A λ-Calculus for the Lambek Calculus

Eliana S. Almeida
Depto.Conputação-UFF and PUC/RJ
e-mail: eliana@inf.puc-rio.br

Edward H. Haeusler
Depto.Informática-PUC/RJ
e-mail:hermann@inf.puc-rio.br

Puc-Rio.InfMCC40/96 Dec, 1996

Abstract: The Lambek Calculus was introduced by Lambek [2] in order to obtain effective rules to allow forming sentences and distinguishing sentences from non-sentences. Two primitives types are defined in this calculus and from these types is allowed to construct compound types to the words of the sentence.
We construct a λ-calculus for the Lambek Calculus, Curry-Howard isomorphic to the sequent calculus. Some proof-theoretical results and a denotational semantic for this calculus, showing its adequacy with regard to the calculus, are shown. A phase semantics, based on Girard's phase semantics for the Linear Logic, searching to implement a verificationistic approach for meaning in construtivism, is also shown for this calculus.

Keywords: Lambek Sintactic Calculus, λ-Calculus, Denotational Semantic, Phase Semantic, Linear Logic.

Resumo: O Lambek Calculus foi introduzido por Lambek [2] com o objetivo de obter regras efetivas que permitam formar sentenças e distinguir sentenças de não-sentenças. Dois tipos primitivos são definidos neste cálculo e, a partir destes tipos, é possível construir tipos compostos para os termos que formam a sentença.
É construido um λ-calculus para o Lambek Calculus, Curry-Horward isomorfico ao cálculo de sequentes. Alguns resultados relacionados a teoria da prova e uma semântica denotacional que mostramos ser adequada para este cálculo é apresentada. Uma semântica de fase, baseada na semântica de fase de Girard para a Lógica Linear, que busca implementar uma abordagem de verificação com significado construtivista, é também apresentada para este cálculo.

Palavras-Chave:Cálculo Sintático do Lambek, λ-Calculus, Semântica Denotacional, Semântica de Fase, Lógica Linear.

1 Introduction

This work takes part in a whole project searching the understanding of the underlying geometry of the computational process. By geometry we may guess any topological or categorical feature that can be observed within a computational process. The idea is to provide (denotational) models for languages with some restrictions on any computational concerns. We hope that these models can give us insights about the structural relationships on a process' components. In computation the concept of sequential processes plays a central rôle. An interesting fact about the Lambek calculus [2] is that it is only able to represent sequential processes (not all of them as it can be noted from its lower computability power). However, despite its computability power, a language based on Lambek-calculus's constructs could be of great help in the theoretical study of the structural (mathematical/geometrical) nature of the sequentiality in computational processes.
Within the above commented, this work implements an initial study of sequential process by defining a functional language adequated to the representation of sequentiality as a λ-calculus Curry-Howard isomorphic to the Lambek calculus ([4], [5]). A denotational model for this calculus is provided on the basis of another, although mathematical, sequential process, namely, the functional composition operation. As it is expected, an adequacy result is shown in order to assure the faithfulness of the model. This adequacy concerns the fact that the model preserves extension trought the reduction rules defined on the intensional level of the calculus. A normalization theorem is shown in order to ensure that the later agrees with the proof-theoretical results obtained by Lambek on its calculus.

With the purpose of showing a diferent kind of semantics, and hence with a distinct amount of information provided about the calculus, we show a "Tarskian" semantics based on, another sequential concept, monoids. This last issue is stronlgy influenced by the work of Girard [3], when providing a phase-semantics for the classical-linear logic, and the work of Abrusci [6], who investigate the sequent calculus and phase semantics of pure non-commutative classical linear propositional logic. The name of our structure "Quasi-Phase Spaces" is inherited from these works.

2 The Lambek Calculus

The Lambek calculus was introduced by Lambek [2] in order to obtain effective rules to allow forming sentences and distinguishing sentences from nonsentences. The calculus is so defined: Initially, is presented two primitives types: the type of sentences,$(\mathbf{s})$, and the type of nouns,$(\mathbf{n})$.From the primitives types, it is possible to form compound types. These types will describe the function of a word or expression, they are the **syntatic types**. So, given two types A and B, we have the types A/B (A **over** B) and $A \setminus B$ (A **under** B). To A/B, we should read "A modifies from the left B" and to $A \setminus B$ "A modifies from the right B". The examples below, presented in [2], will make clear the meaning of syntatic types:

- **example 1**

  John works

  $$\underbrace{n \quad n \setminus s}_{s}$$

  In this case, "John" is a name, so we can assign to it the type n. The verb "works" transforms the name "John" form the right, producing the sentence "John works". So we should assign the type $n \setminus s$ to the verb "works".

- **example 2**

  (PoorJohn) works

  $$\underbrace{\underbrace{n/n \quad n}_{n} \quad n \setminus s}_{s}$$

  Now, the adjective "Poor" transforms the name "John" into the name "Poor John" (compound name). "poor" has type n/n. The verb "works" here has the same function as the example above.

- **example 3**

  (John works) here

  $$\underbrace{\underbrace{n \quad n \setminus s}_{s} \quad s \setminus s}_{s}$$

  In this case, the name "here" transforms the sentence "John works" into the new sentence "John works here". So, we can assign the type $s \setminus s$ to the name "here".

Looking carefully these examples, it is possible to note that when we have an expression of type A/B, followed by an expression of type B, it is produced an expression of type A, and the same happens when we have an expression of type B followed by an expression of type $B \setminus A$. Symbolically:

$$(A/B)B \rightarrow A \qquad B(B \setminus A) \rightarrow A$$

Now, suppose that we have expressions x and y of type A and B, respectively. It is assigned to expression xy the type A.B. In this way, the type A/B is assigned to all expressions z such that zy has type A for any y of type B, and the type $A \setminus B$ is assigned to all expression z such that xz has type B for any x of type A.
Bellow, we present the sequent calculus introduced by Lambek [2]. To clarify, we adopted the following convention, based on convention used in [7]:

- $A \rightharpoonup B$ indicates the type $A \setminus B$
- $A \leftharpoonup B$ indicates the type A/B
- $A \otimes B$ indicates the type A.B

The Lambek Calculus

$$A \Rightarrow A \quad (identity)$$

$$\frac{A, \Gamma \Rightarrow B}{\Gamma \Rightarrow A \rightharpoonup B}(\rightharpoonup_i) \qquad \frac{\Gamma \Rightarrow A \quad B, \Delta \Rightarrow C}{\Gamma, A \rightharpoonup B, \Delta \Rightarrow C}(\rightharpoonup_e)$$

$$\frac{\Gamma, B \Rightarrow A}{\Gamma \Rightarrow A \leftharpoonup B}(\leftharpoonup_i) \qquad \frac{\Gamma \Rightarrow B \quad A, \Delta \Rightarrow C}{A \leftharpoonup B, \Gamma, \Delta \Rightarrow C}(\leftharpoonup_e)$$

$$\frac{\Gamma \Rightarrow A \quad \Delta \Rightarrow B}{\Gamma, \Delta \Rightarrow A \otimes B}(\otimes_i) \qquad \frac{\Gamma, A, B, \Delta \Rightarrow C}{\Gamma, A \otimes B, \Delta \Rightarrow C}(\otimes_e)$$

$$\frac{\Gamma \Rightarrow A \quad A, \Gamma \Rightarrow B}{\Gamma, \Delta \Rightarrow B}(cut)$$

In the next section, we will present a λ-calculus for the Lambek calculus thal we will call **λ-Lambek calculus**.

3 λ-Lambek Calculus

We construct a λ-Lambek calculus in order to have it Curry-Howard isomorphic to the sequent calculus presented in the last section. We should take care with the context to be defined to obey the order that the terms of the calculus are presented. In this way, the context is formed by a list:

1. Types:

   (a) Atomic types: A, B,...

   (b) If A and B are types, then A $\otimes$ B, A $\rightharpoonup$ B, A $\leftharpoonup$ B are types

2. Terms:

   (a) x:A[x:A] is a term, where A is a type and x is an expression.

   (b) $\dfrac{t : B[x : A, \Gamma]}{\overrightarrow{\lambda}\ x.t : A \rightharpoonup B[\Gamma]} \qquad \dfrac{t : A[\Gamma, x : B]}{\overleftarrow{\lambda}\ x.t : A \leftharpoonup B[\Gamma]}$

   (c) $\dfrac{t : A[\Gamma] \quad t` : A \rightharpoonup B[\Delta]}{A\overrightarrow{P}P\ (t, t`) : B[\Gamma, \Delta]} \qquad \dfrac{t` : B \leftharpoonup A[\Delta] \quad t : B[\Gamma]}{A\overleftarrow{P}P\ (t`, t) : A[\Delta, \Gamma]}$

   (d) $\dfrac{t : A[\Gamma] \quad t` : B[\Gamma`]}{\langle t, t` \rangle : A \otimes B[\Gamma, \Gamma`]} \qquad \dfrac{t : C[\Gamma, x : A, y : B, \Gamma`] \quad t` : A \otimes B[\Delta]}{LET((x : A, y : B) = t`)t : C[\Gamma, \Delta, \Gamma']}$

4 The Curry-Howard Isomorphism

The idea is to show the correspondence between the λ-Lambek Calculus and the Sequent Calculus. We have the following theorems:

THEOREM 4.1 *Let $\Gamma^* \Rightarrow A$ be a sequent, where $\Gamma = \{x_1 : B_1, \ldots, x_n : B_n\}$ and $\Gamma^* = \{B_1, \ldots, B_n\}$.If exists a proof of $\Gamma^* \Rightarrow A$ in the Lambek calculus then there is a term t:A[Γ] of the λ-Lambek Calculus.*

proof: By induction on the length of the proof.
base step: The identity is directly related to a term t:

$$A \Rightarrow A \rightarrow t : A[t : A]$$

induction step: assuming that the theorem is valid to all deductions indicated by the premisses from rules of sequent calculus defined above, it is supposed that joint contexts will imply a renaming of variables, extended to the terms, such that the before existing relationship regarded to variable occurences is assured. That is, when forming a context, there should be an association but the already existing. So,

1. $\dfrac{A, \Gamma^* \Rightarrow B}{\Gamma^* \Rightarrow A \rightharpoondown B}$:
   By I.H. let Π be a proof of A, $\Gamma^* \Rightarrow B$. Then, there is a term $t : [x : A, \Gamma]$.
   By construction of the term we have $\overrightarrow{\lambda}\ x.t : A \rightharpoondown B[\Gamma]$ (application of rule 2b)

2. $\dfrac{\Gamma^* \Rightarrow A \qquad B, \Delta^* \Rightarrow C}{\Gamma^*, B \leftharpoondown A, \Delta^* \Rightarrow C}$:
   By I.H., let Π be a proof of $\Gamma^* \Rightarrow A$ and $\Pi^{‘}$ be a proof of $B, \Delta^* \Rightarrow C$. Then there are the terms t:A[Γ] and d:C[y:B,Δ].
   By construction we have the term $d[y \leftarrow \overrightarrow{APP}\ (t, z)] : C[z : \Gamma, B \leftharpoondown A, \Delta]$ (application of rule 2c), where z does not occurs in either Δ or Γ (after remaning).

3. $\dfrac{\Gamma^* \Rightarrow A \qquad \Delta^* \Rightarrow B}{\Gamma^*, \Delta^* \Rightarrow A \otimes B}$.
   Let Π be a proof of $\Gamma^* \Rightarrow A$ and Π' be a proof of $\Delta^* \Rightarrow B$.
   Then, by I.H. exists terms $t : A[\Gamma]$ and Δ^* and $t' : B[\Delta]$. By application of rule 2d, we obtain $\langle t, t‘\rangle : A \otimes B[\Gamma, \Delta]$

4. $\dfrac{\Gamma^*, A, B, \Delta^* \Rightarrow C}{\Gamma^*, A \otimes B, \Delta^* \Rightarrow C}$.
   By I.H. Let Π a proof of $\Gamma^*, A, B, \Delta^* \Rightarrow C$.
   Then exists the term $t : C[\Gamma, x : A, y : B, \Delta]$. By construction, we have $LET((x : A, y : B) = z)t : C[\Gamma, z : A \otimes B, \Delta]$ where z does not occur in either Γ or Δ (application of rule 2d).

The other cases are similar.

THEOREM 4.2 *Let t:A[Γ] be a term. Then there is a proof of* $\Gamma^* \Rightarrow A$, *where* $\Gamma = \{x_1 : B_1, \ldots, x_n : B_n\}$ *and* $\Gamma^* = \{B_1, \ldots, B_n\}$

proof: by induction on the construction of the term.

5 Proof Theoretical Results

We will present the reduction rules of the λ-Lambek Calculus and relating then to the reduction steps in the normalization process.

5.1 reduction rules:

1. $\overrightarrow{APP}\ (t`, \overrightarrow{\lambda}\ x.t) : B[\Gamma, \Delta] \rhd t[x \leftarrow t`] : B[\Gamma, \Delta]$

2. $\overleftarrow{APP}\ (\overleftarrow{\lambda}\ x.t, t`) : A[\Delta, \Gamma] \rhd t[x \leftarrow t`] : A[\Delta, \Gamma]$

3. $LET((x : A, y : B) =< t_1, t_2 >)t : C[\Gamma, \Delta, \Gamma'] \rhd t[x \leftarrow t_1, y \leftarrow t_2] : C[\Gamma, \Delta, \Gamma']$

Notice that in all rules the occurence of the variables is unique in the respective terms.

5.2 The Normalization Theorem:

A normal term is defined in the usual way:

THEOREM 5.2.1 *Let t:A[Γ] be a term. Then there is a normal term t`:A[Γ] such that* $t \rhd^* t`$.

proof: by induction on the degree of a term defined bellow.

Initially, define a degree d(t) of a term t as the sum of the degrees of the redexes it contains. To define a degree of a redex, we should define a degree of a type. It is defined by:

- d(T) = 1 if T is atomic

- $d(A \rightharpoonup B) = d(A \leftharpoondown B) = d(A \otimes B) = d(A) + d(B) + 1$, to any type A and B

The degree of a redex is defined as the degree of the type corresponding to the associated reduction rule.
base step: the term does not contain redexes. So, $t \rhd t$.

induction step: assuming that the theorem is valid to terms whose degree is less than n ($n > 0$). Let p:A[Γ] be a term, we have three cases to consider:

- The greatest redex that we have in p is $\overrightarrow{APP}\ (t`, \overrightarrow{\lambda}\ x.t) : B[\Gamma, \Delta]$, where $\lambda x.t$:A $\rightharpoonup$ B and t':A. So, we can reduce then to a term t[x $\leftharpoonup$ t`]:B[Γ, Δ] where d(t[x $\leftharpoonup$ t`]) $\leq$ d(p) $-$ d(A $\rightharpoonup$ B) + d(A) $<$ d(p).

- The greatest redex that we have in p is $\overleftarrow{APP}\ (\overleftarrow{\lambda}\ x.t, t`) : A[\Delta, \Gamma]$. This case is similar to the first.

- The greatest redex that we have in p is $LET((x : A, y : B) =< t_1, t_2 >)t : C[\Gamma, \Delta, \Gamma']$. We can reduce then to a term t[x $\leftharpoonup$ t_1, y $\leftharpoonup$ t_2]:C[Γ , Δ , Γ'] where d($t[x \leftharpoonup t_1, y \leftharpoonup t_2]$) $\leq$ d(p) $-$ d(A $\otimes$ B) + d(A) + d(B) $<$ d(p).

In all cases that we have redexes is possible to apply a reduction rule and obtain a term whose degree is less than n. By induction hypothesis, the theorem follows.

6 The Denotational Semantics

To define a denotational semantics for this calculus, we should give the intuition about the meaning of types and terms. To this end, we identify our universe to sets of functions, like this:
Let B be a enumerable set of functions from $I\!N$ to $I\!N$. Define:

$$B_0 = \{f/f = g_0 \times g_1 \times ... \times g_n \text{ or } f = g_0 g_1 \circ ... \circ g_n \text{ where n} \in I\!N \text{ and } g_i \in B, i = 0, ..., n\}$$
$$B_{k+1} = \{f/f = g_0 \times g_1 \times ... \times g_n \text{ or } f = g_0 \circ g_1 \circ ... \circ g_n \text{ where n} \in I\!N \text{ and } g_i \in B_k, i = 0, ..., n\}$$

Our universe U is defined:

$$U = \bigcup_{i<\omega} B_i$$

To identify the interpretations, we define a function [[.]] that we associate each atomic type T to a set [[T]] of functions and each term t:T[Γ] to a function [[t]] from [[Γ]] to [[T]], where we will take [[Γ]] = $\otimes$ [[A]], A $\in$ Γ . Then we have:

1. Types

    - [[A]] = $\{f/f$ is a function $\}$

- $[[A \otimes B]] = \{a \times b / a \in [[A]] \text{ and } b \in [[B]]\}$
- $[[A \rightharpoonup B]] = \{c / \forall b \in [[B]], \exists a \in [[A]] \text{ such that } b = a \circ c\}$
- $[[A \leftharpoonup B]] = \{c / \forall a \in [[A]], \exists b \in [[B]] \text{ such that } a = c \circ b\}$

2. Terms : We associate each term t:T[Γ] to a function $[[t]] : [[\Gamma]] \to [[T]]$. So,

   - $[[x : A[x : A]\,]] = I_{[[A]]}$
   - $[[< t_1, t_2 >: A \otimes B[\Gamma_1, \Gamma_2]\,]] =$ f such that $\forall \gamma_1 \in [[\Gamma_1]], \forall \gamma_2 \in [[\Gamma_2]], f(\gamma_1, \gamma_2) = [[t_1 : A[[\Gamma_1]\,]]_{\gamma_1} \times [[t_2 : B[\Gamma_2]\,]]_{\gamma_2}$. We should remember that we have: $[[t_1 : A[\Gamma_1]\,]] : [[\Gamma_1]] \to [[A]]$ and $[[t_2 : B[\Gamma_2]\,]] : [[\Gamma_2]] \to [[B]]$
   - $[[\overrightarrow{\lambda}\ x.t : A \rightharpoonup B[\Gamma]\,]] =$ f such that $\forall a \in [[A]], \forall \gamma \in [[\Gamma]], f(\gamma) = g$ and $a \circ g = [[t]]_{(a,\gamma)}$. We have $[[t]] : [[A]] \otimes [[\Gamma]] \to [[B]]$
   - $[[\overleftarrow{\lambda}\ x.t : A \leftharpoonup B[\Gamma]\,]] =$ f such that $\forall b \in [[B]], \forall \gamma \in [[\Gamma]], f(\gamma) = g$ and $g \circ b = [[t]]_{(\gamma,b)}$. Here we have $[[t]] : [[\Gamma]] \otimes [[B]] \to [[A]]$
   - $[[A\overrightarrow{P}P\ (t_1, t_2) : B[\Gamma_2, \Gamma_1]\,]] =$f such that $\forall \gamma_1 \in [[\Gamma_1]], \forall \gamma_2 \in [[\Gamma_2]], f(\gamma_1, \gamma_2) = [[t_2]]_{(\gamma_2)} \circ [[t_1]]_{(\gamma_1)}$. In this case, we have: $[[t_1]] : [[\Gamma_1]] \to [[A \rightharpoonup B]]$ and $[[t_2]] : [[\Gamma_2]] \to [[A]]$
   - $[[A\overleftarrow{P}P\ (t_1, t_2) : A[\Gamma_1, \Gamma_2]\,]] =$f such that $\forall \gamma_1 \in [[\Gamma_1]], \forall \gamma_2 \in [[\Gamma_2]], f(\gamma_1, \gamma_2) = [[t_1]]_{(\gamma_1)} \circ [[t_2]]_{(\gamma_2)}$. Where we have: $[[t_1]] : [[\Gamma_1]] \to [[A \leftharpoonup B]]$ and $[[t_2]] : [[\Gamma_2]] \to [[B]]$
   - $[[LET((x : A, y : B) = t')t : C[\Gamma, \Delta, \Gamma']\,]] =$ f such that $f(\gamma, \delta, \gamma') = [[t : C[\Gamma, \Delta, \Gamma']\,]]_{(\gamma, [[t' : A \otimes B[\Delta]]]_{(\delta)}, \gamma')}$. We have $[[t']] : [[\Delta]] \to [[A \otimes B]]$ and $[[t]] : [[\Gamma]] \otimes [[A]] \otimes [[B]] \otimes [[\Gamma']] \to [[C]]$

Now, we should to show that this interpretation is actually a model for the λ-Lambek Calculus.

7 Adequacy

THEOREM 7.1 *If $M \rhd N$ then we have $[[M]] = [[N]]$.*

proof: Its sufficient to verify that all reductions (of one step) preserves the denotation. Since the equality is transitive, the theorem follows. Then, it is necessary to prove the following lemma, called **substituition lemma**:

LEMMA 7.1 *Let $t : B[\Gamma_1, x : A, \Gamma_2]$ and $t' : A[\Delta]$ be terms. Then,*

$[[t : B[\Gamma_1, x : A, \Gamma_2]\,]]_{(\gamma_1, [[t' : A[\Delta]\,]]_{(\delta)}, \gamma_2)} = [[t[x \leftarrow t' : A[\Delta]] : B[\Gamma_1, \Delta, \Gamma_2]\,]]_{(\gamma_1, \delta, \gamma_2)}$

where we can substitute t' by x in t and x is a variable of type A.

proof: Induction on complexity of t.
base step:

- $[[x : A[x : A]\,]]_{([[t' : A[\Delta]\,]]_{(\delta)})} = I_{[[A]]}([[t' : A[\Delta]\,]]_{(\delta)}) = [[t' : A[\Delta]\,]]_{(\delta)} = [[x[x \leftarrow t' : A[\Delta]] : A[\Delta]\,]]_{(\delta)}$

Induction step: Assuming that y occurs in t_1 (The cases where y occurs in t_2 are similar),we have the following cases:

- $[[< t_1, t_2 >: A \otimes B[\Gamma_1, y : C, \Gamma_2, \Gamma]\,]]_{(\gamma_1, [[t' : C[\Delta]\,]]_{(\delta)}, \gamma_2, \gamma)} =< [[t_1 : A[\Gamma_1, y : C, \Gamma_2]\,]]_{(\gamma_1, [[t' : C[\Delta]\,]]_{(\delta)}, \gamma_2)}, [[t_2 : B[\Gamma]\,]]_{(\gamma)} >$.
  By I.H., $[[t_1 : A[\Gamma_1, y : C, \Gamma_2]\,]]_{(\gamma_1, [[t' : C[\Delta]\,]]_{(\delta)}, \gamma_2)} = [[t_1[y \leftarrow t'] : A[\Gamma_1, \Delta, \Gamma_2]\,]]_{(\gamma_1, \delta, \gamma_2)}$.
  Then we have: $[[< t_1, t_2 >: A \otimes B[\Gamma_1, y : C, \Gamma_2, \Gamma]\,]]_{(\gamma_1, [[t' : C[\Delta]\,]]_{(\delta)}, \gamma_2, \gamma)} =< [[t_1[y \leftarrow t'] : A[\Gamma_1, \Delta, \Gamma_2]\,]]_{(\gamma_1, \delta, \gamma_2)}, [[t_2 : B[\Gamma]\,]]_{(\gamma)} >= [[< t_1, t_2 > [y \leftarrow t'] : A \otimes B[\Gamma_1, \Delta, \Gamma_2, \Gamma]\,]]_{(\gamma_1, \delta, \gamma_2, \gamma)}$.

- $[[\overrightarrow{APP}\ (t_2, t_1) : B[\Gamma_1, y : C, \Gamma_2]\,\Gamma]]_{(\gamma_1, [[t' : C[\Delta]\,]]_{(\delta)}, \gamma_2, \gamma)} = [[t_2 : A[\Gamma]\,]]_{(\gamma)} \circ [[t_1 : A \rightharpoonup B[\Gamma_1, y : C, \Gamma_2]\,]]_{(\gamma_1, [[t' : C[\Delta]\,]]_{(\delta)}, \gamma_2)}$.
  By I.H., $[[t_1 : A \rightharpoonup B[\Gamma_1, y : C, \Gamma_2]\,]]_{(\gamma_1, [[t' : C[\Delta]\,]]_{(\delta)}, \gamma_2)} = [[t_1[x \leftarrow t'] : A \rightharpoonup B[\Gamma_1]$.
  So, $[[\overrightarrow{APP}\ (t_2, t_1) : B[\Gamma_1, y : C\Gamma_2]\,\Gamma]]_{(\gamma_1, [[t' : C[\Delta]\,]]_{(\delta)}, \gamma_2, \gamma)} = [[t_1[x \leftarrow t'] : A \rightharpoonup B[\Gamma_1, \Delta, \Gamma_2]\,]]_{(\gamma_1, \delta, \gamma_2)} \circ [[t_2 : A[\Gamma]\,]]_{(\gamma)} = [[\overrightarrow{APP}\ (t_2, t_1)[y \leftarrow t'] : B[\Gamma_1, \Delta\Gamma_2, \Gamma]\,]]_{(\gamma_1, \delta, \gamma_2, \gamma)}$

- $[[\overrightarrow{\lambda}\ x.t : A \rightharpoonup B[\Gamma_1, y : C, \Gamma_2]\,]]_{(\gamma_1, [[t' : C[\Delta]\,]]_{(\delta)}, \gamma_2)} = g$ such that $\forall a \in [[A]], a \circ g = [[t : B[x : A, \Gamma_1, y : C, \Gamma_2]\,]]_{(a, \gamma_1, [[t' : C[\Delta]\,]]_{(\delta)}, \gamma_2)}$.
  By I.H., $[[t : B[x : A, \Gamma_1, y : C, \Gamma_2]\,]]_{(a, \gamma_1, [[t' : C[\Delta]\,]]_{(\delta)}, \gamma_2)} = [[t[y \leftarrow t'] : B[x : A, \Gamma_1, \Delta, \Gamma_2]\,]]_{(a, \gamma_1, \delta, \gamma_2)}$.
  So, we have that $[[\overrightarrow{\lambda}\ x.t : A \rightharpoonup B[\Gamma_1, y : C, \Gamma_2]\,]]_{(\gamma_1, [[t' : C[\Delta]\,]]_{(\delta)}, \gamma_2)} = [[\overrightarrow{\lambda}\ x.t[y \leftarrow t'] : A \rightharpoonup B[\Gamma_1, \Delta, \Gamma_2]\,]]_{(\gamma_1, \delta, \gamma_2)}$

- $[[LET((x : A, y : B) = t`)t : C[\Gamma_1, z : D, \Gamma_2]\,]]_{(\gamma_1, [[t' : D[\Delta]\,]]_{(\delta)}, \gamma_2)} = [[t : C[\Gamma_1, z : D, \Gamma_2]\,]]_{(\gamma_1, [[t' : D[\Delta]\,]]_{(\delta)}, \gamma_2)} = [[t[z \leftarrow t' : D[\Delta]] : C[\Gamma_1, \Delta, \Gamma_2]\,]]_{(\gamma_1, \delta, \gamma_2)}$, by I.H.
  So, $[[t[z \leftarrow t'] : C[\Gamma_1, \Delta, \Gamma_2]\,]]_{(\gamma_1, \delta, \gamma_2)} = [[LET((x : A, y : B) = t`)t[z \leftarrow t'] : C[\Gamma_1, \Delta, \Gamma_2]\,]]_{(\gamma_1, \delta, \gamma_2)}$

The others cases are similar. Then, the Lemma follows.
Now we are able to prove the theorem, in others words, we are able to verify that the denotation is preserved:

- $[[\overrightarrow{APP}\,(t`, \overrightarrow{\lambda}\ x.t) : B[\Delta, \Gamma]\,]]_{(\delta,\gamma)} = [[\overrightarrow{\lambda}\ x.t : A \rightharpoonup B[\Gamma]\,]]_{(\gamma)} \circ [[t' : A[\Delta]\,]]_{(\delta)}$. But,$[[\overrightarrow{\lambda}\ x.t : A \rightharpoonup B[\Gamma]\,]] = g$ such that $\forall a \in [[A]], a \circ g = [[t : B[x : A, \Gamma]\,]]_{([[t':A[\Delta]\,]]_{(\delta)},\gamma)} = [[t[x \leftarrow t'] : B[\Delta, \Gamma]\,]]_{(\delta,\gamma)}$. The case where we have $[[\overleftarrow{APP}\,(\overleftarrow{\lambda}\ x.t, t`) : A[\Gamma, \Delta]\,]]_{(\gamma,\delta)}$ is similar.

- $[[LET((x : A, y : B) =< t_1, t_2 >)t : C[\Gamma_1, \Delta_1, \Delta_2, \Gamma_2]\,]]_{(\gamma_1,\delta_1,\delta_2,\gamma_2)} = [[t : C[\Gamma_1, x : a, y : B, \Gamma_2]\,]]_{(\gamma_1,<[[t_1:A[\Delta_1]]_{(\delta_1)},t_2:B[\Delta_2]\,]]_{(\delta_2)}>,\gamma_2)} = [[t[x \leftarrow t_1], y \leftarrow t_2] : C[\Gamma_1, \Delta_1, \Delta_2, \Gamma_2]\,]]_{(\gamma_1,\delta_1,\delta_2,\gamma_2)}$, by application of the lemma.

8 Phase Semantics

In this section we propose a phase semantics, that we called quasi-phase semantics, based on Girard's phase semantics for the Classical Linear Logic [3] and Abrusci's phase semantics for Pure Non-Commutative Classical Linear Propositional Logic [6], searching to implement a verificationistic approach for meaning in constructivism. To this end, we introduce our quasi-phase space consisting of a non-commutative monoid M, whose elements are called phases and the neutral element is called **e** (We should observe that we do not have the set of antiphases of M).
We will give the following interpretation to the conectives: given two sets A, B $\subset$ M:

- $A \otimes B$ indicates the pairs of shape a.b such that a belongs to a set of phases that verifies A and b belongs to a set of phases that verifies B

- $A \rightharpoonup B$ will be all c such that for all a that verifies A, we have the pair a.c belonging to a set of phases that verifies B.

- $A \leftharpoondown B$ indicates all c such that for all b that verifies B, we have the pair c.b belonging to a set of phase that verifies A.

According to the remarked above, we have the following definition:

DEFINITION 8.1

- $A \otimes B = \{\ a.b \ / \ a \in A \ and \ b \in B\}$
- $A \rightharpoonup B = \{c \ / \ \forall \ a \in A, \ a.c \in B\}$
- $A \leftharpoonup B = \{c \ / \ \forall \ b \in B, \ c.b \in A\}$

We need to introduce some definitions in order to show the soundness and completeness between the Lambek calculus and this semantics.

DEFINITION 8.2 *The quasi-phase structure for the language of letters consists in a function S that associates letters to subsets of M.*

$$S : LETTER \rightarrow 2^M$$

DEFINITION 8.3 *An interpretation $\overline{S}$ in the structure S is a function that associates formulaes to subsets of M such that:*

- $\overline{S}\ (A) = S(A)$
- $\overline{S}\ (A \otimes B) = \overline{S}\ (A) \otimes \overline{S}\ (B)$
- $\overline{S}\ (A \rightharpoonup B) = \overline{S}\ (A) \rightharpoonup \overline{S}\ (B)$
- $\overline{S}\ (A \leftharpoonup B) = \overline{S}\ (A) \leftharpoonup \overline{S}\ (B)$

We will understand a formula α as being valid whenever its verification process terminates. This is realized when the null phase is among its process phases.

DEFINITION 8.4 *A formula α is valid in an interpretation $\overline{S}$ iff* $\mathbf{e} \in \overline{S}\ (\alpha)$

A formula is called a tautology whenever its verification process terminates and is free of the initial setting.

DEFINITION 8.5 *A formula α is a tautology iff* $\forall\ \overline{S}$, $\mathbf{e} \in \overline{S}\ (\alpha)$

According the implicit meaning of a sequent as having the "," as the $\otimes$ and the "$\Rightarrow$ taking the place of $\rightharpoonup$, we have:

DEFINITION 8.6 *We define an interpretation of a sequent in the following way:*

$$\bar{S}\,(\Delta \Rightarrow \alpha) \equiv \bar{S}\,(\Delta) \rightharpoonup \bar{S}\,(\alpha)$$

Where $\bar{S}\,(\Delta) = \bigotimes_{\delta \in \Delta} \bar{S}\,(\delta)$

We now are able to prove the soundness and completeness:

THEOREM 8.1 *(soundness): The Lambek sequent calculus is sound w.r.t. validity in Quasi-phase structures.*

proof: we should prove that if $\vdash_{\text{LAMBEK}} \Delta \Rightarrow \alpha$ then $\bar{S}\,(\Delta \Rightarrow \alpha)$ is valid. We will need the following lemma:

LEMMA 8.1 *$A \rightharpoonup B$ is valid iff $\forall\ \bar{S}, \bar{S}\,(A) \subset \bar{S}\,(B)$.*

proof: $\mathbf{e} \in \bar{S}\,(A \rightharpoonup B) = \bar{S}\,(A) \rightharpoonup \bar{S}\,(B) \Rightarrow \mathbf{e} \in r/\forall q(q \in S(A) \rightarrow q.r \in S(B) \Rightarrow \forall q(q \in S(A) \rightarrow q.\mathbf{e} \in S(B)) \Rightarrow S(A) \subset S(B)$.

Considering all cases,to prove the theorem, we obtain:

1. Axiom: $A \rightarrow A$ is valid iff $\forall\ \bar{S}$, $\mathbf{e} \in \bar{S}\,(A) \rightharpoonup \bar{S}\,(A)$.
   $\mathbf{e} \in \bar{S}\,(A \rightharpoonup A) \Rightarrow \mathbf{e} \in \{r/\forall q(q \in S(A) \rightarrow q.r \in S(A))\}$.
   We know that if q $\in S(A)$ than q.$\mathbf{e} \in S(A)$. So, $\mathbf{e} \in \bar{S}\,(A) \rightharpoonup \bar{S}\,(A)$.

2. $\dfrac{A, \Gamma \Rightarrow B}{\Gamma \Rightarrow A \rightharpoonup B}$
   $A, \Gamma \Rightarrow B$ is valid iff $\forall\ \bar{S}, \mathbf{e} \in \bar{S}\,(A) \otimes\ \bar{S}\,(\Gamma) \rightharpoonup \bar{S}\,(B) = \{r/\forall q(q \in S(A) \otimes S(\Gamma \rightarrow q.r \in S(B)\}$.
   If $q \in S(A) \otimes S(\Gamma)$ then q can be written like a.q', where a $\in S(A)$ and $q' \in S(\Gamma)$.
   $\Gamma \Rightarrow A \rightharpoonup B$ is valid iff $\forall\ \bar{S}, \mathbf{e} \in \otimes\ \bar{S}\,(\Gamma) \rightharpoonup (\bar{S}\,(A) \rightharpoonup \bar{S}\,(B)) = \{r'/\forall q'(q' \in S(\Gamma) \rightarrow q'.r' \in S(A) \rightharpoonup S(B)\}$.
   But q'.r' $\in S(A) \rightharpoonup S(B)$. Then, $\forall a \in S(A)$, we have a.q'.r' $\in S(B)$. Get a.q' = q, q.r'$\in S(B)$.
   So, $\mathbf{e} \in \bar{S}\,(A) \otimes\ \bar{S}\,(\Gamma) \rightharpoonup \bar{S}\,(B)) \Rightarrow \mathbf{e} \in \otimes\ \bar{S}\,(\Gamma) \rightharpoonup \bar{S}\,(A) \rightharpoonup \bar{S}\,(B))$.

3. $\dfrac{\Gamma \Rightarrow A \quad B, \Delta \Rightarrow C}{\Gamma, A \rightharpoonup B, \Delta \Rightarrow C}$

   Assuming that $\Gamma \Rightarrow A$ and $B, \Delta \Rightarrow C$ is valid. Then, $\mathbf{e} \in \overline{S}\ (\Gamma \rightharpoonup A)$ so, $\mathbf{e} \in S(\Gamma) \rightharpoonup S(A)$ and we have that $\forall q \in S(\Gamma)$, q $\in$ S(A) and $\mathbf{e}$ $\in S(B \otimes \Delta) \rightharpoonup S(C)$ implies that $\forall b \in S(B), \forall q' \in S(\Delta)$, b.q' $\in S(C)$.
   $\Gamma, A \rightharpoonup B, \Delta \rightarrow C$ is valid if $\forall q \in S(\Gamma), \forall p \in S(A \rightharpoonup B), \forall q' \in S(\Delta)$ we have q.p.q' $\in S(C)$. But $\forall q \in S(\Gamma)$, q $\in S(A)$ and q.p $\in S(B)$ that implies q.p.q' $\in S(C)$.

4. $\dfrac{\Gamma \Rightarrow A \quad \Delta \Rightarrow B}{\Gamma, \Delta \Rightarrow A \otimes B}$

   In this case, we have: $\mathbf{e} \in S(\Gamma \rightharpoonup A)$ implies that $\forall q \in s(\Gamma) \rightarrow q \in S(A)$ and $\mathbf{e} \in S(\Delta \rightharpoonup B)$. We have that, $\forall q' \in S(\Delta) \rightarrow q' \in S(B)$.
   So, by the lemma 7.1, $S(\Gamma) \subset S(A)$ and $S(\Delta) \subset S(B)$. Then, $S(\Gamma) \otimes S(\Delta) \subset S(A) \otimes S(B)$ and $\Gamma, \Delta \Rightarrow A \otimes B$ is valid.

5. $\dfrac{\Gamma, A, B, \Delta \Rightarrow C}{\Gamma, A \otimes B, \Delta \Rightarrow C}$

   Suppose that $\mathbf{e} \in \overline{S}\ (\Gamma \otimes A \otimes B \otimes \Delta \rightharpoonup C)$. So, $\overline{S}\ (\Gamma \otimes A \otimes B) \subset \overline{S}\ (C)$ that implies $\Gamma, A \otimes B, \Delta \Rightarrow C$is valid.

The other cases are similar, and the theorem follows.

THEOREM 8.2 *(completeness): The Lambek sequent calculus is complete w.r.t. quasi-phase structures.*

proof: We should prove that if α is valid then $\vdash_{\text{LAMBEK}} \Rightarrow \alpha$, that is $\forall S : LETTER \rightarrow 2^M, e \in \overline{S}\ (\alpha)$. The proof is made like usual: we will construct the following structure:

$$\text{S}(\alpha) = \{\Delta / \vdash_{\text{LAMBEK}} \Delta \Rightarrow \alpha\}$$

We will consider the monoid M =<sequences,$\emptyset$ >, as the sequence of formulas, with the operation of concatenation of sequences and the neutral element as the empty sequence. If we prove that $\overline{S}$ is an interpretation, by definition 7.3, we obtain that if α is valid then $\emptyset \in \overline{S}\ (\alpha)$ and $\Rightarrow \alpha$ is provable.Then, we have the lemma to be proved:

LEMMA 8.2 $\overline{S}$ *is an interpretation.*

proof: By induction in the length of α.
base step: α is atomic that implies $\bar{S}(\alpha) = S(\alpha)$

Induction step:

1. $\bar{S}(\alpha \otimes \beta) = \bar{S}(\alpha) \otimes \bar{S}(\beta)$.

   - $\bar{S}(\alpha) \otimes \bar{S}(\beta) \subset \bar{S}(\alpha \otimes \beta)$. If $\Delta \Rightarrow \alpha$ and $\Gamma \Rightarrow \beta$, by rule $(\otimes_i)$ of Lambek sequent calculus, $\Delta, \Gamma \Rightarrow \alpha \otimes \beta$. Then, $\bar{S}(\alpha) \otimes \bar{S}(\beta) \subset \bar{S}(\alpha \otimes \beta)$.

   - $\bar{S}(\alpha \otimes \beta) \subset \bar{S}(\alpha) \otimes \bar{S}(\beta)$. Assuming that $\exists \Delta$ such that $\Delta \Rightarrow \alpha \otimes \beta$ and $\forall \Delta_1 \in \bar{S}(\alpha), \forall \Delta_2 \in \bar{S}(\beta), \Delta \neq \Delta_1, \Delta_2$. But $\Delta_1 \Rightarrow \alpha$ and $\Delta_2 \Rightarrow \beta$ implies that $\Delta_1, \Delta_2 \Rightarrow \alpha \otimes \beta$. Then we do not have any $\Delta \neq \Delta_1, \Delta_2$ - contradiction - and $\bar{S}(\alpha \otimes \beta) \subset \bar{S}(\alpha) \otimes \bar{S}(\beta)$.

2. $\bar{S}(\alpha \rightharpoonup \beta) = \bar{S}(\alpha) \rightharpoonup \bar{S}(\beta)$

   - $\bar{S}(\alpha) \rightharpoonup \bar{S}(\beta) \subset \bar{S}(\alpha \rightharpoonup \beta)$.
     $\bar{S}(\alpha) \rightharpoonup \bar{S}(\beta) = \Delta / \forall \Gamma (\Gamma \in \bar{S}(\alpha) \rightarrow \Gamma, \Delta \in \bar{S}(\beta)$. We know that $\Gamma \in \bar{S}(\alpha)$ implies that $\Gamma \Rightarrow \alpha$ and $\Gamma, \Delta \in \bar{S}(\beta)$ implies that $\Gamma, \Delta \Rightarrow \beta$. Then, $\alpha, \Delta \Rightarrow \beta$ (cut).By $(\rightharpoonup_i)$, we have $\Delta \Rightarrow \alpha \rightharpoonup \beta$, so $\Delta \in \bar{S}(\alpha \rightharpoonup \beta)$.

   - $\bar{S}(\alpha \rightharpoonup \beta) \subset \bar{S}(\alpha) \rightharpoonup \bar{S}(\beta)$.
     Let $\Delta \in \bar{S}(\alpha \rightharpoonup \beta)$. Then, $\Delta \Rightarrow \alpha \rightharpoonup \beta$. Let $\Gamma \in \bar{S}(\alpha)$ that implies $\Gamma \Rightarrow \alpha$. We can construct the following deduction:

$$\dfrac{\Gamma \Rightarrow \alpha \qquad \dfrac{\Delta \Rightarrow \alpha \rightharpoonup \beta \qquad \dfrac{\alpha \Rightarrow \alpha \qquad \beta \Rightarrow \beta}{\alpha, \alpha \rightharpoonup \beta \Rightarrow \beta}}{\alpha, \Delta \Rightarrow \beta}}{\Gamma, \Delta \Rightarrow \beta}$$

     This implies that $\Delta \in \bar{S}(\alpha) \rightharpoonup \bar{S}(\beta)$

The other cases are similar and the theorem follows.

9 Conclusion and Further Works

The initial intended goals were obtained. However, the denotational model is still not too abstract in order to extract a good structural insight. As a next step we should abstract the denotational model in a categorical framework. This could be obtained by firstly defining the sequential product . Here we would like to comment that the existing categorial models for the lambek calculus, namely, the monoidal closed categories, allows that commutative structures may be taken as models. Our intention is to not allow this kind of model. Thus we will begin the definition by a non-commutative product in a category. Following the tradition we get the $\leftharpoonup$ as the left adjoint and the $\rightharpoonup$ as the right adjoint of the adequated sequential product functors respectively.

Another extension of this work is the inclusion of the modality that allow the permutation on the left-hand side of the sequent. The inclusion, it is clear, concerns the semantical level in the denotational approach as well as in the "Tarskian" approach.

References

[1] J.Y. Girard and P. Taylor and Y. Lafont, *Proofs and Types*,Cambridge University Press, 1991.

[2] J. Lambek, *"The Mathematics of Sentence Structure"*, in American Math., 1958, n. 65, p. 154 - 189.

[3] J.Y. Girard, *"Linear Logic"*, in Theoretical Computer Science, 1987, n. 50, p. 1 - 102.

[4] V.M. Abrusci, *"Non-Commutative Intuitionistic Linear Logic"*, in Zeitschrift fur Mathematische Logic and Grundlagen der Mathematik, 1990, n. 36, p. 297 - 318.

[5] V.M. Abrusci, *"A Comparison between Lambek Sintactic Calculus and Intuitionistic Linear Propositional Logic"*, in Zeitschrift fur Mathematische Logic and Grundlagen der Mathematik, 1990,n. 36, p. 1 - 15.

[6] V.M. Abrusci, *Phase Semantics and Sequent Calculus for Pure Non-Commutative Classical Linear Propositional Logic*, in Journal of Symbolic Logic,1991, n. 56, p. 1403 - 1451.

[7] V.C.V. de Paiva, *A Dialectica Model of the Lambek Calculus*, University of Cambridge, 1991, *unpublished.*