \dashv F$.

We can export the closure in $\mathbb{C}$ by

$$A \multimap_{\mathbb{D}} B = F(U(A) \multimap_{\mathbb{C}} U(B))$$

(Another account for linear maps in negative games)

Benefits

1. *CONG* : the category of Conway games
   - Compact closed (thus traced and $*$-autonomous)
   - No binary products

2. $CONG^{-}$: the category negative Conway games
   - SMCC : closed structure deduced from *CONG*
   - A trace operator inherited from *CONG*
   - Finite products

The recipe extends smoothly to Conway games with well-bracketed strategies

Benefits

What about duplication ?

Outline

1. Trace and references
2. Conway games
3. ***Duplication and free comonoid***

The exponential modality

Modelling the copy phenomenon

$$S \to S \otimes S$$

$$S \to 1$$

$\Rightarrow$ introduction of a ! modality

Problem : non unique modality

Solution : compute the free comonoid

The free monoid in monoidal categories (easy case)

- The well known free monoid of words

$$\Sigma^* = \bigcup_n \Sigma^n$$

- When all colimits exist and commute with the tensor product, the formula generalises to

$$T(X) = \bigoplus_{n \in \mathbb{N}} \underbrace{X \otimes X \otimes \ldots \otimes X}_{\text{n times}}$$

The free comonoid in Conway games

Intuition : Opening copies one after the other

Construction of $!A$:

- position v : word on the alphabet of positions of A
- $vxw \longrightarrow vyw \quad$ if $\quad x \xrightarrow{A} y$
- $v \longrightarrow vx \quad$ if $\quad \star \xrightarrow{A} x$ is an opponent move (negative part)

Does not come from the previous colimit, because we do not have co-products in *CONG*

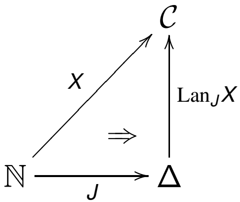
Free monoids as Kan extensions

The categorical recipe to construct the free monoid $T(X)$:

1. See the object X as the strong monoidal functor :

$$n \mapsto X^{\otimes n}$$

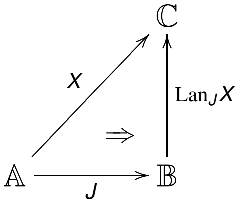
2. Compute the Kan extension in $\mathcal{C}at$:



3. Check that the result $\mathrm{Lan}_J X$ is a strong monoidal functor

Kan extension preserving monoidality

More generally, considering a general Kan extension

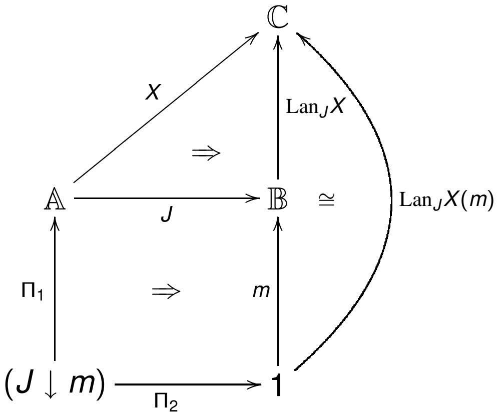


we look for conditions on $\mathbb{A} \xrightarrow{J} \mathbb{B}$ and $\mathbb{C}$ such that the functor

$$\mathrm{Lan}_J : [\mathbb{A}, \mathbb{C}] \to [\mathbb{B}, \mathbb{C}]$$

preserves strong monoidal functors

In good cases, we can compute Kan extensions pointwise by means of colimits

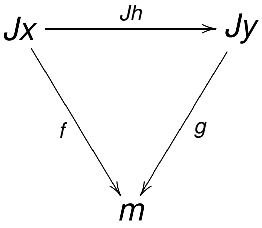


The slice category

Let us recall the definition of $(J \downarrow m)$

Objects : (x, f) where f is an arrow from Jx to m in $\mathbb{B}$,

Arrows : $h : (x, f) \to (y, g)$ where $h : x \to y$ is an arrow such that the following diagram commutes

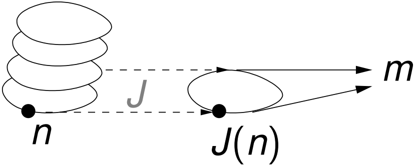


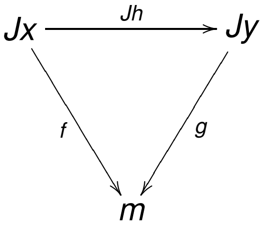
The slice category

Let us recall the definition of $(J \downarrow m)$

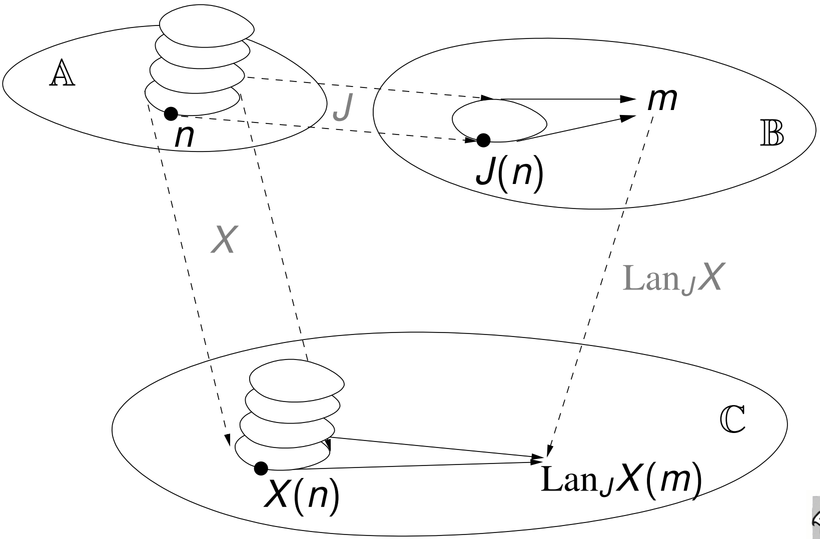
Objects : (x, f) where f is an arrow from Jx to m in $\mathbb{B}$,

Arrows : $h : (x, f) \to (y, g)$ where $h : x \to y$ is an arrow such that the following diagram commutes





Pointwise Kan extension (2)



Proposition

Suppose that J is a colax monoidal functor.

Suppose moreover that the colimits computed on $(J \downarrow m)$ exist in $\mathbb{C}$ and commute with $\otimes_{\mathbb{C}}$.

X is a lax monoidal functor

$\Rightarrow \mathrm{Lan}_J X$ *is a lax monoidal functor*

How to get a strong monoidal functor

Proposition

Suppose that J is a colax monoidal functor.

Suppose moreover that the colimits computed on $(J \downarrow m)$ exist in $\mathbb{C}$ and commute with $\otimes_{\mathbb{C}}$.

$$X \text{ is a lax monoidal functor}$$

$$\Rightarrow \mathrm{Lan}_J X \text{ is a lax monoidal functor}$$

How to get a strong monoidal functor

Decomposable functor

A colax functor J is called decomposable when

$$1 \underset{\longleftarrow}{\overset{\perp}{\longrightarrow}} (J \downarrow 1_{\mathbb{B}})$$

$$(J \downarrow m) \times (J \downarrow n) \underset{\longleftarrow}{\overset{\perp}{\longrightarrow}} (J \downarrow m \otimes n)$$

When is it the case ?

The two canonical situations to get a decomposable functor are

1. **Operads seen as *PROP***

   Monoidal categories coming from operads and functor coming from a map of operads

   Any function of type $Jz \xrightarrow{f} m \otimes n$ can be split into

$$Jz_1 \otimes Jz_2 \xrightarrow{f_1 \otimes f_2} m \otimes n$$

When is it the case ?

The two canonical situations to get a decomposable functor are

1. **Operads seen as *PROP***
2. **Lawvere theories**

   Cartesian categories and cartesian functor

   If the category is cartesian, any function of type $Jz \xrightarrow{f} m \times n$ can be postcomposed by projections to give

$$Jz \times Jz \xrightarrow{(f;\pi_1)\times(f;\pi_2)} m \times n$$

Theorem

Suppose that J is a decomposable functor.

Suppose moreover that the colimits computed on $(J \downarrow m)$ exist in $\mathbb{C}$ and commute with $\otimes_{\mathbb{C}}$.

$$X \text{ is a strong monoidal functor}$$

$$\Rightarrow \mathrm{Lan}_J X \text{ is a strong monoidal functor}$$

The free commutative monoid in Rel

In *Rel*, we have all colimits and commutation.

We want the commutative monoid :

$$\mathbb{A} = Bij \qquad \text{and} \qquad \mathbb{B} = Fun \qquad J = \subseteq$$

So the formula becomes

$$!X(m) = \text{colim}((J \downarrow m) \xrightarrow{\Pi_1} Bij \xrightarrow{X} Rel)$$

$$!X(1) = \underset{m \in Bij}{\text{colim}} \underbrace{X \otimes X \otimes \ldots \otimes X}_{\text{m times}}$$

The free commutative monoid in Rel *(2)*

We compute this colimit in two steps

1. First, the co-equaliser of the permutation on $X^{\otimes n}$

$$X^{\otimes m} \rightrightarrows X^{\otimes m} \longrightarrow S_m X$$

This gives rises to the set $S_m X$ of multisets in X having exactly m elements

2. We compute the sequential colimits of those co-equalisers, which is the disjoint union and find

$$SX = SX(1) = \{\text{finite multisets of elements in } X\}$$

The free cocommutative comonoid in COH

There are basically two choices :

1. Computing the $\parr$-monoid and dualising
2. Dualising the theory and computing the $\otimes$-comonoid

We want the cocommutative comonoid :

$$\mathbb{A} = Bij \qquad \text{and} \qquad \mathbb{B} = Fun^{op} \qquad J = \subseteq$$

So the formula becomes

$$!X(m) = \lim((J \downarrow m) \xrightarrow{\Pi_1} Bij \xrightarrow{X} Rel)$$

$$!X(1) = \lim_{m \in Bij} \underbrace{X \otimes X \otimes \ldots \otimes X}_{\text{m times}}$$

Again, we compute this limit in two steps

1. First, the equaliser of the permutation on $X^{\otimes n}$

$$S_m X \longrightarrow X^{\otimes m} \rightrightarrows X^{\otimes m}$$

   This gives rises to the set $S_m X$ of multicliques in X having exactly m elements

2. We compute the sequential limits of those equalisers

$$!X = !X(1) = \{\text{finite multicliques of } X\}$$

The free cocommutative comonoid in CONG

In *CONG*,

- $\otimes$ commutes will colimits as it has a left adjoint (compact closure)
- we only have filtered colimits

$\Rightarrow$ we compute in Inj^{op} on pointing object $X \to 1$

$$\mathbb{A} = Inj^{op} \qquad \text{and} \qquad \mathbb{B} = Fun^{op} \qquad J = \subseteq$$

Conclusion and Future work

- We have a SMMC with traces, finite products and free cocommutative comonoid
- We can lift the trace operator to the co-Kleisli category
- Study preservation of Kan extensions 2-categorically
  Understand the required properties of the underlying pseudo-monads
- Study the semantics of reference and other effects
  in games, using operads

Conclusion and Future work

- We have a SMMC with traces, finite products and free cocommutative comonoid
- We can lift the trace operator to the co-Kleisli category
- Study preservation of Kan extensions 2-categorically
  Understand the required properties of the underlying pseudo-monads
- Study the semantics of reference and other effects in games, using operads

Conclusion and Future work

- We have a SMMC with traces, finite products and free cocommutative comonoid
- We can lift the trace operator to the co-Kleisli category
- Study preservation of Kan extensions 2-categorically
  Understand the required properties of the underlying pseudo-monads
- Study the semantics of reference and other effects in games, using operads

Conclusion and Future work

- We have a SMMC with traces, finite products and free cocommutative comonoid
- We can lift the trace operator to the co-Kleisli category
- Study preservation of Kan extensions 2-categorically
  Understand the required properties of the underlying pseudo-monads
- Study the semantics of reference and other effects
  in games, using operads

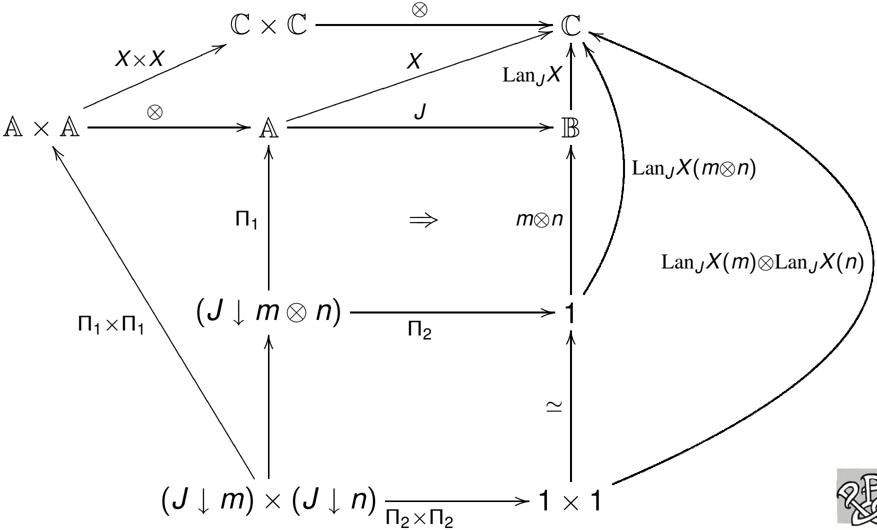
Bibliography

- Joyal, A. (1977).
  Remarques sur la théorie des jeux à deux personnes.
  Gazette des Sciences Mathématiques du Québec, 1(4) :46–52.

- Joyal, A., Street, R., and Verity, D. (1996).
  Traced monoidal categories.
  Mathematical Proceedings of the Cambridge Philosophical Society, (119(3)) :447–468.

- Melliès, P.-A. (2004).
  Asynchronous games 3 : an innocent model of linear logic.
  In Birkedal, L., editor, *Category Theory in Computer Science*. Electronic Notes in Theoretical Computer Science.

Sketch of the proof



Free comonoid and GoI

If we have free comonoid operator *Free* instead of the traditional exponential !, we automatically get a GoI situation with

- $T = Free$

  The retractions require are automatic

- $U = Free(1)$

  $Free(1)$ is always a reflexive object