Ribbon tensorial logic

A functorial bridge between proofs and knots

Paul-André Melliès*

Abstract

Tensorial logic is a primitive logic of tensor and negation which refines linear logic by relaxing the hypothesis that tensorial negation $A \mapsto \neg A$ is involutive. The resulting logic of linear continuations provides a proof-theoretic account of game semantics, where the formulas and proofs of the logic reflect univoquely dialogue games and innocent strategies. In the present paper, we introduce a topologically-aware version of tensorial logic, called ribbon tensorial logic. We associate to every proof of the logic a ribbon tangle which tracks the flow of tensorial negations inside the proof. The translation is functorial: it is performed by exhibiting a correspondence between the notion of dialogue category in proof theory and the notion of ribbon category in knot theory. Our main theorem is that the translation is faithful: two tensorial proofs are equal modulo commuting conversions if and only if the associated ribbon tangles are equal up to topological deformation. The "proof-as-tangle" theorem may be also understood as a coherence theorem for ribbon dialogue categories. By connecting in this functorial way tensorial logic and knot theory, we hope to investigate further the unexpected topological nature of proofs and programs, and of their dialogical interpretation in game semantics.

1 Introduction

One insight of contemporary proof theory is that logical proofs behave in the same way as interactive protocols exchanging information in the course of time. In this dynamic and procedural account of logic, cut-elimination consists in plugging a proof π of a given formula A into a counter-proof π' or execution environment of the same formula. The exchange of information is then performed by letting the two sides π and π' emit and receive atomic tokens in turn. The formula A is the arena where the interaction takes place: it provides the atomic tokens and regulates the exchange. The main reasons for studying these logical protocols is that (1) they are just as expressive as the underlying logic and (2) they may be safely combined together, without ever producing a deadlock or a livelock. This modularity property is structural: it comes from the fact that logical systems are designed to be "consistent" in the sense that there exists no logical proof π of the formula $\perp =$ false.

Depending on the corner of literature, the atomic tokens exchanged between a proof π and a counter-proof π' have received different names: they are called "particles" in the geometry of interaction, and "moves" in game semantics. This shift in terminology from "particles" to "moves" is not entirely innocuous: one intuition conveyed by the geometry of interaction [5, 2] and partly overshadowed by game semantics is the fact that these

*CNRS, Laboratoire IRIF, UMR 8243, Université Paris Diderot, F-75205 Paris, France.

particles "circulate" inside the proof. Typically, working in linear logic but using the traditional notation of classical logic for conjunction and disjunction, one can write as sequents

$$(a) \quad A \wedge A^* \vdash \mathbf{false} \qquad\qquad (b) \quad \mathbf{true} \vdash A \vee A^*$$

these two fundamental principles of logic that:

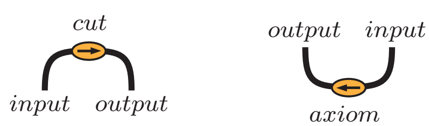
(a) a formula A and its negation A^* cannot be true at the same time,
(b) a formula A or its negation A^* is always true.

Once transcribed in the language of category theory, the sequents define two combinators

$$\begin{array}{rcccc} cut & : & A \wedge A^* & \longrightarrow & \mathbf{false} \\ axiom & : & \mathbf{true} & \longrightarrow & A \vee A^* \end{array}$$

interpreted as specific morphisms in a $*$-autonomous category. The basic idea of the geometry of interaction is that these two combinators *cut* and *axiom* implement input-output channels which may be drawn directly in the following way on the formulas:

The orientation of the arrows on these input-output channels indicates the direction in which the particle will cross the channel:



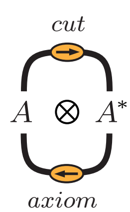
Now, imagine that you fall in the middle of a logical nightmare, where the conjunction $\wedge = \otimes$ and the disjunction $\vee = \parr$ have been identified to the same connective, noted $\otimes$. In that case, the two combinators *cut* and *axiom* have the following type

$$\begin{array}{rcccc} cut & : & A \otimes A^* & \longrightarrow & \mathbf{false} \\ axiom & : & \mathbf{true} & \longrightarrow & A \otimes A^* \end{array}$$

and they may be thus composed into a morphism from **true** to false, in the following way:

$$\mathbf{true} \xrightarrow{axiom} A \otimes A^* \xrightarrow{cut} \mathbf{false}$$

The resulting morphism should be seen as a "proof" of the disjunctive unit $\bot =$ false and thus as a logical inconsistency produced by cut-eliminating the proof $\pi = axiom$ with the counter-proof $\pi' = cut$ on the formula $A \otimes A^*$. When transcribed in the geometry of interaction, the composite morphism induces a loop consisting of two input-output channels:



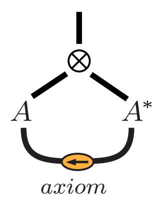
(1)

Depending on the inductive or coinductive nature of the interpretation of logical protocols, such a loop will induce a deadlock where no particle is created inside the loop, or a livelock where a number of particles are created inside the loop and are captured there for ever.

A categorical account of proof-structures. The creation of such loops is a direct threat to the nice modularity of proofs and of their associated logical protocols. One key observation made by Girard is that no such loop is ever produced in linear logic during the cut-elimination of a proof $\pi : A$ against a proof $\pi' : A \multimap B$. This observation underlies the fundamental distinction in linear logic between the two cardinal notions of

$$\text{proof-structure} \qquad \leftrightharpoons \qquad \text{proof-net.}$$

Recall from [4] that a proof-structure of multiplicative linear logic is a "proof-like" structure constructed using the connectives $\otimes$ and $⅋$ of linear logic as well as the axiom and cut links. A proof-net is then defined as a proof-structure generated by a derivation tree π of linear logic. A typical instance of proof-structure which is not a proof-net (and thus does not represent any proof of linear logic) is the following one:



(2)

The starting point of the present paper is to recast in the language of categorical semantics the relationship between proof-nets and proof-structures in multiplicative linear logic. This categorical reformulation is instructive and useful, in particular because it leads us to a very natural definition of "proof-net" and of "proof-structure" for symmetric as well as ribbon tensorial logic.

Following Girard, we have just defined "proof-nets" as specific "proof-structures" generated by derivation trees π of multiplicative linear logic. However, it is customary today to replace the original definition of proof-net by the following one, of a more conceptual flavour: a multiplicative proof-net is a morphism of the free symmetric $*$-autonomous category

$$\textbf{star-autonomous}(\mathscr{C})$$

generated by a given small category $\mathscr{X}$. A typical choice for the category $\mathscr{X}$ is a discrete category (which may be seen as a set) whose objects define the atomic formulas of the logic. In order to investigate the relationship between proof-nets and proof-structures, we suggest to reformulate here the notion of proof-structure in a similarly conceptual way. To that purpose, we consider the free compact closed category

$$\textbf{compact-closed}(\mathscr{X})$$

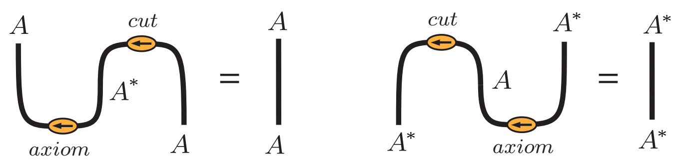
generated by a given small category $\mathscr{X}$. Recall that a compact closed category is a symmetric monoidal category, with unit noted I, where each object A comes equipped with an object A^* and a pair of morphisms

$$\begin{array}{rcccc} cut & : & A^* \otimes A & \longrightarrow & I \\ axiom & : & I & \longrightarrow & A \otimes A^* \end{array}$$

making the diagrams commute:

$$
\begin{tikzcd}
 & A \otimes A^* \otimes A \arrow[dr, "A\otimes cut"] & \\
A \arrow[ur, "axiom\otimes A"] \arrow[rr, "id"] & & A
\end{tikzcd}
\qquad
\begin{tikzcd}
A^* \arrow[rr, "id"] \arrow[dr, "A^*\otimes axiom"'] & & A^* \\
 & A^* \otimes A \otimes A^* \arrow[ur, "cut\otimes A^*"'] &
\end{tikzcd}
$$

These two triangular diagrams are depicted in the language of string diagrams in the following way (see [17] for a nice introduction to string diagrams):



These data make the object A^* a right dual of the object A, what we write $A \dashv A^*$. In a compact closed category $\mathscr{C}$, the operation $A \mapsto A^*$ defines an equivalence of categories

$$(-)^* \quad : \quad \mathscr{C}^{op} \quad \longrightarrow \quad \mathscr{C}$$

between the original category $\mathscr{C}$ and its opposite category. As a matter of fact, a compact-closed category is the same thing as a $*$-autonomous category where the tensor product $\otimes$ and its opposite $⅋$ coincide up to symmetric monoidal equivalence. The construction of the free compact-closed category generated by $\mathscr{X}$ comes with a functor

$$\mathscr{X} \quad \longrightarrow \quad \textbf{compact-closed}(\mathscr{X}).$$

By definition of the free $*$-autonomous category, and since the category $\textbf{compact-closed}(\mathscr{X})$ is $*$-autonomous, this functor can be lifted to a structure-preserving functor of $*$-autonomous categories

$$[-] \;:\; \textbf{star-autonomous}(\mathscr{C}) \longrightarrow \textbf{compact-closed}(\mathscr{C})$$

unique up to isomorphism, which makes the diagram commute:

$$
\begin{tikzcd}
\textbf{star-autonomous}(\mathscr{X}) \arrow[rr, "{[-]}"] & & \textbf{compact-closed}(\mathscr{X}) \\
 & \mathscr{X} \arrow[ul, bend left] \arrow[ur, bend right] &
\end{tikzcd}
$$

This leads us to the following categorical definition of proof-structure:

Definition 1 (proof-structure) *Given two formulas A, B of multiplicative linear logic with objects of $\mathscr{X}$ as atoms, a proof-structure from A to B is defined as a morphism*

$$\Theta \quad : \quad [A] \quad \longrightarrow \quad [B]$$

in the free compact-closed category $\textbf{compact-closed}(\mathscr{X})$ *generated by the category $\mathscr{X}$. By extension, a proof-structure of a formula A of multiplicative linear logic is defined as a proof-structure from the unit 1 to the formula A.*

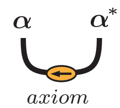
Consider for instance the case where the discrete category $\mathscr{X}$ contains exactly one object noted α, and where the formula A is defined as $A = \alpha$. The functor $[-]$ applied to a formula A of multiplicative linear logic replaces each linear conjunction $\otimes$ and linear disjunction $\parr$ of the formula A by a tensor product. Typically, the two formulas $A \otimes A^* = \alpha \otimes \alpha^*$ and $A \parr A^* = \alpha \parr \alpha^*$ are interpreted as the same object

$$[\,\alpha \otimes \alpha^*\,] \;=\; [\,\alpha \parr \alpha^*\,] \;=\; \alpha \otimes \alpha^*$$

According to our definition, the morphism

$$axiom : I \to \alpha \otimes \alpha^*$$

of the category **compact-closed**$(\mathscr{X})$ depicted as



defines a proof-structure Θ of the formula $\alpha \otimes \alpha^*$. Quite obviously, this proof-structure Θ should be identified with the proof-structure depicted in the more traditional notation (2) used by Girard [4]. Note that the morphism $axiom$ also defines a proof-structure Θ' of the formula $\alpha \parr \alpha^*$. The key difference between the two proof-structures Θ and Θ' is that the proof-structure Θ' is the image

$$\Theta' \quad = \quad [\pi] \quad : \quad I \quad \longrightarrow \quad [\alpha \parr \alpha^*]$$

of a proof-net $\pi : 1 \to \alpha \parr \alpha^*$ living in the free $*$-autonomous category. This is not the case for the proof-structure Θ.

A well-known limitation. One well-known limitation of the theory of proof-nets and of proof-structures in multiplicative linear logic is that the proof-structure $\Theta = [\pi]$ associated to a proof-net π does not characterize uniquely the proof-net. Now that we have reformulated the notions of proof-net and of proof-structure in a categorical way, a simple and concise way to understand this limitation is to observe that the identity and symmetry morphisms

$$id, symm \quad : \quad \bot \otimes \bot \quad \longrightarrow \quad \bot \otimes \bot$$

do not coincide in general in a $*$-autonomous category. From this follows that the two derivation trees π_1 and π_2 below

$$\pi_1 = \quad \dfrac{\dfrac{\dfrac{}{\vdash 1, \bot} \quad \dfrac{}{\vdash 1, \bot}\ \text{axiom}}{\vdash 1, 1, \bot \otimes \bot}\ \otimes\text{-intro}}{\vdash 1 \parr 1, \bot \otimes \bot}\ \parr\text{-intro}$$

$$\pi_2 = \quad \dfrac{\dfrac{\dfrac{\dfrac{}{\vdash 1, \bot} \quad \dfrac{}{\vdash 1, \bot}\ \text{axiom}}{\vdash 1, 1, \bot \otimes \bot}\ \otimes\text{-intro}}{\vdash 1, 1, \bot \otimes \bot}\ \text{exchange}}{\vdash 1 \parr 1, \bot \otimes \bot}\ \parr\text{-intro}$$

which only differ by the exchange rule which permutes the formulas 1 and 1 in the derivation tree π_2, define *different* morphisms of the free $*$-autonomous category, and thus *different* proof-nets π_1 and π_2 of multiplicative linear logic. However, their image $[\pi_1]$ and $[\pi_2]$ in the free compact closed category coincide. The reason is that the two objects 1 and $\bot$ are transported by the functor $[-]$ to the tensor unit I, and that the identity and symmetry morphisms

$$id, symm \quad : \quad I \otimes I \quad \longrightarrow \quad I \otimes I$$

coincide in any symmetric monoidal category. This means that some fundamental information about the proof-nets π_1 and π_2 has been lost when one translates them into the same proof-structure $\Theta = [\pi_1] = [\pi_2]$. Let us stress that this problem has nothing to do with our categorical formulation of proof-nets and of proof-structures: it is a problem inherent to linear logic, see [20, 10, 11, 19, 8, 9] for discussion. As a matter of fact, this problem has haunted the theory of multiplicative proof-nets since the notion was introduced by Girard. A slightly ad hoc solution has been formulated in the literature: the idea is to extend the original notion of proof-structure with "jumps" connecting the disjunctive units $\perp$ to other parts of the proof-structure. Unfortunately, this solution requires to consider proof-structures modulo an equational theory on "rewirings" originally formulated by Trimble. More recently, Heijtljes and Houston [8] have established that the problem of proof-net equivalence is PSPACE-complete, which means that unless P=PSPACE, it is unlikely that there exists an easy graph-theoretic solution to this problem.

The ongoing discussion on the imperfect correspondence between proof-nets and proof-structures may be summarized into the following purely categorical fact:
Annoying fact. The canonical functor

$$[-] \ : \ \mathbf{star\text{-}autonomous}(\mathscr{X}) \ \to \ \mathbf{compact\text{-}closed}(\mathscr{X})$$

which transports a proof-net of multiplicative linear logic to its underlying proof-structure, is not faithful.

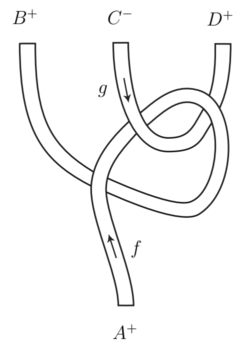
Ribbon categories. The canonical functor $[-]$ from proof-nets to proof-structures transports every derivation tree π to a set of links $[\pi]$ describing the flow of particles through the axiom and cut links of the proof. Unfortunately, we have just seen that the translation $[-]$ is not faithful: two proofs π_1 and π_2 of the same formula A with the same proof-structure Θ may be very well be different. We will explain in a few paragraphs how to correct this uncomfortable situation by shifting from linear logic to tensorial logic. Before that, we would like to take advantage of a number of recent ideas coming from knot theory and representation theory in order to upgrade our current account of proof-structures. What we are aiming at eventually is to "materialise" the set of links $[\pi]$ into a topological ribbon tangle reflecting the interactive behavior of the proof π.

To that purpose, we start from the notion of *ribbon category* which emerged at the interface of knot theory and of representation theory for quantum groups, see [14, 18] for a detailed description. A *ribbon category* is defined as a monoidal category equipped with combinators for braiding and U-turns, satisfying a series of expected equations, see (Def. 7, §2.3) for a definition. The notion of ribbon category is supported by an elegant coherence theorem, which states that the free ribbon category on a category $\mathscr{X}$ has

- as objects: sequences $(A_1^{\epsilon_1}, \ldots, A_n^{\epsilon_n})$ of signed objects of $\mathscr{X}$ where each A_i is an object of the category $\mathscr{X}$, and each ϵ_i is either $+$ or $-$,

- as morphisms: oriented ribbon tangles considered modulo topological deformation, where every open strand is colored by a morphism of $\mathscr{X}$, and every closed strand is colored by an equivalence class of morphisms of $\mathscr{X}$, modulo the equality $g \circ f \sim f \circ g$ for every pair of morphisms of the form $f : A \to B$ and $g : B \to A$.

So, a typical morphism from (A^+) to (B^+, C^-, D^+) in the category $\mathbf{free\text{-}ribbon}(\mathscr{X})$ looks like this



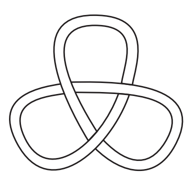
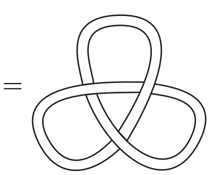
where $f : A \longrightarrow B$ and $g : C \longrightarrow D$ are morphisms in the category $\mathscr{X}$. Now, consider the full and faithful functor

$$\mathscr{X} \quad \longrightarrow \quad \textbf{free-ribbon}(\mathscr{X})$$

which transports every object A of $\mathscr{X}$ to the corresponding signed sequence (A^+). By construction, every functor from the category $\mathscr{X}$ to a ribbon category $\mathscr{D}$ lifts as a structure-preserving functor $[-]$ which makes the diagram below commute:

$$\begin{tikzcd}
\textbf{free-ribbon}(\mathscr{X}) \arrow[rr, "{[-]}"] & & \mathscr{D} \\
& \mathscr{X} \arrow[ul, bend left] \arrow[ur, bend right] &
\end{tikzcd}$$

Once properly oriented and colored, every topological ribbon knot P defines a morphism $P : I \longrightarrow I$ from the tensorial unit $I = ()$ to itself in the category $\textbf{free-ribbon}(\mathscr{X})$. Hence, its image $[P]$ defines an invariant of the ribbon knot P modulo topological deformation. This functorial method enables for instance to establish that the Jones polynomial $[P]$ associated to a ribbon knot P defines a topological invariant, see [14] for details. This kind of topological invariant is quite useful. By way of illustration, the non trivial fact that the left trefoil K_L and the right trefoil K_R depicted below

$K_L =$  $K_R =$ 

are not the same knot modulo deformation, is easily proved by computing their Jones polynomials, and by observing that they are different:

$$[K_L] = \frac{2}{x^2} + \frac{1}{x^4} + \frac{y^2}{x^2} \qquad\qquad [K_R] = 2x^2 - x^4 + x^2y^2$$

An important point is that these topological diagrams can be drawn in ribbon categories precisely because the conjunctive tensor product $\otimes$ and the disjunctive tensor product $\parr$ coincide there. Seen from that operational point of view, the topological ribbon tangles like K_L and K_R are nothing but a sophisticated instance of logical inconsistency, producing a deadlock or a livelock loop (1) in the protocol.

Ribbon dialogue categories. In order to connect proof theory and knot theory, we find convenient to start from a braided notion of dialogue category. The notion of dialogue category has been already used by the author in order to reflect the *dialogical interpretations* of proofs as interactive strategies. A dialogue category is defined as a monoidal category equipped with a primitive notion of duality.

Definition 2 (Dialogue categories) *A dialogue category is a monoidal category $\mathscr{C}$ equipped with an object $\perp$ together with two functors*

$$\begin{array}{rcccc} x \mapsto (x \multimap \perp) & : & \mathscr{C}^{op} & \longrightarrow & \mathscr{C} \\ x \mapsto (\perp \multimapinv x) & : & \mathscr{C}^{op} & \longrightarrow & \mathscr{C} \end{array}$$

and two families of isomorphisms

$$\begin{array}{rcccc} \varphi_{x,y} & : & \mathscr{C}(x \otimes y, \perp) & \cong & \mathscr{C}(y, x \multimap \perp) \\ \psi_{x,y} & : & \mathscr{C}(x \otimes y, \perp) & \cong & \mathscr{C}(x, \perp \multimapinv y) \end{array}$$

natural in x and y.

A ribbon dialogue category is then defined as a dialogue category whose underlying monoidal category $\mathscr{C}$ is balanced in the sense of Joyal and Street [12, 13]. This means that the category $\mathscr{C}$ is equipped with a braiding and a twist, and that it satisfies a series of coherence diagrams reflecting topological equalities of ribbon tangles. Interestingly, no additional coherence property is required between the dialogue structure and the balanced structure.

The proof-theoretic nature of ribbon dialogue categories is witnessed by the fact that they come together with an internal logic: a braided and twisted variant of tensorial logic which we call *ribbon tensorial logic*. The logic is formulated in §3 in the traditional style of proof theory, that is, as a sequent calculus whose derivation trees are identified modulo a notion of proof equality. Just as for linear logic and $*$-autonomous categories, one establishes that the free ribbon dialogue category generated by a category $\mathscr{X}$ has

- objects: the formulas of ribbon tensorial logic (constructed with the binary tensor product $\otimes$ and its unit $I = 1$ together with the left negation $A \mapsto A \multimap \perp$ and the right negation $A \mapsto \perp \multimapinv A$) with atoms provided by the objects of the category $\mathscr{X}$,
- morphisms from A to B: the derivation trees π of the sequent $A \vdash B$ in ribbon tensorial logic, modulo the equational theory of the logic.

The proof-as-tangle theorem. Once the proof-theoretic nature of ribbon dialogue categories firmly established, there remains to relate them to topology. This is achieved by a simple but fundamental observation. A pointed category $(\mathscr{C}, \perp)$ is defined as a category $\mathscr{C}$ equipped with an object $\perp$ singled out in the category. A pointed category may be alternatively defined as an S-algebra for the monad S : ***Cat*** $\rightarrow$ ***Cat*** which transports every category $\mathscr{X}$ to the category $\mathscr{X} + \mathbf{1}$ defined as the disjoint sum of $\mathscr{X}$ with the terminal category $\mathbf{1}$. The unique object of $\mathbf{1}$ is noted $\perp$ and provides the singled-out object of the pointed category $(\mathscr{X} + \mathbf{1}, \perp)$. Every category $\mathscr{X}$ induces a free ribbon category **free-ribbon**$(\mathscr{X} + \mathbf{1})$ generated by the category $\mathscr{X} + \mathbf{1}$. The category **free-ribbon**$(\mathscr{X} + \mathbf{1})$ is monoidal and balanced by construction. The key observation is that it is also a dialogue category where the left and right negation functors are defined as

$$x \multimap \perp \quad \stackrel{def}{=} \quad x^* \otimes \perp \qquad\qquad \perp \multimapinv x \quad \stackrel{def}{=} \quad \perp \otimes x^*.$$

Note that the resulting ribbon dialogue category is somewhat degenerate, since the canonical morphism

$$(\bot \multimapinv (x \multimap \bot)) \otimes y \quad \longrightarrow \quad \bot \multimapinv ((x \otimes y) \multimap \bot)$$

which defines the strength of the double negation monad, is an isomorphism. Now, the unit of the monad S instantiated at the category $\mathscr{X}$

$$inc \quad : \quad \mathscr{X} \quad \longrightarrow \quad \mathscr{X} + \mathbf{1}$$

induces a functor

$$\mathscr{X} \quad \longrightarrow \quad \mathscr{X} + \mathbf{1} \quad \longrightarrow \quad \textbf{free-ribbon}(\mathscr{X} + \mathbf{1})$$

from $\mathscr{X}$ to the ribbon dialogue category **free-ribbon**$(\mathscr{X} + \mathbf{1})$. From this follows that there exists a structure-preserving functor between ribbon dialogue categories

$$[-] \quad : \quad \textbf{free-dialogue}(\mathscr{X}) \quad \longrightarrow \quad \textbf{free-ribbon}(\mathscr{X} + \mathbf{1})$$

which makes the diagram below commute:

$$\begin{tikzcd}
\textbf{free-dialogue}(\mathscr{X}) \arrow[r, "{[-]}"] & \textbf{free-ribbon}(\mathscr{X} + \mathbf{1}) \\
\mathscr{X} \arrow[u] \arrow[r, "inc"] & \mathscr{X} + \mathbf{1} \arrow[u]
\end{tikzcd}$$

The functor $[-]$ transports:

- the formulas of ribbon tensorial logic into signed sequences of $\bot$'s and of logical atoms provided by the objects of the underlying category $\mathscr{X}$,
- the proofs of ribbon tensorial logic modulo proof equality into ribbon tangles modulo topological deformation.

Definition 3 (Proof-nets) *A tensorial proof-net π of ribbon tensorial logic is defined as a morphism of the free ribbon dialogue category.*

Definition 4 (Proof-structures) *A tensorial proof-structure of ribbon tensorial logic from a formula A to a formula B is a morphism $\Theta : [A] \to [B]$ of the free ribbon category.*

We establish in §4 the following "proof-as-tangle" theorem:

Theorem. The functor $[-]$ which transports tensorial proof-nets to tensorial proof-structures, is faithful.

This theorem is important because it enables one to identify every derivation tree π of ribbon tensorial logic modulo commuting conversions, with the underlying ribbon tangle $[\pi]$ modulo topological deformation. The ribbon tangle $[\pi] : [A] \to [B]$ should be understood as a topological "materialisation" of the dialogical interpretation of the proof $\pi : A \to B$ as an innocent strategy between the dialogue games A and B: each strand of the tangle $[\pi]$ describes a specific pair of Opponent and Player moves played by the innocent strategy associated to the tensorial proof π. In this way, the proof-as-tangle theorem provides a topological and type-theoretic foundation to game semantics. Indeed, It should be mentioned that the theorem still holds when one removes the topology of ribbon tangles, and replaces ribbon tensorial logic by commutative tensorial logic, and ribbon categories by compact-closed categories.

Theorem. The canonical functor

$$[-] : \mathbf{free\text{-}dialogue}(\mathscr{C}) \longrightarrow \mathbf{compact\text{-}closed}(\mathscr{C} + \mathbf{1})$$

which transports tensorial proof-nets to tensorial proof-structures in commutative tensorial logic, is faithful.

This means that a derivation tree π of tensorial logic is entirely characterized by its proof-structure $[\pi]$ in ribbon tensorial logic as well as in commutative tensorial logic. The proof-as-tangle theorem resolves in this way the old and annoying problem of the theory of proof-nets of linear logic discussed earlier in the introduction. It also connects proof theory and knot theory by providing a topological coherence theorem for ribbon (or symmetric) dialogue categories.

Plan of the paper. We start by introducing in §2 the notion of ribbon dialogue category. We formulate in §3 the corresponding ribbon tensorial logic, whose proofs are designed to be interpreted in ribbon dialogue categories. The proof-as-tangle theorem for ribbon tensorial logic is stated and established in §4. We finally illustrate in §5 how to use proof-as-tangle theorem as a coherence theorem.

Related works. We would like to mention the early work by Arnaud Fleury [3] who considered a sequent calculus for a braided version of linear logic which is very similar to our sequent calculus for ribbon tensorial logic. Besides the connection already mentioned to the theory of multiplicative proof-nets in linear logic [20, 10, 11, 19, 8, 9], our interpretation of ribbon tensorial proofs as ribbon tangles induces an interpretation of these proofs as sums of planar diagrams in Temperley-Lieb algebras. It would be interesting to compare this interpretation of ribbon tensorial logic with the work by Abramsky [1].

2 Ribbon dialogue categories

We introduce the notion of ribbon dialogue category. To that purpose, we start by recalling the definition of braided monoidal category in § 2.1, of balanced monoidal category in § 2.2 and of ribbon category in § 2.3. We finally formulate our notion of balanced dialogue category in § 2.4.

2.1 Braided monoidal categories

In order to fix notations, we recall that a monoidal category $\mathscr{C}$ is a category equipped with a functor $\otimes : \mathscr{C} \times \mathscr{C} \to \mathscr{C}$ and an object I and three natural isomorphisms

$$\alpha_{A,B,C} \quad : \quad (A \otimes B) \otimes C \quad \longrightarrow \quad A \otimes (B \otimes C)$$

$$\lambda_A \quad : \quad I \otimes A \quad \longrightarrow \quad A \qquad\qquad \rho_A \quad : \quad A \otimes I \quad \longrightarrow \quad A$$

making the two coherence diagrams below commute.

$$\begin{tikzcd}
& (A \otimes B) \otimes (C \otimes D) \arrow[dr, "\alpha"] & \\
((A \otimes B) \otimes C) \otimes D \arrow[ur, "\alpha"] \arrow[d, "\alpha \otimes D"'] & & A \otimes (B \otimes (C \otimes D)) \\
(A \otimes (B \otimes C)) \otimes D \arrow[rr, "\alpha"] & & A \otimes ((B \otimes C) \otimes D) \arrow[u, "A \otimes \alpha"']
\end{tikzcd}$$

$$\begin{tikzcd}
(A \otimes I) \otimes B \arrow[rr, "\alpha"] \arrow[dr, "\rho \otimes B"'] & & A \otimes (I \otimes B) \arrow[dl, "A \otimes \lambda"] \\
& A \otimes B &
\end{tikzcd}$$

Definition 5 (braiding) *A braiding in a monoidal category $\mathscr{C}$ is a family of isomorphisms*

$$\sigma_{A,B} \quad : \quad A \otimes B \quad \longrightarrow \quad B \otimes A$$

natural in x and y such that the two diagrams

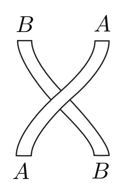
$$\begin{tikzcd}
& A \otimes (B \otimes C) \arrow[r, "\sigma"] & (B \otimes C) \otimes A \arrow[dr, "\alpha"] & \\
(A \otimes B) \otimes C \arrow[ur, "\alpha"] \arrow[dr, "\sigma \otimes C"'] & & (a) & B \otimes (C \otimes A) \\
& (B \otimes A) \otimes C \arrow[r, "\alpha"'] & B \otimes (A \otimes C) \arrow[ur, "B \otimes \sigma"'] &
\end{tikzcd}$$

$$\begin{tikzcd}
& (A \otimes B) \otimes C \arrow[r, "\sigma"] & C \otimes (A \otimes B) \arrow[dr, "\alpha^{-1}"] & \\
A \otimes (B \otimes C) \arrow[ur, "\alpha^{-1}"] \arrow[dr, "A \otimes \sigma"'] & & (b) & (C \otimes A) \otimes B \\
& A \otimes (C \otimes B) \arrow[r, "\alpha^{-1}"'] & (A \otimes C) \otimes B \arrow[ur, "\sigma \otimes B"'] &
\end{tikzcd}$$

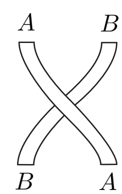
commute.

The braiding map $\sigma_{A,B}$ is depicted in string diagrams as a positive braiding of the ribbon strands A and B where its inverse is depicted as the negative braiding:

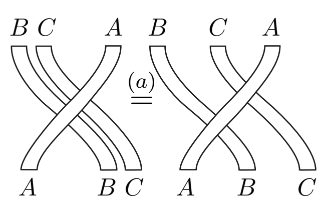
$\sigma_{A,B}$ =

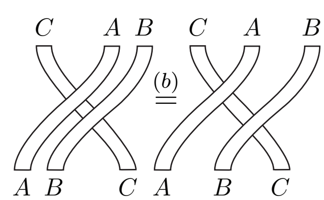


$\sigma^{-1}_{A,B}$ =



The two coherence diagrams (a) and (b) are then depicted as topological equalities between string diagrams:





2.2 Balanced categories

Definition 6 (balanced category) *A balanced category $\mathscr{C}$ is a braided monoidal category equipped with a family of morphisms*

$$\theta_A \quad : \quad A \quad \longrightarrow \quad A$$

natural in A, satisfying the equality

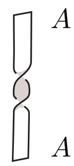
$$\theta_I \quad = \quad \mathrm{id}_I$$

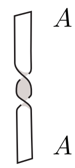
where I is the monoidal unit, and making the diagram

$$\begin{tikzcd} A \otimes B \arrow[r, "\sigma_{A,B}"] \arrow[d, "\theta_{A\otimes B}"'] & B \otimes A \arrow[d, "\theta_B \otimes \theta_A"] \\ A \otimes B & B \otimes A \arrow[l, "\sigma_{B,A}"] \end{tikzcd} \tag{3}$$

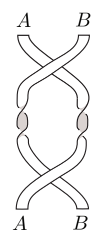
commute for all objects A and B of the category $\mathscr{C}$.

The twist θ_A is depicted as the ribbon A twisted positively in the trigonometric direction with an angle 2π whereas its inverse θ_A^{-1} is depicted as the same ribbon A twisted this time negatively with an angle -2π:

θ_A = 

θ_A^{-1} = 

This notation enables us to give a topological motivation to the axioms of a balanced category. The first requirement that θ_I is the identity means that the ribbon strand I should be thought as ultra thin. The second requirement that the coherence diagram (3) commutes reflects the following topological equality between string diagrams:

$\theta_{A\otimes B}$ = 

2.3 Ribbon categories

This leads us to the well-known definition of ribbon category.

Definition 7 (ribbon category) *A ribbon category $\mathscr{C}$ is a balanced category where every object A has a right dual A^*, what we write $A \dashv A^*$.*

See the introduction for a definition of right dual.

2.4 Ribbon dialogue categories

At this stage, we are ready to introduce the notion of *ribbon dialogue category* which provides a functorial bridge between proof theory and knot topology.

Definition 8 (ribbon dialogue categories) *A balanced dialogue category is a dialogue category $\mathscr{C}$ in the sense of Def. 2, moreover equipped with a braiding and a twist defining a balanced category.*

An interesting aspect of the definition is that it does not require any coherence relation between the dialogue structure and balanced structure of the category $\mathscr{C}$.

Illustration. An instructive example of ribbon dialogue category $\mathscr{D}$ coming from algebra, and more specifically from the representation theory of quantum groups, is the following one: the category $\mathbf{Mod}(H)$ of (finite and infinite dimensional) H-modules associated to a ribbon Hopf algebra H. Note that the full subcategory $\mathscr{C}$ of rigid objects A in a ribbon dialogue category $\mathscr{D}$ (that is, objects with a right dual) is a ribbon category. Typically, the category $\mathbf{Mod}_f(H)$ of finite dimensional H-modules associated to a ribbon Hopf algebra H defines a ribbon category, see [14] for details.

3 Ribbon tensorial logic

We introduce below the sequent calculus of ribbon tensorial logic, and mention a number of commuting conversions involved in the cut-elimination procedure.

3.1 The ribbon groups

Recall that the braid group $\mathbf{B}_n$ on n strands is presented by the generators σ_i for $1 \leq i \leq n-1$ and the equations

$$\begin{array}{rcll} \sigma_i \circ \sigma_{i+1} \circ \sigma_i & = & \sigma_{i+1} \circ \sigma_i \circ \sigma_{i+1} & \\ \sigma_i \circ \sigma_j & = & \sigma_j \circ \sigma_i & \text{when } |j-i| \geq 2. \end{array} \tag{4}$$

There is an obvious left action

$$\triangleright \quad : \quad \mathbf{B}_n \times [n] \quad \longrightarrow \quad [n] \tag{5}$$

of the group $\mathbf{B}_n$ on the set $[n] = \{1, \cdots, n\}$ of strands. This action enables one to define a wreath product of $\mathbf{B}_n$ on the additive group $(\mathbb{Z}, +, 0)$. The resulting group $\mathbf{G}_n$ is called the *ribbon group* on n strands. The group is presented by the generators σ_i for $1 \leq i \leq n-1$ and θ_i for $1 \leq i \leq n$, together with the equations (4) of the braid group $\mathbf{B}_n$ and the equations below:

$$\begin{array}{rcll} \sigma_i \circ \theta_i & = & \theta_{i+1} \circ \sigma_i & \\ \sigma_i \circ \theta_{i+1} & = & \theta_i \circ \sigma_i & \\ \sigma_i \circ \theta_j & = & \theta_j \circ \sigma_i & \text{when } j < i \text{ or when } j \geq j+2. \end{array}$$

Each group $\mathbf{G}_n$ may be alternatively seen as a groupoid noted $S\,\mathbf{G}_n$, with a unique object $*$ and $S\,\mathbf{G}_n(*,*) = \mathbf{G}_n$. A nice and conceptual definition of the ribbon groups $\mathbf{G}_n$ is possible, as follows. The groupoid $\mathscr{G}$ defined as the disjoint sum of the groupoids $S\,\mathbf{G}_n$ coincides with the free balanced category generated by the terminal category $\mathbf{1}$. Recall that the category $\mathbf{1}$ has a unique object $*$ and a unique map. Hence, the group $\mathbf{G}_n$ may be alternatively defined as $\mathscr{G}(n,n)$ where $n = 1 \otimes \cdots \otimes 1$ is the n-fold tensor product of the generator 1 of the category $\mathscr{G}$. The fact that the free balanced category $\mathscr{G}$ generated by the category $\mathbf{1}$ coincides with the disjoint sum of the groupoids $S\,\mathbf{G}_n$ is just the ribbon-theoretic counterpart to the well-known fact that the free braided monoidal category $\mathscr{B}$ generated by the category $\mathbf{1}$ coincides with the disjoint sum of the groupoids $S\,\mathbf{B}_n$. From this observation follows that there exists a family of group homomorphisms

$$\otimes \quad : \quad \mathbf{G}_p \times \mathbf{G}_q \quad \longrightarrow \quad \mathbf{G}_{p+q}$$

which reflects the monoidal structure of the balanced category $\mathscr{G}$. Moreover, the action (5) extends to a left action

$$\triangleright \quad : \quad \mathbf{G}_n \times [n] \quad \longrightarrow \quad [n]$$

Axiom	$\dfrac{f : \alpha \to \beta \text{ in the category } \mathscr{X}}{\alpha \vdash \beta}$
Cut	$\dfrac{\Gamma \vdash A \qquad \Upsilon_1, A, \Upsilon_2 \vdash B}{\Upsilon_1, \Gamma, \Upsilon_2 \vdash B}$
Right $\otimes$-introduction	$\dfrac{\Gamma \vdash A \qquad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}$
Left $\otimes$-introduction	$\dfrac{\Upsilon_1, A, B, \Upsilon_2 \vdash C}{\Upsilon_1, A \otimes B, \Upsilon_2 \vdash C}$
Right 1-introduction	$\dfrac{}{\vdash 1}$
Left 1-introduction	$\dfrac{\Upsilon_1, \Upsilon_2 \vdash A}{\Upsilon_1, 1, \Upsilon_2 \vdash A}$
Right $(\bot \multimapinv)$-introduction	$\dfrac{\Gamma, A \vdash \bot}{\Gamma \vdash \bot \multimapinv A}$
Left $(\bot \multimapinv)$-introduction	$\dfrac{\Gamma \vdash A}{\bot \multimapinv A, \Gamma \vdash \bot}$
Right $(\multimap \bot)$-introduction	$\dfrac{A, \Gamma \vdash \bot}{\Gamma \vdash A \multimap \bot}$
Left $(\multimap \bot)$-introduction	$\dfrac{\Gamma \vdash A}{\Gamma, A \multimap \bot \vdash \bot}$

Figure 1: Sequent calculus of tensorial logic.

where each generator θ_i acts trivially on $[n] = \{1, \dots, n\}$, in the sense that $\theta_i \rhd k = k$ for all $k \in [n]$.

3.2 The sequent calculus

The formulas of ribbon tensorial logic are finite trees generated by the grammar

$$A, B ::= A \otimes B \;\mid\; 1 \;\mid\; A \multimap \bot \;\mid\; \bot \multimapinv A \;\mid\; \bot \;\mid\; \alpha$$

where α is an object of a fixed small category $\mathscr{X}$ of atoms. The sequents are two-sided

$$A_1, \dots, A_m \;\vdash\; B$$

with a *sequence* of formulas $A_1, ..., A_m$ on the left-hand side, and a *unique* formula B on the right-hand side. The proofs of ribbon tensorial logic are defined as derivation trees in a carefully designed sequent calculus, which we formulate now. The sequent calculus is defined as the usual sequent calculus of tensorial logic, recalled in Figure 1, together with a family of exchange rules

$$[g] \qquad \dfrac{A_1, \dots, A_n \vdash B}{A_{g\rhd 1}, \dots, A_{g\rhd n} \vdash B}$$

parametrized by the elements g of the ribbon group $\mathbf{G}_n$.

3.3 The commutative conversions

Ribbon tensorial logic is inspired by the topology of knots, and one thus needs to take an extra care in order to design the equational theory on its derivation trees. Nonetheless, the basic recipe to identify two derivation trees π_1 and π_2 is the same in ribbon tensorial logic as in any other sequent calculus: the equality is defined by a series of local commuting conversions

$$\pi_1 \quad \leftrightsquigarrow \quad \pi_2$$

on derivation trees. Moreover, these commuting conversions rules are essentially the same for ribbon tensorial logic as for traditional (commutative) tensorial logic. The only difference is that every exchange rule of ribbon tensorial logic is labelled by an element $g \in \mathbf{G}_n$ of the ribbon group. For that reason, one needs to treat with an extreme attention every commuting conversion $\pi_1 \leftrightsquigarrow \pi_2$ involving an exchange rule. For each such commuting conversion, the challenge is to label properly the exchange rules appearing on each side π_1 and π_2 of the conversion in commutative tensorial logic, in order for the conversion $\pi_1 \leftrightsquigarrow \pi_2$ to make sense in ribbon tensorial logic. The archetypal illustration of commuting conversion in ribbon tensorial logic is provided by the conversion which transforms the derivation tree

$$\mathrm{Cut}\ \dfrac{\dfrac{\begin{matrix}\pi_1\\ \vdots\end{matrix}}{A_1, \ldots, A_n \vdash A} \qquad \dfrac{\dfrac{\begin{matrix}\pi_2\\ \vdots\end{matrix}}{\Upsilon_1, B, A, \Upsilon_2 \vdash C}}{\Upsilon_1, A, B, \Upsilon_2 \vdash C}\ [p \otimes \sigma \otimes q]}{\Upsilon_1, A_1, \ldots, A_n, B, \Upsilon_2 \vdash C}$$

into the derivation tree

$$\dfrac{\mathrm{Cut}\ \dfrac{\dfrac{\begin{matrix}\pi_1\\ \vdots\end{matrix}}{A_1, \ldots, A_n \vdash A} \qquad \dfrac{\begin{matrix}\pi_2\\ \vdots\end{matrix}}{\Upsilon_1, B, A, \Upsilon_2 \vdash C}}{\Upsilon_1, B, A_1, \ldots, A_n, \Upsilon_2 \vdash C}}{\Upsilon_1, A_1, \ldots, A_n, B, \Upsilon_2 \vdash C}\ [p \otimes \sigma_{n,1} \otimes q]$$

where p and q are the respective lengths of Υ_1 and of Υ_2 and where $\sigma_{m,n}$ is defined as the positive braid permuting m strands above n strands. Another important illustration is the conversion which transforms the derivation tree

$$\mathrm{Cut}\ \dfrac{\dfrac{\begin{matrix}\pi_1\\ \vdots\end{matrix}}{A_1, \ldots, A_n \vdash A} \qquad \dfrac{\dfrac{\begin{matrix}\pi_2\\ \vdots\end{matrix}}{\Upsilon_1, A, \Upsilon_2 \vdash C}}{\Upsilon_1, A, \Upsilon_2 \vdash C}\ [p \otimes \theta \otimes q]}{\Upsilon_1, A_1, \ldots, A_n, \Upsilon_2 \vdash C}$$

into the derivation tree

$$\dfrac{\mathrm{Cut}\ \dfrac{\dfrac{\begin{matrix}\pi_1\\ \vdots\end{matrix}}{A_1, \ldots, A_n \vdash A} \qquad \dfrac{\begin{matrix}\pi_2\\ \vdots\end{matrix}}{\Upsilon_1, A, \Upsilon_2 \vdash C}}{\Upsilon_1, A_1, \ldots, A_n, \Upsilon_2 \vdash C}}{\Upsilon_1, A_1, \ldots, A_n, \Upsilon_2 \vdash C}\ [p \otimes \theta\langle n\rangle \otimes q]$$

where $\theta\langle n\rangle$ is the positive twist on n strands. These two commutative conversions should be understood as naturality conditions on the braiding σ and on the twist θ. Yet another important commuting conversion identifies for every pair $g, h \in \mathbf{G}_n$ the derivation tree

$$
[g]\;\dfrac{[h]\;\dfrac{\dfrac{\begin{array}{c}\pi\\ \vdots\end{array}}{A_1,\ldots,A_n \vdash B}}{A_{h\rhd 1},\ldots,A_{h\rhd n}\vdash B}}{A_{g\rhd(h\rhd 1)},\ldots,A_{g\rhd(h\rhd n)}\vdash B}
$$

with the derivation tree

$$
[g\circ h]\;\dfrac{\dfrac{\begin{array}{c}\pi\\ \vdots\end{array}}{A_1,\ldots,A_n \vdash B}}{A_{g\circ h\rhd 1},\ldots,A_{g\circ h\rhd n}\vdash B}
$$

This commuting conversion comes with a similar conversion for the unit element $e \in \mathbf{G}_n$. Together, the two commuting conversions ensure that the action of the ribbon group $\mathbf{G}_n$ on a sequent $A_1,\ldots,A_n \vdash B$ with n hypothesis is algebraic in the traditional sense, modulo conversion.

One main technical observation of the paper is that the traditional coherence diagrams which define a braiding (in §2.1) and a twist (in §2.2) can be "internalized" as commuting conversions of ribbon tensorial logic. Typically, the coherence diagram (a) for braiding (Def. 5, §2.1) is reflected in ribbon tensorial logic by the commuting conversion which identifies the derivation tree

$$
[p\otimes\sigma\otimes q]\;\dfrac{\text{Left}\otimes\;\dfrac{\dfrac{\begin{array}{c}\pi\\ \vdots\end{array}}{\Upsilon_1,A,B,C,\Upsilon_2\vdash D}}{\Upsilon_1,A\otimes B,C,\Upsilon_2\vdash D}}{\Upsilon_1,C,A\otimes B,\Upsilon_2\vdash D}
$$

with the derivation tree

$$
\text{Left}\otimes\;\dfrac{[p\otimes\sigma\otimes 1\otimes q]\;\dfrac{[p\otimes 1\otimes\sigma\otimes q]\;\dfrac{\dfrac{\begin{array}{c}\pi\\ \vdots\end{array}}{\Upsilon_1,A,B,C,\Upsilon_2\vdash D}}{\Upsilon_1,A,C,B,\Upsilon_2\vdash D}}{\Upsilon_1,C,A,B,\Upsilon_2\vdash D}}{\Upsilon_1,C,A\otimes B,\Upsilon_2\vdash D}
$$

Similarly, the coherence diagram (3) for the twist in the definition of a balanced category (Def. 6,§2.2) is reflected by the commuting conversion which identifies the derivation tree

$$
[p\otimes\theta\otimes q]\;\dfrac{\text{Left}\otimes\;\dfrac{\dfrac{\begin{array}{c}\pi\\ \vdots\end{array}}{\Upsilon_1,A,B,\Upsilon_2\vdash C}}{\Upsilon_1,A\otimes B,\Upsilon_2\vdash C}}{\Upsilon_1,A\otimes B,\Upsilon_2\vdash C}
$$

with the derivation tree

$$
\text{Left}\otimes\;\dfrac{[p\otimes\sigma\otimes q]\;\dfrac{[p\otimes\sigma\otimes q]\;\dfrac{[p\otimes\theta\otimes\theta\otimes q]\;\dfrac{\dfrac{\begin{array}{c}\pi\\ \vdots\end{array}}{\Upsilon_1,A,B,\Upsilon_2\vdash C}}{\Upsilon_1,A,B,\Upsilon_2\vdash C}}{\Upsilon_1,B,A,\Upsilon_2\vdash C}}{\Upsilon_1,A,B,\Upsilon_2\vdash C}}{\Upsilon_1,A\otimes B,\Upsilon_2\vdash C}
$$

The fact that the coherence diagrams required of a balanced category can be "lifted" in this way to ribbon tensorial logic is somewhat surprising, because it means that the topological notions of braiding σ and of twist θ are compatible with the logical and multi-categorical (rather than categorical) nature of sequents $A_1, \cdots, A_n \vdash B$ of tensorial logic.

3.4 The cut elimination theorem

The equational theory of ribbon tensorial logic is thus obtained from the equational theory of traditional (commutative) tensorial logic by selecting very carefully the label $g \in \mathbf{G}_n$ associated for each conversion rule $\pi_1 \leftrightsquigarrow \pi_2$ and for each exchange rule appearing in the derivation trees π_1 and of π_2. Once these choices of labelling have been done properly, it is not difficult to establish the following cut-elmination theorem, in just the same way as for commutative tensorial logic:

Theorem 1 (cut-elimination) *Every derivation tree of ribbon tensorial logic is equivalent to a cut-free proof modulo commuting conversions.*

3.5 A focalisation theorem

The commuting conversions of ribbon tensorial logic are not only useful to prove the cut-elimination theory. They also enable us to establish a focalisation theorem for the derivation trees of the logic. The theorem is important because it ensures that every cut-free derivation tree π can be transformed by a series of commuting conversions to a normal form π_{nf} where the construction of the derivation is performed in phases. A cycle of construction starts with a number of derivation trees

$$\dfrac{\begin{matrix}\pi_1\\ \vdots\end{matrix}}{\Gamma_1 \vdash A_1} \quad \cdots \quad \dfrac{\begin{matrix}\pi_i\\ \vdots\end{matrix}}{\Gamma_i \vdash A_i} \quad \cdots \quad \dfrac{\begin{matrix}\pi_n\\ \vdots\end{matrix}}{\Gamma_n \vdash A_n}$$

where all the formulas of the context Γ_i are either atomic: that is, equal to an object α of the category $\mathscr{X}$, or negated: that is, of the form $A \multimap \bot$ or $\bot \multimapinv A$. One then applies the phases below one after the other in order to get a derivation tree π of the same form, whose sequent $\Gamma \vdash A$ has all the formulas of its context Γ either atomic or negated.

1. A left introduction rule of the left negation or of the right negation which produces a sequent whose conclusion formula is $\bot$,
2. A series of exchange rules which permute the formulas of the context,
3. A series of left $\otimes$-introduction and of left 1-introduction rules, which produces a sequent where at most one formula in the context is not negated or atomic,
4. A right introduction of the left negation or of the right negation, or an axiom rule, which produces a sequent where all the formulas (context and conclusion) are either negated or atomic,
5. A series of right $\otimes$-introduction rules, and of right 1-introduction rules,
6. A series of exchange rules which permute the atomic or negated formulas of the context,

As just claimed, one obtains at the end of each cycle a sequent $\Gamma \vdash A$ where all the formulas of the context Γ are either negated or atomic. A derivation tree π is called *focused* when it has been produced by a number of such construction cycles.

Theorem 2 (focalisation) *Every derivation tree π is equivalent to a focused derivation tree π_{nf} modulo the commuting conversions of ribbon tensorial logic.*

The theorem is based on the ability of permuting the order of introduction rules using commuting conversions. The proof is essentially standard, except for the special care required by the exchange rules.

3.6 Soundness theorem

Suppose given a functor $\mathscr{X} \longrightarrow \mathscr{D}$ from the category of atoms of our ribbon tensorial logic, to a given ribbon dialogue category $\mathscr{D}$. Then, one establishes that

Theorem 3 (soundness) *Every derivation tree π of a sequent*

$$A_1 \otimes \cdots \otimes A_n \quad \vdash \quad B$$

in ribbon tensorial logic may be interpreted as a morphism

$$[\pi] \quad : \quad A_1 \otimes \cdots \otimes A_n \quad \longrightarrow \quad B$$

of the ribbon dialogue category $\mathscr{D}$. Moreover, the interpretation $[\pi]$ provides an invariant of the derivation tree π modulo commuting conversions.

The interpretation $[\pi]$ of the derivation tree π is defined by structural induction on the height of the derivation tree. The only interesting point of the construction is that the exchange rule

$$[g] \qquad \frac{A_1, \ldots, A_n \vdash B}{A_{g\triangleright 1}, \ldots, A_{g\triangleright n} \vdash B}$$

is interpreted by precomposing the interpretation

$$[\pi] \quad : \quad A_1, \ldots, A_n \quad \longrightarrow \quad B$$

of the proof π with the morphism

$$A_{g\triangleright 1}, \ldots, A_{g\triangleright n} \quad \longrightarrow \quad A_1, \ldots, A_n$$

associated to the element $g \in \mathbf{G}_n$ of the ribbon group acting on the object $A_1 \otimes \ldots \otimes A_n$ in the ribbon dialogue category $\mathscr{D}$.

4 The proof-as-tangle theorem

A more conceptual and sophisticated way to formulate the soundness theorem (Thm. 3, §3.6) is to state that the free ribbon dialogue category

$$\textbf{free-ribbon}(\mathscr{X})$$

generated by the category $\mathscr{X}$ of atoms, coincides with a category of tensorial formulas and of derivation trees modulo the equational theory of ribbon tensorial logic. We have seen in the introduction how to deduce from this property a functor

$$[-] \quad : \quad \textbf{free-ribbon}(\mathscr{X}) \quad \longrightarrow \quad \textbf{free-ribbon}(\mathscr{X} + \mathbf{1})$$

which transports every tensorial proof-net π into a topological tangle $[\pi]$. We establish now the main result of the paper.

Theorem 4 (proof-as-tangle) *The functor $[-]$ is faithful.*

Proof. The proof is to a large extent based on the focalisation theorem (Thm. 2). Suppose that two cut-free derivation trees

$$\dfrac{\begin{array}{c}\pi_1\\ \vdots\end{array}}{A \vdash B} \qquad \dfrac{\begin{array}{c}\pi_2\\ \vdots\end{array}}{A \vdash B}$$

of ribbon logic induce the same tangle $[\pi_1] = [\pi_2]$ modulo topological deformation in the free ribbon category

$$\mathbf{free\text{-}ribbon}(\mathscr{X} + 1).$$

We show that $\pi_1 \leftrightsquigarrow \pi_2$ and conclude. We proceed by induction on the number of links in the tangle. By the focalisation theorem, we know that the proofs π_1 and π_2 are equal modulo logical equality to:

$$\dfrac{\begin{array}{c}\pi_1'\\ \vdots\end{array}}{A_1, \ldots, A_n \vdash B} \qquad \dfrac{\begin{array}{c}\pi_2'\\ \vdots\end{array}}{A_1, \ldots, A_n \vdash B}$$

where each A_i is either a negation or an atom, followed by the same sequence of left introduction of tensor and left introduction of unit. Suppose that $B = \bot$. In that case, the formula $\bot$ was either introduced by:

- the left introduction of a left negation $X \multimap \bot$,
- or the left introduction of a right negation $\bot \multimapinv X$.

We may suppose without loss of generality that this last rule introduces a left negation $X \multimap \bot$ in the context. In that case, the proof π_1' is equal to

$$[g]\,\dfrac{\text{Left} \multimap \dfrac{\dfrac{\begin{array}{c}\pi_1''\\ \vdots\end{array}}{X_1, \ldots, X_{n-1} \vdash X}}{X_1, \ldots, X_{n-1}, X \multimap \bot \vdash \bot}}{A_1, \ldots, A_n \vdash \bot}$$

The equality of $[\pi_1]$ and $[\pi_2]$ modulo deformation implies that $\bot$ is connected in $[\pi_2']$ to the same formula $X \multimap \bot$. From that follows that $X \multimap \bot$ is also introduced in π_2 by the left introduction of a left negation. In other words, the proof π_2' factors as

$$[h]\,\dfrac{\text{Left} \multimap \dfrac{\dfrac{\begin{array}{c}\pi_2''\\ \vdots\end{array}}{Y_1, \ldots, Y_{n-1} \vdash Y}}{Y_1, \ldots, Y_{n-1}, X \multimap \bot \vdash \bot}}{A_1, \ldots, A_n \vdash \bot}$$

From this, we conclude that the derivation tree

$$\text{Left} \multimap \dfrac{\dfrac{\begin{array}{c}\pi_1''\\ \vdots\end{array}}{X_1, \ldots, X_{n-1} \vdash X}}{X_1, \ldots, X_{n-1}, X \multimap \bot \vdash \bot}$$

induces the same topological tangle as the derivation tree

$$
\begin{array}{rc}
 & \pi_2'' \\
 & \vdots \\ \cline{2-2}
 & Y_1, \ldots, Y_{n-1} \vdash Y \\ \cline{2-2}
\text{Left} \multimap & Y_1, \ldots, Y_{n-1}, X \multimap \bot \vdash \bot \\ \cline{2-2}
[h] & A_1, \ldots, A_n \vdash \bot \\ \cline{2-2}
[g^{-1}] & X_1, \ldots, X_{n-1}, X \multimap \bot \vdash \bot
\end{array}
$$

Since all the formulas A_i are either negated formulas or atoms, we deduce from the topology of tangles that

$$g^{-1} \circ h \in \mathbf{G}_n$$

is of the form $f \otimes 1$. From this follows that the proof

$$
\begin{array}{c}
\pi_1'' \\
\vdots \\ \hline
X_1, \ldots, X_{n-1} \vdash X
\end{array}
$$

has the same topological tangle as the proof

$$
\begin{array}{rc}
 & \pi_2'' \\
 & \vdots \\ \cline{2-2}
 & Y_1, \ldots, Y_{n-1} \vdash X \\ \cline{2-2}
[f] & X_1, \ldots, X_{n-1} \vdash X
\end{array}
$$

From this, we deduce by induction hypothesis that they are equal proofs — in the sense that their proof-nets coincide. The proof π_1' is thus equal to the proof

$$
\begin{array}{rc}
 & \pi_2'' \\
 & \vdots \\ \cline{2-2}
 & Y_1, \ldots, Y_{n-1} \vdash X \\ \cline{2-2}
[f] & X_1, \ldots, X_{n-1} \vdash X \\ \cline{2-2}
\text{Left} \multimap & X_1, \ldots, X_{n-1}, X \multimap \bot \vdash \bot \\ \cline{2-2}
[g] & A_1, \ldots, A_n \vdash \bot
\end{array}
$$

which may be rewritten into

$$
\begin{array}{rc}
 & \pi_2'' \\
 & \vdots \\ \cline{2-2}
 & Y_1, \ldots, Y_{n-1} \vdash X \\ \cline{2-2}
\text{Left} \multimap & Y_1, \ldots, Y_{n-1}, X \multimap \bot \vdash \bot \\ \cline{2-2}
[f \otimes 1] & X_1, \ldots, X_{n-1}, X \multimap \bot \vdash \bot \\ \cline{2-2}
[g] & A_1, \ldots, A_n \vdash \bot
\end{array}
$$

which may be rewritten into the proof π_2:

$$
\begin{array}{rc}
 & \pi_2'' \\
 & \vdots \\ \cline{2-2}
 & Y_1, \ldots, Y_{n-1} \vdash X \\ \cline{2-2}
\text{Left} \multimap & Y_1, \ldots, Y_{n-1}, X \multimap \bot \vdash \bot \\ \cline{2-2}
[h] & A_1, \ldots, A_n \vdash \bot
\end{array}
$$

This concludes the proof by induction when the conclusion B of the two sequents π_1 and π_2 is equal to $B = \bot$.

We have seen the most difficult part of the topological argument establishing the "proof-as-tangle" theorem. The remaining part of the argument works in essentially the same way. For instance, suppose that the conclusion of the sequent

$$A_1, \cdots, A_n \quad \vdash \quad B$$

produced by π_1 and π_2 is $B = B_1 \otimes B_2$, and that all the hypothesis $A_1, \ldots, A_n$ are either negated or atomic. In that case, one may suppose without loss of generality that the last rule of π_1 introduces a tensor on the right. The derivation tree π_1 thus factors as

$$\text{Right} \otimes \frac{\dfrac{\begin{matrix}\pi_{11}\\ \vdots\end{matrix}}{A_1, \ldots, A_k \vdash B_1} \qquad \dfrac{\begin{matrix}\pi_{12}\\ \vdots\end{matrix}}{A_{k+1}, \ldots, A_n \vdash B_2}}{A_1, \ldots, A_k, A_{k+1}, \ldots, A_n \vdash B_1 \otimes B_2}$$

The fact that the tangles $[\pi_1]$ and$[\pi_2]$ are equal modulo deformation implies that π_2 splits in the same way as two proofs π_{21} and π_{22} ; moreover, the tangles $[\pi_{11}]$ and $[\pi_{21}]$ are equal modulo deformation, and similarly for $[\pi_{12}]$ and $[\pi_{22}]$. This enables one to conclude by induction that π_{11} and π_{21} are equal modulo commuting conversions, and similarly for π_{12} and π_{22}. This concludes our argument that π_1 and π_2 are equal modulo commuting conversions.

5 Illustration

The proof-as-tangle theorem (Thm. 4, §4) is not just meaningful for proof-theory: it also provides a useful coherence theorem for ribbon dialogue categories, such as the category $\mathbf{Mod}(H)$ of finite and infinite dimensional H-modules associated to a ribbon Hopf algebra H, mentioned in §2.4. By way of illustration, imagine that one wants to establish that the diagram

$$\begin{tikzcd}
\bot \multimapdotinv (\bot \multimapdotinv A) \arrow[rr, "\bot \multimapdotinv turn_A"] & & \bot \multimapdotinv (A \multimap \bot) \\
(\bot \multimapdotinv A) \multimap \bot \arrow[u, "turn_{\bot \multimapdotinv A}"] & A \arrow[l, "\eta'_A"'] \arrow[r, "\eta_A"] & \bot \multimapdotinv (A \multimap \bot) \arrow[u, "\theta_{\bot \multimapdotinv (A \multimap \bot)}"']
\end{tikzcd} \tag{6}$$

commutes in every ribbon dialogue category $\mathscr{D}$, where

$$\begin{array}{c} \eta_A \ : \ A \ \to \ \bot \multimapdotinv (A \multimap \bot) \qquad \eta'_A \ : \ A \ \to \ (\bot \multimapdotinv A) \multimap \bot \\ turn_A \ : \ \bot \multimapdotinv A \ \to \ A \multimap \bot \end{array}$$

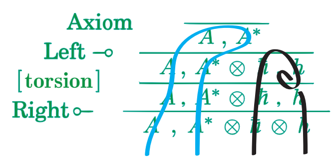
denote the units η, η' of the two double negation monads, and the canonical isomorphism $turn_A$ between the left and right negation of A. Commutativity of (6) in any ribbon dialogue category is equivalent to the fact that the following derivation trees π_1 and π_2 of ribbon tensorial logic are equal modulo commuting conversions:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A}\ \text{Axiom}}{A\,,\ A \multimap \bot \vdash \bot}\ \text{Left} \multimap}{A\,,\ A \multimap \bot \vdash \bot}\ [\mathbf{torsion}]}{A \vdash \bot \multimapinv (A \multimap \bot)}\ \text{Right} \multimapinv$$

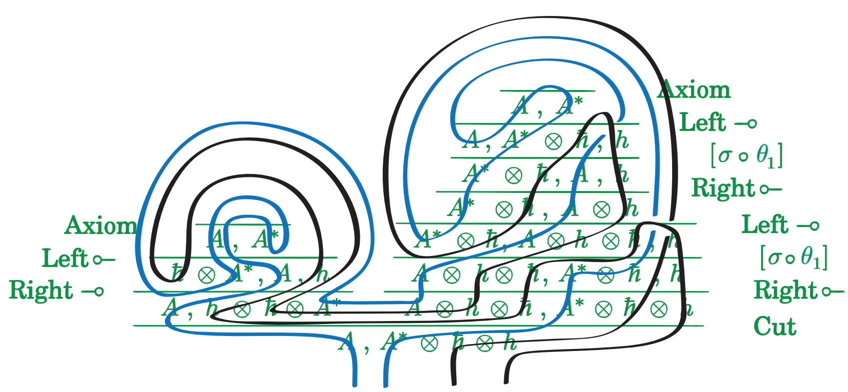
$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A}\ \text{Axiom}}{\bot \multimapinv A\,,\ A \vdash \bot}\ \text{Left} \multimapinv}{A \vdash (\bot \multimapinv A) \multimap \bot}\ \text{Right} \multimap \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A}\ \text{Axiom}}{A\,,\ A \multimap \bot \vdash \bot}\ \text{Left} \multimap}{A \multimap \bot\,,\ A \vdash \bot}\ [\sigma \circ \theta_1]}{A \multimap \bot \vdash \bot \multimapinv A}\ \text{Right} \multimapinv}{A \multimap \bot\,,\ (\bot \multimapinv A) \multimap \bot \vdash \bot}\ \text{Left} \multimap}{(\bot \multimapinv A) \multimap \bot\,,\ A \multimap \bot \vdash \bot}\ [\sigma \circ \theta_1]}{(\bot \multimapinv A) \multimap \bot \vdash \bot \multimapinv (A \multimap \bot)}\ \text{Right} \multimapinv}{A \vdash \bot \multimapinv (A \multimap \bot)}\ \text{Cut}$$

where the element $\mathbf{torsion} = \theta\langle 2 \rangle$ of $\mathbf{G}_2$ twists the two hypothesis of the proof, or (equivalently) twists the conclusion $\bot$ with an angle 2π, see §3.3 for a definition of $\theta\langle n \rangle$. One convenient way to construct the ribbon tangles $[\pi_1]$ and $[\pi_2]$ associated to the derivation trees π_1 and π_2 is to proceed by structural induction, and to interpret every derivation tree π of a sequent $A_1, \ldots, A_n \vdash B$ as a form $[\pi]$ on the object $[A_1] \otimes \ldots \otimes [A_n] \otimes [B]^*$ in the category $\mathbf{free\text{-}ribbon}(\mathscr{X} + 1)$. Recall that a *form* on an object A in a ribbon category $\mathscr{C}$, is defined as a morphism from A to the tensorial unit I. If we use the notation $\hbar = \bot$ for the tensorial pole object of $\mathbf{free\text{-}ribbon}(\mathscr{X} + 1)$ and $h = \bot^*$ for its right dual, we can then describe $[\pi_1]$ and $[\pi_2]$ as a sequence of local transformations performed on forms. When we apply this recipe to $[\pi_1]$, we obtain the following sequence of local transformations

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A\,,\ A^*}\ \text{Axiom}}{A\,,\ A^* \otimes \hbar\,,\ h}\ \text{Left} \multimap}{A\,,\ A^* \otimes \hbar\,,\ h}\ [\mathbf{torsion}]}{A\,,\ A^* \otimes \hbar \otimes h}\ \text{Right} \multimapinv$$



This intermediate representation enables us to compute the associated ribbon tangle $[\pi_1]$ by a step-by-step procedure, as done in the right-hand side figure above, see Chap. 7 of [16] for details. In the ribbon tangle $[\pi_1]$, the black strand tracks the circulation of $\hbar = \bot$ inside the proof while the blue strands tracks the circulation of the formula A. The ribbon tangle $[\pi_2]$ associated to the derivation tree π_2 is computed by the same procedure below. The fact that the two ribbon tangles $[\pi_1]$ and $[\pi_2]$ are equal modulo topological deformation implies (by Thm. 4) that the two derivation trees π_1 and π_2 are equal modulo commuting conversions. This establishes the non-trivial fact that the diagram (6) is commutative in every ribbon dialogue category.



References

[1] Samson Abramsky. Temperley-Lieb Algebra: From Knot Theory to Logic and Computation via Quantum Mechanics. *Mathematics of Quantum Computing and Technology*, G. Chen and L. Kauffman and S. Lomonaco editors, Taylor and Francis, 2007.

[2] Samson Abramsky, Radha Jagadeesan: Games and Full Completeness for Multiplicative Linear Logic. J. Symb. Log. 59(2): 543-574 (1994)

[3] Arnaud Fleury. Ribbon braided multiplicative linear logic. Matematica Contemporanea, 01/2003; 24.

[4] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[5] Jean-Yves Girard. Towards a geometry of interaction. In Categories in Computer Science and Logic, pages 69 – 108, Providence, 1989. American Mathematical Society. Proceedings of Symposia in Pure Mathematics no 92.

[6] Jean-Yves Girard. Geometry of interaction I: interpretation of system F . In Ferro, Bonotto, Valentini, and Zanardo, editors, Logic Colloquium '88, pages 221 – 260, Amsterdam, 1989. North-Holland.

[7] Jean-Yves Girard. Geometry of interaction II : deadlock-free algorithms. In Martin-Löf and Mints, editors, Proceedings of COLOG 88, volume 417 of LNCS, pages 76 – 93, Heidelberg, 1990. Springer-Verlag.

[8] Willem Heijltjes and Robin Houston. No proof nets for MLL with units: Proof equivalence in MLL is PSPACE-complete. Proc. CSL-LICS 2014.

[9] Willem Heijltjes and Lutz Straßburger. Proof nets and semi-star-autonomous categories. Mathematical Structures in Computer Science, 2014.

[10] Dominic J.D. Hughes. Simple free star-autonomous categories and full coherence. JPAA, 216(11):2386 –2410, 2012.

[11] Dominic J.D. Hughes. Simple multiplicative proof nets with units. Annals of Pure and Applied Logic, 2012.

[12] André Joyal and Ross Street. *The geometry of tensor calculus, I.* Adv. Math. 88, 55–113. (1991)

[13] André Joyal and Ross Street. *Braided tensor categories.* Adv. Math. 102, 20–78, 1993.

[14] Christian Kassel, *Quantum groups*, Graduate Texts in Mathematics 155, Springer, 1994.

[15] Paul-André Mellies. *Game semantics in string diagrams.* Proceedings LICS 2012.

[16] Paul-André Mellies. *Une étude micrologique de la négation.* Habilitation à Diriger des Recherches, 2016.

[17] Peter Selinger. A survey of graphical languages for monoidal categories. In *New Structures for Physics* (ed. Bob Coecke), Springer Lecture Notes in Physics 813, 289–355, 2011.

[18] M. C. Shum. *Tortile tensor categories.* Journal of Pure and Applied Algebra, 93:57–110, 1994.

[19] François Lamarche, Lutz Straßburger: On Proof Nets for Multiplicative Linear Logic with Units. CSL 2004: 145-159.

[20] Todd Trimble. Linear logic, bimodules, and full coherence for autonomous categories. PhD thesis, Rutgers University, 1994.

6 Appendix: the cut-elimination procedure

In this appendix, we give an exhaustive list of the commuting conversions of ribbon tensorial logic.

6.1 The commuting conversions involving an exchange rule

For the reader's convenience, we put together the commuting conversions involving the exchange rule in two independent figures (see pages 12 and 13). All the commuting conversions involving the exchange rule are described in these two figures, except for the two commuting conversions below, which involve the exchange rule and the cut rule.

Exchange vs. cut — The derivation tree

$$\dfrac{\text{Exchange}[g]\ \dfrac{\dfrac{\pi_1 \atop \vdots}{A_1,\cdots,A_n\vdash B}}{A_{g\rhd 1},\ldots,A_{g\rhd n}\vdash B} \qquad \dfrac{\pi_2 \atop \vdots}{\Upsilon_1,B,\Upsilon_2\vdash C}}{\Upsilon_1, A_{g\rhd 1},\ldots,A_{g\rhd n},\Upsilon_2\vdash C}\,\text{Cut}$$

is transformed into

$$\dfrac{\dfrac{\pi_1 \atop \vdots}{A_1,\cdots,A_n\vdash B} \qquad \dfrac{\dfrac{\pi_2 \atop \vdots}{\Upsilon_1,B,\Upsilon_2\vdash C}}{\Upsilon_1,A_1,\cdots,A_n,\Upsilon_2\vdash C}\,\text{Cut}}{\Upsilon_1, A_{g\rhd 1},\ldots,A_{g\rhd n},\Upsilon_2\vdash C}\,\text{Exchange}[h]$$

where $h = p \otimes h \otimes q$ is deduced from g and the size p and q of the two contexts Υ_1 and Υ_2.

Cut vs. exchange — The proof

$$\dfrac{\dfrac{\pi_1 \atop \vdots}{\Gamma\vdash A_i} \qquad \dfrac{\dfrac{\pi_2 \atop \vdots}{A_1,\ldots,A_n\vdash B}}{A_{g\rhd 1},\ldots,A_{g\rhd n}\vdash B}\,\text{Exchange}[g]}{A_{g\rhd 1},\ldots,A_{j-1},\Gamma,A_{j+1},\ldots,A_{g\rhd n}\vdash B}\,\text{Cut}$$

where $j = g^{-1}\rhd i$ is the unique index such that $g\rhd j = i$, is transformed into the derivation tree

$$\dfrac{\dfrac{\dfrac{\pi_1 \atop \vdots}{\Gamma\vdash A_i} \qquad \dfrac{\pi_2 \atop \vdots}{A_1,\ldots,A_n\vdash B}}{A_1,\ldots,A_{i-1},\Gamma,A_{i+1},A_n\vdash B}\,\text{Cut}}{A_{g\rhd 1},\ldots,A_{j-1},\Gamma,A_{j+1},\ldots,A_{g\rhd n}\vdash B}\,\text{Exchange}[h]$$

where h is deduced from g and the size of the context Γ.

6.2 The commuting conversions between a cut rule and a cut rule

The two derivation trees below are considered equivalent from the point of view of cut-elimination:

$$\dfrac{\dfrac{\pi_1}{\Gamma \vdash A} \qquad \dfrac{\dfrac{\pi_2}{\Upsilon_2, A, \Upsilon_3 \vdash B} \qquad \dfrac{\pi_3}{\Upsilon_1, B, \Upsilon_4 \vdash C}}{\Upsilon_1, \Upsilon_2, A, \Upsilon_3, \Upsilon_4 \vdash C}\,\text{Cut}}{\Upsilon_1, \Upsilon_2, \Gamma, \Upsilon_3, \Upsilon_4 \vdash C}\,\text{Cut}$$

$$\dfrac{\dfrac{\dfrac{\pi_1}{\Gamma \vdash A} \qquad \dfrac{\pi_2}{\Upsilon_2, A, \Upsilon_3 \vdash B}}{\Upsilon_2, \Gamma, \Upsilon_3 \vdash B}\,\text{Cut} \qquad \dfrac{\pi_3}{\Upsilon_1, B, \Upsilon_4 \vdash C}}{\Upsilon_1, \Upsilon_2, \Gamma, \Upsilon_3, \Upsilon_4 \vdash C}\,\text{Cut}$$

In particular, the cut-elimination procedure is allowed to transform the first derivation tree into the second one, and conversely. The two derivation trees below are also equivalent from the point of view of cut-elimination:

$$\dfrac{\dfrac{\pi_1}{\Gamma \vdash A} \qquad \dfrac{\dfrac{\pi_2}{\Delta \vdash B} \qquad \dfrac{\pi_3}{\Upsilon_1, A, \Upsilon_2, B, \Upsilon_3 \vdash C}}{\Upsilon_1, A, \Upsilon_2, \Delta, \Upsilon_3 \vdash C}\,\text{Cut}}{\Upsilon_1, \Gamma, \Upsilon_2, \Delta, \Upsilon_3 \vdash C}\,\text{Cut}$$

$$
\dfrac{\dfrac{\pi_2 \atop \vdots}{\Delta \vdash B} \qquad \dfrac{\dfrac{\pi_1 \atop \vdots}{\Gamma \vdash A} \qquad \dfrac{\pi_3 \atop \vdots}{\Upsilon_1, A, \Upsilon_2, B, \Upsilon_3 \vdash C}}{\Upsilon_1, \Gamma, \Upsilon_2, B, \Upsilon_3 \vdash C}\,\text{Cut}}{\Upsilon_1, \Gamma, \Upsilon_2, \Delta, \Upsilon_3 \vdash C}\,\text{Cut}
$$

6.3 The commuting conversions between an axiom rule and a cut rule

Axiom vs. cut The derivation tree

$$
\dfrac{\text{Axiom}\,\dfrac{}{A \vdash A} \qquad \dfrac{\pi \atop \vdots}{\Upsilon_1, A, \Upsilon_2 \vdash B}}{\Upsilon_1, A, \Upsilon_2 \vdash B}\,\text{Cut}
$$

is transformed into the derivation tree

$$
\dfrac{\pi \atop \vdots}{\Upsilon_1, A, \Upsilon_2 \vdash B}
$$

Cut vs. axiom The derivation tree

$$
\dfrac{\dfrac{\pi \atop \vdots}{\Gamma \vdash A} \qquad \dfrac{}{A \vdash A}\,\text{Axiom}}{\Gamma \vdash A}\,\text{Cut}
$$

is transformed into the derivation tree

$$
\dfrac{\pi \atop \vdots}{\Gamma \vdash A}
$$

6.4 The commuting conversions between a cut rule and a principal formula

The tensor product The derivation tree

$$
\dfrac{\text{Right}\otimes\,\dfrac{\dfrac{\pi_1 \atop \vdots}{\Gamma \vdash A} \qquad \dfrac{\pi_2 \atop \vdots}{\Delta \vdash B}}{\Gamma, \Delta \vdash A \otimes B} \qquad \dfrac{\dfrac{\pi_3 \atop \vdots}{\Upsilon_1, A, B, \Upsilon_2 \vdash C}}{\Upsilon_1, A \otimes B, \Upsilon_2 \vdash C}\,\text{Left}\otimes}{\Upsilon_1, \Gamma, \Delta, \Upsilon_2 \vdash C}\,\text{Cut}
$$

is transformed into the derivation tree

$$
\dfrac{\dfrac{\pi_1 \atop \vdots}{\Gamma \vdash A} \qquad \dfrac{\dfrac{\pi_2 \atop \vdots}{\Delta \vdash B} \qquad \dfrac{\pi_3 \atop \vdots}{\Upsilon_1, A, B, \Upsilon_2 \vdash C}}{\Upsilon_1, A, \Delta, \Upsilon_2 \vdash C}\,\text{Cut}}{\Upsilon_1, \Gamma, \Delta, \Upsilon_2 \vdash C}\,\text{Cut}
$$

A choice has been made here: indeed, the cut rule on the formula $A \otimes B$ is replaced by a cut rule on the formula B, followed by a cut rule on the formula A. The other order could have been followed instead, with the cut rule on A applied before the cut rule on B. However, this choice is innocuous, because the two derivations resulting from this choice are equivalent modulo the conversion rule.

Tensor unit The derivation tree

$$\dfrac{\text{Right } I\ \dfrac{}{\vdash I} \qquad \dfrac{\dfrac{\begin{array}{c}\pi\\ \vdots\end{array}}{\Upsilon_1, \Upsilon_2 \vdash A}}{\Upsilon_1, I, \Upsilon_2 \vdash A}\ \text{Left } I}{\Upsilon_1, \Upsilon_2 \vdash A}\ \text{Cut}$$

is transformed into the derivation tree

$$\dfrac{\begin{array}{c}\pi\\ \vdots\end{array}}{\Upsilon_1, \Upsilon_2 \vdash A}$$

Left negation The derivation tree

$$\dfrac{\text{Right} \multimap \dfrac{\dfrac{\begin{array}{c}\pi_1\\ \vdots\end{array}}{A, \Delta \vdash \bot}}{\Delta \vdash A \multimap \bot} \qquad \dfrac{\dfrac{\begin{array}{c}\pi_2\\ \vdots\end{array}}{\Gamma \vdash A}}{\Gamma, A \multimap \bot \vdash \bot}\ \text{Left} \multimap}{\Gamma, \Delta \vdash \bot}\ \text{Cut}$$

is transformed into the derivation tree

$$\dfrac{\dfrac{\begin{array}{c}\pi_2\\ \vdots\end{array}}{\Gamma \vdash A} \qquad \dfrac{\begin{array}{c}\pi_1\\ \vdots\end{array}}{A, \Delta \vdash \bot}}{\Gamma, \Delta \vdash \bot}\ \text{Cut}$$

Right negation The derivation tree

$$\dfrac{\text{Right} \multimapinv \dfrac{\dfrac{\begin{array}{c}\pi_1\\ \vdots\end{array}}{\Gamma, A \vdash \bot}}{\Gamma \vdash \bot \multimapinv A} \qquad \dfrac{\dfrac{\begin{array}{c}\pi_2\\ \vdots\end{array}}{\Delta \vdash A}}{\bot \multimapinv A, \Delta \vdash \bot}\ \text{Left} \multimapinv}{\Gamma, \Delta \vdash \bot}\ \text{Cut}$$

is transformed into the derivation tree

$$\dfrac{\dfrac{\begin{array}{c}\pi_2\\ \vdots\end{array}}{\Delta \vdash A} \qquad \dfrac{\begin{array}{c}\pi_1\\ \vdots\end{array}}{\Gamma, A \vdash \bot}}{\Gamma, \Delta \vdash \bot}\ \text{Cut}$$

6.5 The commuting conversions implementing an eta-expansion step

Tensor product The derivation tree

$$\dfrac{}{A \otimes B \vdash A \otimes B}\ \text{Axiom}$$

is transformed into the derivation tree

$$\dfrac{\dfrac{\dfrac{}{A \vdash A}\ \text{Axiom} \qquad \dfrac{}{B \vdash B}\ \text{Axiom}}{A, B \vdash A \otimes B}\ \text{Right}\ \otimes}{A \otimes B \vdash A \otimes B}\ \text{Left}\ \otimes$$

Left negation The derivation tree

$$\dfrac{}{A \multimap \bot \vdash A \multimap \bot}\ \text{Axiom}$$

is transformed into the derivation tree

$$\dfrac{\dfrac{\dfrac{}{A \vdash A}\ \text{Axiom}}{A, A \multimap \bot \vdash \bot}\ \text{Left} \multimap}{A \multimap \bot \vdash A \multimap \bot}\ \text{Right} \multimap$$

Right negation The derivation tree

$$\dfrac{}{\bot \multimapinv A \vdash \bot \multimapinv A}\ \text{Axiom}$$

is transformed into the derivation tree

$$\dfrac{\dfrac{\dfrac{}{A \vdash A}\ \text{Axiom}}{\bot \multimapinv A, A \vdash \bot}\ \text{Left} \multimapinv}{\bot \multimapinv A \vdash \bot \multimapinv A}\ \text{Right} \multimapinv$$

Tensor unit The derivation tree

$$\dfrac{}{I \vdash I}\ \text{Axiom}$$

is transformed into the derivation tree

$$\dfrac{\dfrac{}{\vdash I}\ \text{Right}\ I}{I \vdash I}\ \text{Left}\ I$$

6.6 The commuting conversion between a cut rule and a secondary hypothesis

Tensor product (right introduction) The derivation tree

$$\dfrac{\dfrac{\begin{matrix}\pi_1\\ \vdots\end{matrix}}{\Gamma \vdash A} \qquad \dfrac{\dfrac{\begin{matrix}\pi_2\\ \vdots\end{matrix}}{\Upsilon_1, A, \Upsilon_2 \vdash B} \qquad \dfrac{\begin{matrix}\pi_3\\ \vdots\end{matrix}}{\Delta \vdash C}}{\Upsilon_1, A, \Upsilon_2, \Delta \vdash B \otimes C}\ \text{Right}\ \otimes}{\Upsilon_1, \Gamma, \Upsilon_2, \Delta \vdash B \otimes C}\ \text{Cut}$$

is transformed into the derivation tree

$$\dfrac{\text{Cut}\;\dfrac{\dfrac{\overset{\pi_1}{\vdots}}{\Gamma \vdash A} \qquad \dfrac{\overset{\pi_2}{\vdots}}{\Upsilon_1, A, \Upsilon_2 \vdash B}}{\Upsilon_1, \Gamma, \Upsilon_2 \vdash B} \qquad \dfrac{\overset{\pi_3}{\vdots}}{\Delta \vdash C}}{\Upsilon_1, \Gamma, \Upsilon_2, \Delta \vdash B \otimes C}\;\text{Right}\otimes$$

Similarly, the derivation tree

$$\dfrac{\dfrac{\overset{\pi_1}{\vdots}}{\Gamma \vdash A} \qquad \dfrac{\dfrac{\overset{\pi_2}{\vdots}}{\Delta \vdash B} \qquad \dfrac{\overset{\pi_3}{\vdots}}{\Upsilon_1, A, \Upsilon_2 \vdash C}}{\Delta, \Upsilon_1, A, \Upsilon_2 \vdash B \otimes C}\;\text{Right}\otimes}{\Delta, \Upsilon_1, \Gamma, \Upsilon_2 \vdash B \otimes C}\;\text{Cut}$$

is transformed into the derivation tree

$$\dfrac{\dfrac{\overset{\pi_2}{\vdots}}{\Delta \vdash B} \qquad \dfrac{\dfrac{\overset{\pi_1}{\vdots}}{\Gamma \vdash A} \qquad \dfrac{\overset{\pi_3}{\vdots}}{\Upsilon_1, A, \Upsilon_2 \vdash C}}{\Upsilon_1, \Gamma, \Upsilon_2 \vdash C}\;\text{Cut}}{\Delta, \Upsilon_1, \Gamma, \Upsilon_2 \vdash B \otimes C}\;\text{Right}\otimes$$

Tensor product (left introduction) The derivation tree

$$\dfrac{\dfrac{\overset{\pi_1}{\vdots}}{\Gamma \vdash A} \qquad \dfrac{\dfrac{\overset{\pi_2}{\vdots}}{\Upsilon_1, A, \Upsilon_2, B, C, \Upsilon_3 \vdash D}}{\Upsilon_1, A, \Upsilon_2, B \otimes C, \Upsilon_3 \vdash D}\;\text{Left}\otimes}{\Upsilon_1, \Gamma, \Upsilon_2, B \otimes C, \Upsilon_3 \vdash D}\;\text{Cut}$$

is transformed into the derivation tree

$$\dfrac{\dfrac{\dfrac{\overset{\pi_1}{\vdots}}{\Gamma \vdash A} \qquad \dfrac{\overset{\pi_2}{\vdots}}{\Upsilon_1, A, \Upsilon_2, B, C, \Upsilon_3 \vdash D}}{\Upsilon_1, \Gamma, \Upsilon_2, B, C, \Upsilon_3 \vdash D}\;\text{Cut}}{\Upsilon_1, \Gamma, \Upsilon_2, B \otimes C, \Upsilon_3 \vdash D}\;\text{Left}\otimes$$

Similarly, the derivation tree

$$\dfrac{\dfrac{\overset{\pi_1}{\vdots}}{\Gamma \vdash C} \qquad \dfrac{\dfrac{\overset{\pi_2}{\vdots}}{\Upsilon_1, A, B, \Upsilon_2, C, \Upsilon_3 \vdash D}}{\Upsilon_1, A \otimes B, \Upsilon_2, C, \Upsilon_3 \vdash D}\;\text{Left}\otimes}{\Upsilon_1, A \otimes B, \Upsilon_2, \Gamma, \Upsilon_3 \vdash D}\;\text{Cut}$$

is transformed into the derivation tree

$$\dfrac{\dfrac{\dfrac{\overset{\pi_1}{\vdots}}{\Gamma \vdash C} \qquad \dfrac{\overset{\pi_2}{\vdots}}{\Upsilon_1, A, B, \Upsilon_2, C, \Upsilon_3 \vdash D}}{\Upsilon_1, A, B, \Upsilon_2, \Gamma, \Upsilon_3 \vdash D}\;\text{Cut}}{\Upsilon_1, A \otimes B, \Upsilon_2, \Gamma, \Upsilon_3 \vdash D}\;\text{Left}\otimes$$

Tensor unit (left introduction) The derivation tree

$$\dfrac{\dfrac{\begin{matrix}\pi_1\\ \vdots\end{matrix}}{\Gamma \vdash A} \qquad \dfrac{\dfrac{\begin{matrix}\pi_2\\ \vdots\end{matrix}}{\Upsilon_1, A, \Upsilon_2, \Upsilon_3 \vdash D}}{\Upsilon_1, A, \Upsilon_2, I, \Upsilon_3 \vdash D}\,\text{Left } I}{\Upsilon_1, \Gamma, \Upsilon_2, I, \Upsilon_3 \vdash D}\,\text{Cut}$$

is transformed into the derivation tree

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\pi_1\\ \vdots\end{matrix}}{\Gamma \vdash A} \qquad \dfrac{\begin{matrix}\pi_2\\ \vdots\end{matrix}}{\Upsilon_1, A, \Upsilon_2, \Upsilon_3 \vdash D}}{\Upsilon_1, \Gamma, \Upsilon_2, \Upsilon_3 \vdash D}\,\text{Cut}}{\Upsilon_1, \Gamma, \Upsilon_2, I, \Upsilon_3 \vdash D}\,\text{Left } I$$

Similarly, the derivation tree

$$\dfrac{\dfrac{\begin{matrix}\pi_1\\ \vdots\end{matrix}}{\Gamma \vdash A} \qquad \dfrac{\dfrac{\begin{matrix}\pi_2\\ \vdots\end{matrix}}{\Upsilon_1, \Upsilon_2, A, \Upsilon_3 \vdash D}}{\Upsilon_1, I, \Upsilon_2, A, \Upsilon_3 \vdash D}\,\text{Left } I}{\Upsilon_1, I, \Upsilon_2, \Gamma, \Upsilon_3 \vdash D}\,\text{Cut}$$

is transformed into the derivation tree

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\pi_1\\ \vdots\end{matrix}}{\Gamma \vdash A} \qquad \dfrac{\begin{matrix}\pi_2\\ \vdots\end{matrix}}{\Upsilon_1, \Upsilon_2, A, \Upsilon_3 \vdash D}}{\Upsilon_1, \Upsilon_2, \Gamma, \Upsilon_3 \vdash D}\,\text{Cut}}{\Upsilon_1, I, \Upsilon_2, \Gamma, \Upsilon_3 \vdash D}\,\text{Left } I$$

Left negation (left introduction) The derivation tree

$$\dfrac{\dfrac{\begin{matrix}\pi_1\\ \vdots\end{matrix}}{\Gamma \vdash A} \qquad \dfrac{\dfrac{\begin{matrix}\pi_2\\ \vdots\end{matrix}}{\Upsilon_1, A, \Upsilon_2 \vdash B}}{\Upsilon_1, A, \Upsilon_2, B \multimap \bot \vdash \bot}\,\text{Left} \multimap}{\Upsilon_1, \Gamma, \Upsilon_2, B \multimap \bot \vdash \bot}\,\text{Cut}$$

is transformed into the derivation tree

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\pi_1\\ \vdots\end{matrix}}{\Gamma \vdash A} \qquad \dfrac{\begin{matrix}\pi_2\\ \vdots\end{matrix}}{\Upsilon_1, A, \Upsilon_2 \vdash B}}{\Upsilon_1, \Gamma, \Upsilon_2 \vdash B}\,\text{Cut}}{\Upsilon_1, \Gamma, \Upsilon_2, B \multimap \bot \vdash \bot}\,\text{Left} \multimap$$

Right negation (left introduction) The derivation tree

$$\dfrac{\dfrac{\begin{matrix}\pi_1\\ \vdots\end{matrix}}{\Gamma \vdash B} \qquad \dfrac{\dfrac{\begin{matrix}\pi_2\\ \vdots\end{matrix}}{\Upsilon_1, B, \Upsilon_2 \vdash A}}{\perp \multimapinv A, \Upsilon_1, B, \Upsilon_2 \vdash \perp}\,\text{Left}\multimapinv}{\perp \multimapinv A, \Upsilon_1, \Gamma, \Upsilon_2 \vdash \perp}\,\text{Cut}$$

is transformed into the derivation tree

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\pi_1\\ \vdots\end{matrix}}{\Gamma \vdash B} \qquad \dfrac{\begin{matrix}\pi_2\\ \vdots\end{matrix}}{\Upsilon_1, B, \Upsilon_2 \vdash A}}{\Upsilon_1, \Gamma, \Upsilon_2 \vdash A}\,\text{Cut}}{\perp \multimapinv A, \Upsilon_1, \Gamma, \Upsilon_2 \vdash \perp}\,\text{Left}\multimapinv$$

Left negation (right introduction) The derivation tree

$$\dfrac{\dfrac{\begin{matrix}\pi_1\\ \vdots\end{matrix}}{\Gamma \vdash A} \qquad \dfrac{\dfrac{\begin{matrix}\pi_2\\ \vdots\end{matrix}}{\Upsilon_1, A, \Upsilon_2, B \vdash \perp}}{\Upsilon_1, A, \Upsilon_2 \vdash \perp \multimapinv B}\,\text{Right}\multimapinv}{\Upsilon_1, \Gamma, \Upsilon_2 \vdash \perp \multimapinv B}\,\text{Cut}$$

is transformed into the derivation tree

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\pi_1\\ \vdots\end{matrix}}{\Gamma \vdash A} \qquad \dfrac{\begin{matrix}\pi_2\\ \vdots\end{matrix}}{\Upsilon_1, A, \Upsilon_2, B \vdash \perp}}{\Upsilon_1, \Gamma, \Upsilon_2, B \vdash \perp}\,\text{Cut}}{\Upsilon_1, \Gamma, \Upsilon_2 \vdash \perp \multimapinv B}\,\text{Right}\multimapinv$$

Right negation (right introduction) The derivation tree

$$\dfrac{\dfrac{\begin{matrix}\pi_1\\ \vdots\end{matrix}}{\Gamma \vdash B} \qquad \dfrac{\dfrac{\begin{matrix}\pi_2\\ \vdots\end{matrix}}{A, \Upsilon_1, B, \Upsilon_2 \vdash \perp}}{\Upsilon_1, B, \Upsilon_2 \vdash A \multimap \perp}\,\text{Right}\multimap}{\Upsilon_1, \Gamma, \Upsilon_2 \vdash A \multimap \perp}\,\text{Cut}$$

is transformed into the derivation tree

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\pi_1\\ \vdots\end{matrix}}{\Gamma \vdash B} \qquad \dfrac{\begin{matrix}\pi_2\\ \vdots\end{matrix}}{A, \Upsilon_1, B, \Upsilon_2 \vdash \perp}}{A, \Upsilon_1, \Gamma, \Upsilon_2 \vdash \perp}\,\text{Cut}}{\Upsilon_1, \Gamma, \Upsilon_2 \vdash A \multimap \perp}\,\text{Right}\multimap$$

6.7 The commuting conversions between a cut rule and a secondary conclusion

Tensor product The derivation tree

$$\dfrac{\text{Left}\otimes\ \dfrac{\dfrac{\begin{array}{c}\pi_1\\ \vdots\end{array}}{\Upsilon_2, A, B, \Upsilon_3 \vdash C}}{\Upsilon_2, A \otimes B, \Upsilon_3 \vdash C} \qquad \dfrac{\begin{array}{c}\pi_2\\ \vdots\end{array}}{\Upsilon_1, C, \Upsilon_4 \vdash D}}{\Upsilon_1, \Upsilon_2, A \otimes B, \Upsilon_3, \Upsilon_4 \vdash D}\ \text{Cut}$$

is transformed into the derivation tree

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\pi_1\\ \vdots\end{array}}{\Upsilon_2, A, B, \Upsilon_3 \vdash C} \qquad \dfrac{\begin{array}{c}\pi_2\\ \vdots\end{array}}{\Upsilon_1, C, \Upsilon_4 \vdash D}}{\Upsilon_1, \Upsilon_2, A, B, \Upsilon_3, \Upsilon_4 \vdash D}\ \text{Cut}}{\Upsilon_1, \Upsilon_2, A \otimes B, \Upsilon_3, \Upsilon_4 \vdash D}\ \text{Left}\otimes$$

Tensor unit The derivation tree

$$\dfrac{\text{Left}\ I\ \dfrac{\dfrac{\begin{array}{c}\pi_1\\ \vdots\end{array}}{\Upsilon_2, \Upsilon_3 \vdash A}}{\Upsilon_2, I, \Upsilon_3 \vdash A} \qquad \dfrac{\begin{array}{c}\pi_2\\ \vdots\end{array}}{\Upsilon_1, A, \Upsilon_4 \vdash B}}{\Upsilon_1, \Upsilon_2, I, \Upsilon_3, \Upsilon_4 \vdash B}\ \text{Cut}$$

is transformed into the derivation tree

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\pi_1\\ \vdots\end{array}}{\Upsilon_2, \Upsilon_3 \vdash A} \qquad \dfrac{\begin{array}{c}\pi_2\\ \vdots\end{array}}{\Upsilon_1, A, \Upsilon_4 \vdash B}}{\Upsilon_1, \Upsilon_2, \Upsilon_3, \Upsilon_4 \vdash B}\ \text{Cut}}{\Upsilon_1, \Upsilon_2, I, \Upsilon_3, \Upsilon_4 \vdash B}\ \text{Left}\ I$$

6.8 The commuting conversions involving the left introduction of the tensor product

We describe a series of commuting conversions between left introduction and right introduction rules. These rules are generic in tensorial logic, and have nothing to do with the topological nature of ribbon tensorial logic.

Right introduction of the tensor product

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\pi_1\\ \vdots\end{array}}{\Upsilon_1, A, B, \Upsilon_2 \vdash C} \qquad \dfrac{\begin{array}{c}\pi_2\\ \vdots\end{array}}{\Delta \vdash D}}{\Upsilon_1, A, B, \Upsilon_2, \Delta \vdash C \otimes D}\ \text{Right}\otimes}{\Upsilon_1, A \otimes B, \Upsilon_2, \Delta \vdash C \otimes D}\ \text{Left}\otimes$$

is transformed into

$$
\dfrac{\dfrac{\dfrac{\pi_1}{\Upsilon_1, A, B, \Upsilon_2 \vdash C}}{\Upsilon_1, A \otimes B, \Upsilon_2 \vdash C}\ \text{Left}\otimes \qquad \dfrac{\pi_2}{\Delta \vdash D}}{\Upsilon_1, A \otimes B, \Upsilon_2, \Delta \vdash C \otimes D}\ \text{Right}\otimes
$$

Similarly, the derivation tree

$$
\dfrac{\dfrac{\dfrac{\pi_1}{\Gamma \vdash C} \qquad \dfrac{\pi_2}{\Upsilon_1, A, B, \Upsilon_2 \vdash D}}{\Gamma, \Upsilon_1, A, B, \Upsilon_2 \vdash C \otimes D}\ \text{Right}\otimes}{\Gamma, \Upsilon_1, A \otimes B, \Upsilon_2 \vdash C \otimes D}\ \text{Left}\otimes
$$

is transformed into

$$
\dfrac{\dfrac{\pi_1}{\Gamma \vdash C} \qquad \dfrac{\dfrac{\pi_2}{\Upsilon_1, A, B, \Upsilon_2 \vdash D}}{\Upsilon_1, A \otimes B, \Upsilon_2 \vdash D}\ \text{Left}\otimes}{\Gamma, \Upsilon_1, A \otimes B, \Upsilon_2 \vdash C \otimes D}\ \text{Right}\otimes
$$

Right introduction of the left negation

$$
\dfrac{\dfrac{\dfrac{\pi}{A, \Upsilon_1, B, C, \Upsilon_2 \vdash \bot}}{\Upsilon_1, B, C, \Upsilon_2 \vdash A \multimap \bot}\ \text{Right}\multimap}{\Upsilon_1, B \otimes C, \Upsilon_2 \vdash A \multimap \bot}\ \text{Left}\otimes
$$

is transformed into

$$
\dfrac{\dfrac{\dfrac{\pi}{A, \Upsilon_1, B, C, \Upsilon_2 \vdash \bot}}{A, \Upsilon_1, B \otimes C, \Upsilon_2 \vdash \bot}\ \text{Left}\otimes}{\Upsilon_1, B \otimes C, \Upsilon_2 \vdash A \multimap \bot}\ \text{Right}\multimap
$$

Right introduction of the right negation

$$
\dfrac{\dfrac{\dfrac{\pi}{\Upsilon_1, A, B, \Upsilon_2, C \vdash \bot}}{\Upsilon_1, A, B, \Upsilon_2 \vdash \bot \multimapinv C}\ \text{Right}\multimapinv}{\Upsilon_1, A \otimes B, \Upsilon_2 \vdash \bot \multimapinv C}\ \text{Left}\otimes
$$

is transformed into

$$
\dfrac{\dfrac{\dfrac{\pi}{\Upsilon_1, A, B, \Upsilon_2, C \vdash \bot}}{\Upsilon_1, A \otimes B, \Upsilon_2, C \vdash \bot}\ \text{Left}\otimes}{\Upsilon_1, A \otimes B, \Upsilon_2 \vdash \bot \multimapinv C}\ \text{Right}\multimapinv
$$

6.9 The commuting conversions involving the left introduction of the tensorial unit

Right introduction of the unit

$$\dfrac{\dfrac{}{\vdash I}\ \text{Right } I}{I \vdash I}\ \text{Left } I$$

is transformed into

$$\dfrac{}{I \vdash I}\ \text{Axiom}$$

Right introduction of the tensor

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\pi_1\\ \vdots\end{matrix}}{\Upsilon_1, \Upsilon_2 \vdash A} \qquad \dfrac{\begin{matrix}\pi_2\\ \vdots\end{matrix}}{\Delta \vdash B}}{\Upsilon_1, \Upsilon_2, \Delta \vdash A \otimes B}\ \text{Right } \otimes}{\Upsilon_1, I, \Upsilon_2, \Delta \vdash A \otimes B}\ \text{Left } I$$

is transformed into

$$\dfrac{\text{Left } I\ \dfrac{\dfrac{\begin{matrix}\pi_1\\ \vdots\end{matrix}}{\Upsilon_1, \Upsilon_2 \vdash A}}{\Upsilon_1, I, \Upsilon_2 \vdash A} \qquad \dfrac{\begin{matrix}\pi_2\\ \vdots\end{matrix}}{\Delta \vdash B}}{\Upsilon_1, I, \Upsilon_2, \Delta \vdash A \otimes B}\ \text{Right } \otimes$$

Similarly, the derivation tree

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\pi_1\\ \vdots\end{matrix}}{\Gamma \vdash A} \qquad \dfrac{\begin{matrix}\pi_2\\ \vdots\end{matrix}}{\Upsilon_1, \Upsilon_2 \vdash B}}{\Gamma, \Upsilon_1, \Upsilon_2 \vdash A \otimes B}\ \text{Right } \otimes}{\Gamma, \Upsilon_1, I, \Upsilon_2 \vdash A \otimes B}\ \text{Left } I$$

is transformed into

$$\dfrac{\dfrac{\begin{matrix}\pi_1\\ \vdots\end{matrix}}{\Gamma \vdash A} \qquad \dfrac{\dfrac{\begin{matrix}\pi_2\\ \vdots\end{matrix}}{\Upsilon_1, \Upsilon_2 \vdash B}}{\Upsilon_1, I, \Upsilon_2 \vdash B}\ \text{Left } I}{\Gamma, \Upsilon_1, I, \Upsilon_2 \vdash A \otimes B}\ \text{Right } \otimes$$

Right introduction of the left negation.

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\pi\\ \vdots\end{matrix}}{A, \Upsilon_1, \Upsilon_2 \vdash \bot}}{\Upsilon_1, \Upsilon_2 \vdash A \multimap \bot}\ \text{Right } \multimap}{\Upsilon_1, I, \Upsilon_2 \vdash A \multimap \bot}\ \text{Left } I$$

is transformed into

$$\dfrac{\dfrac{\begin{array}{c}\pi\\ \vdots\\ \hline A, \Upsilon_1, \Upsilon_2 \vdash \bot\end{array}}{A, \Upsilon_1, I, \Upsilon_2 \vdash A \multimap \bot}\ \text{Left } I}{\Upsilon_1, I, \Upsilon_2 \vdash A \multimap \bot}\ \text{Right } \multimap$$

Right introduction of the right negation.

$$\dfrac{\dfrac{\begin{array}{c}\pi\\ \vdots\\ \hline \Upsilon_1, \Upsilon_2, A \vdash \bot\end{array}}{\Upsilon_1, \Upsilon_2 \vdash \bot \multimapinv A}\ \text{Right} \multimapinv}{\Upsilon_1, I, \Upsilon_2 \vdash \bot \multimapinv A}\ \text{Left } I$$

is transformed into

$$\dfrac{\dfrac{\begin{array}{c}\pi\\ \vdots\\ \hline \Upsilon_1, \Upsilon_2, A \vdash \bot\end{array}}{\Upsilon_1, I, \Upsilon_2, A \vdash \bot}\ \text{Left } I}{\Upsilon_1, I, \Upsilon_2 \vdash \bot \multimapinv A}\ \text{Right } \multimap$$