Linearity 2018 Presentation

2. Objective

As we know, linear logic uses the of-course modality $!A$ to isolate weakening and contraction, so that only formulas with the of-course modality can have the following weakening and contraction rules.

3. Objective

The objective of this project is to isolate exchange using a modality, which behaves in a smiliar manner as the of-course modality. So only formulas with the exchange modality can have the following exchange rule.

4. Motivation

The main motivation of the project lies in the usage of process calculi.
On the left, we have the commutative tensor product, which is used to represent processes that run in parallel.
On the right, we define the non-commutative tensor product, which is to represent sequential processes.
By formalizing such an exchange modality, we would be able to specify two commutative processes as a sequential composition in either order.
The exhange modality also has a more elegant representation for sequential composition of processes.

5. Basic Approach

Our basic approach is to define a generalized, or more abstract model of Benton's LNL model by removing the exchange structural rule. Note that the exchange rule is implicit in the syntax of the contexts.
The resulting model would be a combination of two logics: the intuitionistic linear logic and Lambek calculus.

Transition: We will first briefly discuss the LNL model and then demonstrat how to extend from it to get our model.

6. Linear/Non-Linear Model

The LNL model is based on a symmetric monoidal adjunction between a cartesian closed category and a symmetric monoidal closed category.
The left functor F is from the CCC to the SMCC, and the right functor G is the other way around.
The counit is defined as the natural transformation from the composed functor FG to the identity functor of CCC.
The unit is the natural transformation from the identity functor of SMCC to the composed functor GF.

7. Linear/Non-Linear Model

Benton has proved that the monad on the CCC is strong and commutative, while the comonad on the SMCC is symmetric monoidal.
Finally, the of-course modality is defined as the composed functor FG.

8. Commutative/Non-commutative Model

We call our model commutative/non-commutative model.
And it is developed by add to the right of the SMCC a new category: a Lambek category, such that the SMCC and the Lambek category would form a new adjunction.

9. Commutative/Non-commutative Model

Following Benton's tradition, we denote the left functor from SMCC to the Lambek category as F, and right functor from the Lambek category to SMCC.
We have proved that the adjunction is monoidal, and its counit and unit are definied similarly as in LNL model.

10. Commutative/Non-commutative Model

We have also proved several properties of the adjunction.
First, as mentioned earlier, in LNL model, the monad is strong and commutative. But in our CNC model, the monad on the SMCC is strong but non-commutative.
Second, the comonad is monoidal, instead of symmetric monoidal
Finally and most importantly, we have been able to show that the co-Eilenberg Moore category on the comonad is symmetric monoidal. So a natural transformation for exchange can be easily defined.
(If anyone asks how the exchange rule is defined, refer them to Theorem 10 on page 7 in the paper.)

11. CNC Logic

Therefore, on the left of the SMCC, we have intuitionistic linear logic.
While on the right of the Lambek category, we have a mixed commutative/non-commutative Lambek calculus.

Transition: we will show some example typing rules in our CNC logic. But first, we need to clarify some notation.

12. CNC Logic: Notation

Also following Benton's traditiona, for the logic that is on the left of the adjuntion, which is intuitionistic linear logic, we use the capital letters at the end of the alphabet, such as W, X, Y, Z to denote types, the lower case letter t with subscripts for terms, the Greek letters Φ and Ψ for contexts.
As for the logic that is on the right of the adjunction, which is Lambek calculus, we use the capital letters at the beginning of the alphabet, such as A, B, C, D to denote the types, the lower case letter s with subscripts for terms, and the Greek letters Γ and Δ for contexts.
Note that in the typing judgments of the ILL part, we use commas to separate different contexts while for the Lambek calculus part, we use semicolons.
The reason is that for the ILL part, formulas can be freely exchanged while for the Lambek part, there does not exist an exchange rule for the Lambek calculus terms. So to distinguish between the two different kinds of contexts, we use different separators.