On Associative Lambek Calculus Extended with Basic Proper Axioms

Annie Foret

IRISA, University of Rennes 1
Campus de Beaulieu, 35042 Rennes cedex, France
foret@irisa.fr

Abstract. The purpose of this article is to show that the associative Lambek calculus extended with basic proper axioms can be simulated by the usual associative Lambek calculus, with the same number of types per word in a grammar. An analogue result had been shown for pregroups grammars [1]. We consider Lambek calculus with product, as well as the product-free version.

1 Introduction

The associative Lambek calculus (L) has been introduced in [6], we refer to [3,8] for details on (L) and its non-associative variant (NL). The pregroup formalism (PG) has been later introduced [7] as a simplification of Lambek calculus. These formalisms are considered for the syntax modeling and parsing of various natural languages. In contrast to (L), pregroups allow some kind of postulates ; we discuss this point below.

Postulates in Pregroups. The order on primitive types has been introduced in Pregroups (PG) to simplify the calculus for simple types. The consequence is that PG is not fully lexicalized. From the results in [1], this restriction is not so important because a PG using an order on primitive types can be transformed into a PG based on a simple free pregroup using a pregroup morphism, s.t. : its size is bound by the size of the initial PG times the number of primitive types (times a constant which is approximatively 4), moreover, this transformation does not change the number of types that are assigned to a word (a k-valued PG is transformed into a k-valued PG).

Postulates in the Lambek Calculus. Postulates (non-logical axioms) in Lambek calculus have also been considered. We know from [2,5], that :

(i) the associative version (L) with nonlogical axioms generate ϵ-free r.e. languages (the result also holds for L without product). The proof in the case with product is based on *binary grammars* whose production are of the form :

$$p \to q \ , \ p \to q\, r \ , \ p\, q \to r$$

C. Casadio et al. (Eds.): Lambek Festschrift, LNCS 8222, pp. 172–187, 2014.

for which is constructed a language-equivalent categorial grammar $L(\Phi(G))$ where $\Phi(G)$ is a finite set of non-logical axioms.

(ii) the non-associative version (NL) with nonlogical axioms generate context-free languages [5].

This article adresses the associative version (L). It is organized as follows : section 2 gives a short background on categorial grammars and on L extended with proper axioms $L(\Phi)$; section 3 gives some preliminary facts on $L(\Phi)$, when Φ corresponds to a preorder $\leq$ on primitive types (written $\Phi_{\leq}$) ; section 4 defines the simulation (written h) ; section 5 gives the main results on the h simulation ; section 6 gives the lemmas and proof details. Section 7 concludes.

Such a result also aims at clarifying the properties of classes of rigid and k-valued type logical grammars (TLG).

2 Categorial Grammars, their Languages and Systems

2.1 Categorial Grammars and their Languages

A *categorial grammar* is a structure $G = (\Sigma, I, S)$ where: Σ is a finite alphabet (the words in the sentences); given a set of types $Tp(Pr)$, where Pr denotes a set of primitive types, $I : \Sigma \mapsto \mathcal{P}^f(Tp(Pr))$ is a function that maps a finite set of types from each element of Σ (the possible categories of each word); $S \in Tp(Pr)$ is the *main type* associated to correct sentences.

Language. Given a relation on $Tp(Pr)^*$ called the derivation relation on types : a sentence $v_1 \ldots v_n$ then belongs to the *language of* G, written $\mathcal{L}(G)$, provided its words v_i can be assigned types X_i whose sequence $X_1 \ldots X_n$ derives S according to the derivation relation on types.

An *AB-grammar* is a categorial grammar $G = (\Sigma, I, S)$, such that its set of types $Tp(Pr)$ is constructed from Pr (primitive), using two binary connectives $/$, $\backslash$, and its language is defined using two deduction rules:

$A \,,\, A \backslash B \vdash B$ (Backward elimination, written $\backslash_e$)

$B \,/\, A \,,\, A \vdash B$ (Forward elimination, written $/_e$)

For example, using $\backslash_e$, the string of types $(N, N \backslash S)$ associated to "John swims" entails S, the type of sentences. Another typical example is $(N, \underline{((N \backslash S) / N), N)})$ associated to "John likes Mary", where the right part is associated to "likes Mary".

Lambek-grammars AB-grammars are the basis of a hierarchy of type-logical grammars (TLG). The associative Lambek calculus (L) has been introduced in [6], we refer to [3] for details on (L) and its non-associative variant (NL). A sequent-style presentation of (L) is detailed after.

The above examples illustrating AB-grammars also hold for (L) and (NL).

The pregroup formalism has been introduced in [7] as a simplification of Lambek calculus [6]. See [7] for a definition.

2.2 Type Calculus for (L)

By a sequent we mean a pair written $\Gamma \vdash A$, where Γ is a sequence of types of $Tp(Pr)$ and A is a type in $Tp(Pr)$. We give a "Gentzen style" sequent presentation, by means of introduction rules on the left or on the right of a sequent :

Lambek Calculus (associative) (Gentzen style)

$$\frac{\Gamma, A, \Gamma' \vdash C \quad \Delta \vdash A}{\Gamma, \Delta, \Gamma' \vdash C}\,\textbf{Cut} \qquad A \vdash A$$

$$\frac{\Gamma \vdash A \quad \Delta, B, \Delta' \vdash C}{\Delta, B \,/\, A, \Gamma, \Delta' \vdash C}\,/L \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash B \,/\, A}\,/R$$

$$\frac{\Gamma \vdash A \quad \Delta, B, \Delta' \vdash C}{\Delta, \Gamma, A \setminus B, \Delta' \vdash C}\,\backslash L \qquad \frac{A, \Gamma \vdash B}{\Gamma \vdash A \setminus B}\,\backslash R$$

$$\frac{\Delta, A, B, \Delta' \vdash C}{\Delta, A \bullet B, \Delta' \vdash C}\,\bullet L \qquad \frac{\Gamma \vdash A \quad \Gamma' \vdash B}{\Gamma, \Gamma' \vdash A \bullet B}\,\bullet R$$

The calculus denoted by L consists in this set of rules and has the extra requirement when applying a rule : the left-handside of a sequent cannot be empty. We may consider the system restricted to $/$ and $\backslash$ or its full version, where the set of types has a product type constructor $\bullet$ (non-commutative). The Cut rule can be eliminated from the type system (proving the same sequents). This property with the subformula property entail the decidability of the system.

2.3 Type Calculus for (L) Enriched with Postulates

$L(\Phi)$. In the general setting (as in [5]) nonlogical axioms are of the form :
$A \vdash B$, where $A, B \in Tp(Pr)$
and $L(\Phi)$ denotes the system L with all $A \vdash B$ from Φ as new axioms.

The calculus corresponds to adding a new rule of the form : $\dfrac{A \vdash B \in \Phi}{A \vdash B}\,Ax_\Phi$

$L(\Phi_\leq)$. In the following of the paper, we shall restrict to axioms of the form :
$p \vdash q$, where p, q are primitive (elements of Pr). Moreover, to keep the parallel with pregroups, we consider a preorder $\leq$ on a *finite* set of primitive types Pr and consider : $L(\Phi_\leq)$ where $\Phi_\leq$ is the set of axioms $p \vdash q$ whenever $p \leq q$, for $p, q \in$ Pr.

The calculus corresponds to adding a new rule of the form : $\dfrac{p \leq q}{p \vdash q}\,Ax_\leq$

Some Remarks and Known Facts

On Axioms. As in L, we get an equivalent version of $L(\Phi)$, where axioms $A \vdash A$ in the type calculus are supposed basic (A primitive).

A Remark on Substitutions. In general $L(\Phi)$ is not substitution closed, see [4].

Facts on Models. [4] discusses several completeness results, in particular, L is *strongly complete* with respect to *residuated semigroups* (RSG in short)[1] : the sequents provable in $L(\Phi)$ are those which are true in all RSG where all sequents from Φ are true.

3 Some Preliminary Facts with Basic Postulates

3.1 Cut Elimination and the Subformula Property

Proposition 1. *Let $\leq$ denote a preorder on the set of primitive types, and $\Phi_\leq$ denote the corresponding set of axioms. The type calculus $L(\Phi_\leq)$ admits cut elimination and the subformula property : every derivation of $\Gamma \vdash A$ in $L(\Phi_\leq)$ can be transformed into a cut-free derivation in $L(\Phi_\leq)$ of the same sequent, such that all formulas occurring in it are subformulas of this sequent.*

Proof Sketch. The proof is standard (see [8]), on derivations, by induction on (d, r) where r is the number of rules above the cut rule (to be eliminated) and d is the depth (as a subformula tree) of the cut formula (that disappears by the cut rule). The proof shows how to remove one cut having the smallest number of rules above it, by a case analysis considering the subproof $\mathcal{D}_l$ which ends at the left premise of the cut rule and the subproof $\mathcal{D}_r$ which ends at the right of the cut rule.
The only new specific case is when $\mathcal{D}_l$ and $\mathcal{D}_r$ are both axioms :

Original derivation	New derivation
$\dfrac{\dfrac{p_i \leq p_j}{p_i \vdash p_j} \qquad \dfrac{p_j \leq p_k}{p_j \vdash p_k}}{p_i \vdash p_k}\, cut$	$\dfrac{p_i \leq p_k}{p_i \vdash p_k}$

Observe that the transitivity of $\leq$ on Pr is crucial here.

Corollary 1. *Let $\leq$ denote a preorder on the set of primitive types, and $\Phi_\leq$ denote the corresponding set of axioms. The type calculus $L(\Phi_\leq)$ is decidable.*

These above propositions apply for full L and product-free L.

---

[1] A *residuated semigroup* (RSG) is a structure $(M, \leq, ., \backslash, /)$ such that $(M, \leq)$ is a nonempty poset, $\forall a, b, c \in M : a.b \leq c$ iff $b \leq a \backslash c$ iff $a \leq c / b$ (residuation), . is associative ; $\Gamma \vdash B$ is said true in a model (M, μ), where M is a RSG and μ from Pr into M iff $\mu(\Gamma) \leq \mu(B)$, where μ from Pr into M is extended as usual by $\mu(A \backslash B) = \mu(A) \backslash \mu(B)$, $\mu(A / B) = \mu(A) / \mu(B)$, $\mu(A \bullet B) = \mu(A).\mu(B)$, $\mu(\Gamma, \Delta) = \mu(\Gamma).\mu(\Delta)$.

3.2 Rule Reversibility

Proposition 2 (Reversibility of $/R$ and $\backslash R$).

In the type calculus $L(\Phi_{\leq})$: $\begin{cases} \Gamma \vdash B \,/\, A \text{ iff } \Gamma, A \vdash B \\ \Gamma \vdash A \,\backslash\, B \text{ iff } A, \Gamma \vdash B \end{cases}$

Proof ($\rightarrow$) *:* easy by induction on the derivation, according to the last rule.

This proposition holds for the full calculus and its product-free version. In the full calculus, the reversibility of rule $\bullet L$ also holds.

Main Type in the Product-Free Calculus. For a type-formula built over Pr / , \ , its main type is :

- the formula if it is primitive ;
- the main type of B if it is of the form $B \,/\, A$ or the form $A \,\backslash\, B$.

In the product-free case, any type A can thus be written (ommitting parenthesis) as $X_1 \backslash \ldots \backslash X_n \backslash p_A / Y_m / \ldots / Y_1$ where p_A is the main type of A. Reversibility then gives : $\Gamma \vdash A$ in $L(\Phi_{\leq})$ iff $X_1, ..., X_n, \Gamma, Y_m, ..., Y_1 \vdash p_A$ in $L(\Phi_{\leq})$.

3.3 Count Checks

This notion will be useful for proofs on the simulation defined in section 4.

Polarity. We first recall the notion of *polarity* of an occurrence of $p \in Pr$ in a formula : p is positive in p ; if p is positive in A, then p is positive in $B \,\backslash\, A$, $A \,/\, B$, $A \bullet B$, $B \bullet A$, and p is negative in $A \,\backslash\, B$, $B \,/\, A$; if p is negative in A, then p is negative in $B \,\backslash\, A$, $A \,/\, B$, $A \bullet B$, $B \bullet A$, and p is positive in $A \,\backslash\, B$, $B \,/\, A$.

For a sequent $\Gamma \vdash B$, the *polarity* of an occurrence of $p \in Pr$ in B is the same as its polarity in B, but the *polarity* of an occurrence of p in Γ is the opposite of its polarity in the formula of Γ.

In the presence of non-logical axioms Φ on primitive types, a *count check property* can be given as follows :

Proposition 3 (Count check in $L(\Phi)$, on primitive types). *If $\Gamma \vdash B$ is provable in $L(\Phi)$, then for each primitive type p that is not involved in any axiom $p \vdash q$ in $L(\Phi)$ where $p \neq q$: the number of positive occurrences of p in $\Gamma \vdash B$ equals the number of negative occurrences of p in $\Gamma \vdash B$.*

The proof is easy by induction on derivations.

3.4 A Duplication Method

As is the case for pregroups [1], we may propose to duplicate assignments for each primitive type occurring in a basic postulate $p_i \leq p_j$. We give more details below.

Definition 1 (polarized duplication sets).

1. *We write* $(q)^{\uparrow}_{\leq} = \{p_j \mid q \leq p_j\}$ *and* $(q)^{\downarrow}_{\leq} = \{p_j \mid p_j \leq q\}$ *for primitive types.*
2. *We use the following operations on sets of types, that extend* $/\ ,\ \backslash\ , \bullet$ *:*
   $\mathbb{T}_1 \mathbin{//} \mathbb{T}_2 = \{X_1 \,/\, X_2 \mid X_1 \in \mathbb{T}_1 \text{ and } X_2 \in \mathbb{T}_2\}$
   $\mathbb{T}_1 \mathbin{\backslash\backslash} \mathbb{T}_2 = \{X_1 \backslash X_2 \mid X_1 \in \mathbb{T}_1 \text{ and } X_2 \in \mathbb{T}_2\}$
   $\mathbb{T}_1 \bigodot \mathbb{T}_2 = \{X_1 \bullet X_2 \mid t_1 \in \mathbb{T}_1 \text{ and } X_2 \in \mathbb{T}_2\}$
   $\mathbb{T}_1 \circ \mathbb{T}_2 \circ ... \circ \mathbb{T}_n = \{X_1, X_2, ...X_n \mid X_1 \in \mathbb{T}_1\ X_2 \in \mathbb{T}_2 ... X_n \in \mathbb{T}_n\}$ *for sequences*
3. *We define the upper-duplication* $Dupl^{\uparrow}_{\leq}(.)$ *and lower-duplication* $Dupl^{\downarrow}_{\leq}(.)$ *inductively on types, for* $\delta \in \{\uparrow, \downarrow\}$*, where we write* $op(\uparrow) = \downarrow$*,* $op(\downarrow) = \uparrow$ *:*
   $Dupl^{\uparrow}_{\leq}(q) = (q)^{\uparrow}_{\leq}$ *and* $Dupl^{\downarrow}_{\leq}(q) = (q)^{\downarrow}_{\leq}$ *for primitive types.*
   $Dupl^{\delta}_{\leq}(X_1 \,/\, X_2) = Dupl^{\delta}_{\leq}(X_1) \mathbin{//} Dupl^{op(\delta)}_{\leq}(X_2)$
   $Dupl^{\delta}_{\leq}(X_1 \backslash X_2) = Dupl^{op(\delta)}_{\leq}(X_1) \mathbin{\backslash\backslash} Dupl^{\delta}_{\leq}(X_2)$
   $Dupl^{\delta}_{\leq}(X_1 \bullet X_2) = Dupl^{\delta}_{\leq}(X_1) \bigodot Dupl^{\delta}_{\leq}(X_2)$
   and $Dupl^{\delta}_{\leq}(X_1, X_2, \ldots, X_n) = Dupl^{\delta}_{\leq}(X_1) \circ Dupl^{\delta}_{\leq}(X_2) \circ ... \circ Dupl^{\delta}_{\leq}(X_n)$

This amounts to consider all replacements, according to $\leq$ *and the two polarities.*

Proposition 4 (Simulation 1). *For* $p \in Pr$ *(primitive) :*
if $\boxed{X_1, \ldots X_n \vdash p \text{ in } L(\Phi_{\leq})}$ *then* $\exists X'_1 \in Dupl^{\uparrow}_{\leq}(X_1) \ldots \exists X'_n \in Dupl^{\uparrow}_{\leq}(X_n)$ *such that* $X'_1, \ldots X'_n \vdash p$ *in L (without postulates).*

Proof Sketch. See annex.

Drawbacks. However this transformation does not preserve the size of the lexicon in general, nor the k-valued class of grammars to which the original lexicon belongs.

4 Simulation over *k*-valued Classes

4.1 Basic Definitions

Using morphisms-based encodings will enable to stay in a k-valued class and to keep a strong parse similarity (through the simulation).

Definition 2 (preorder-preserving mapping).
Let $(P, \leq)$ *and* $(P', \leq')$ *denote two sets of primitive types with a preorder. Let* h *denote a mapping from types of* $Tp(P)$ *(with* $\leq$ *on* P*) to types of* $Tp(P')$ *(with* $\leq'$ *on* P'*)*

- *h is a type-homomorphism iff*
  1. $\forall X, Y \in Tp(P) : h(X \,/\, Y) = h(X) \,/\, h(Y)$
  2. $\forall X, Y \in Tp(P) : h(X \backslash Y) = h(X) \backslash h(Y)$
  3. $\forall X, Y \in Tp(P) : h(X \bullet Y) = h(X) \bullet h(Y)$

– *h is said monotonic iff*

 4a. $\forall X, Y \in Tp(P)$:

 if $X \vdash Y$ *in* $L(\Phi_{\leq})$ *then* $h(X) \vdash h(Y)$ *in* $L(\Phi_{\leq'})$ *[Monotonicity]*

– *h is said preorder-preserving iff*

 4b. $\forall p_i, p_j \in P$: *if* $p_i \leq p_j$ *then* $h(p_i) \vdash h(p_j)$ *in* $L(\Phi_{\leq'})$.

Condition (4b) ensures (4a) for a type-homomorphism. This can be shown by induction on derivations. Next sections define and study a type-homomorphism that fullfills all these conditions.

4.2 Construction on One Component

We consider the type calculus without empty sequents on the left, and with product. The result also holds for the product-free calculus, because the constructed simulation does not add any product.

In this presentation, we allow to simulate either a fragment (represented as Pr below) or the whole set of primitive types ; for example, we may want not to transform isolated primitive types, or to proceed incrementally.

Primitive Types. Let $P = \{p_1, \ldots, p_n\}$ and $P = Pr \cup Pr'$, denote the set of primitive types, in which Pr a connex component, where no element of Pr is related by $\leq$ to an element of Pr', and each element of Pr is related by $\leq$ to another element of Pr.
We introduce new letters q_0, q_1 and β_k for each p_k of Pr (no new postulate) [2].
We take as preordered set $P' = Pr' \cup \{q_0, q_1\} \cup \{\beta_k \mid p_k \in Pr\}$,
$\leq'$ denotes the restriction of $\leq$ on Pr' (Pr' may be empty).

Notation. We write $X \vdash_{\leq'} Y$ for a sequent provable in the type calculus $\mathrm{L}(\Phi_{\leq'})$ and we write $X \vdash_{\leq} Y$ for a sequent provable in the type calculus $\mathrm{L}(\Phi_{\leq})$

We now define the simulation-morphism h for Pr as follows:

Definition 3 (Simulation-morphism h for Pr).

$h(X \,/\, Y) = h(X) \,/\, h(Y)$ $h(X \backslash Y) = h(X) \backslash h(Y)$ $h(X \bullet Y) = h(X) \bullet h(Y)$	*for* $p_i \in Pr$ *let* $Num^{\uparrow}(p_i) = \{k \mid p_i \leq p_k\} = \{i_1 \ldots i_k\}$ *s. t.* $i_1 < \ldots < i_k$ $h(p_i) = q_0 \,/\, exp(q_1, \beta_{i_1}. \ \ldots \ .\beta_{i_k})$	*for* $p_i \in Pr'$ $h(p_i) = p_i$

where

 $exp(X, \beta) = \beta \,/\, (X \backslash \beta)$

and the notation is extended to sequences on the right by :

 $exp(X, \epsilon) = X$

 $exp(X, \beta_{i_1}. \ \ldots \ .\beta_{i_{k-1}}.\beta_{i_k}) = \beta_{i_k} \,/\, (exp(X, \beta_{i_1}. \ \ldots \ .\beta_{i_{k-1}}) \backslash \beta_{i_k})$

 $= exp(exp(X, \beta_{i_1}. \ \ldots \ .\beta_{i_{k-1}}), \beta_{i_k})$

[2] q_0, q_1 can also be written q_{0Pr}, q_{1Pr} if necessary w.r.t. Pr.

Notation. In the following, in expressions of the form $exp(X, \Pi)$, Π is assumed to denote a sequence (possibly empty) $\beta_{k_1} \dots \beta_{k_n}$ (where β_k is the new letter for p_k of Pr) ; we will then write $Num(\Pi) = Num(\beta_{k_1} \dots \beta_{k_n}) = \{k_1, \dots, k_n\}$.

Fact. The h mapping of definition 3 is a type-homomorphism by construction.
Next sections will show that it is monotonic and a simulation (verifying the converse of monotonicity).

5 Main Results

Proposition 5 (Preorder-preserving property). *The homomorphism h of definition 3 satisfies : (4b.) $\forall p_i, p_j \in P$: if $p_i \leq p_j$ then $h(p_i) \vdash h(p_j)$ in $L(\Phi_{\leq'})$.*

Proof. This is a corollary of this type-raise property : $A \vdash B \,/\, (A \setminus B)$; we have $A \vdash exp(A, \Pi)$ and more generally : if $\{k \mid \beta_k \in \Pi\} \subseteq \{k \mid \beta_k \in \Pi'\}$ then $exp(A, \Pi) \vdash exp(A, \Pi')$; by construction, if $p_i \leq p_j$ then $Num^{\uparrow}(p_j) \subseteq Num^{\uparrow}(p_i)$, hence the result.

Proposition 6 (Equivalence property). *The homomorphism h of definition 3 satisfies :*

$\forall X, Y \in Tp(P) : h(X) \vdash h(Y)$ *holds in* $L(\Phi_{\leq'})$ *iff* $X \vdash Y$ *holds in* $L(\Phi_{\leq})$

Proof. For the $\leftarrow$ part, this is a corollary of the preorder-preserving property, that entails monotonicity, for a type-homomorphism. For the $\rightarrow$ part, see lemmas in the next section.

Proposition 7 (Grammar Simulation). *Given a grammar $G = (\Sigma, I, S)$ and a preorder $\leq$ on the primitive types P, we define h from types on $(P, \leq)$ to types on $(P', \leq')$ such that $P = Pr \cup Pr'$, where Pr is a connex component, as in definition 3. We construct a grammar on $(P', \leq')$ and $L(\phi_{\leq'})$ as follows :*

$G' = (\Sigma, h(I), h(S))$
where $h(I)$ is the assignment of $h(X_i)$ to a_i for $X_i \in I(a_i)$,

as a result we have : $\mathcal{L}(G) = \mathcal{L}(G')$

Note. This corresponds to the standard case of grammar, when $h(S)$ is primitive.

This proposition can apply the transformation to *the whole set* of primitive types, thus providing a fully lexicalized grammar G' (no order postulate).
A similar result holds to *a fragment* Pr of $P = Pr \cup Pr'$.

A Remark on Constructions to Avoid. For other constructions based on the same idea of chains of type-raise, we draw the attention on the fact that a simplication such as h' below would not be correct. Suppose $\Phi_{\leq}$ consists in $p_0 \leq p_1$ as postulate, define h' a type-morphism such that
$h'(p_0) = exp(q, \beta_0)$ and $h'(p_1) = exp(q, \beta_0.\beta_1)$,
this is preorder-preserving : we have $h'(p_0) \vdash h'(p_1)$,
but this is not a correct simulation, because
$h'(p_1), h'(p_0 \setminus p_0) \vdash h'(p_1)$ whereas $p_1\ (p_0 \setminus p_0) \not\vdash p_1$ (in $L(\Phi_{\leq})$).
In more details, the sequent on the left is proved by :
$h'(p_0), h'(p_0) \setminus h'(p_0), h'(p_0) \setminus \beta_1 \vdash \beta_1$,
then by $\setminus R$: $h'(p_0) \setminus h'(p_0), h'(p_0) \setminus \beta_1 \vdash h'(p_0) \setminus \beta_1$,
then by $/L$: $\beta_1 \ / \ h'(p_0) \setminus \beta_1, h'(p_0) \setminus h'(p_0), h'(p_0) \setminus \beta_1 \vdash \beta_1$, then apply $/R$.

6 Lemmas

Fact (1) [count checks for new letters]

for $X \in Tp^+(P)$: if $Y_1, h(X), Y_2 \vdash_{\leq'} Z$ and X is not empty, then :
(a) the number of positive occurrences of q_0 or q_1 in $Y_1, Y_2 \vdash Z$ equals the number of negative occurrences of q_0 or q_1 in $Y_1, Y_2 \vdash Z$
(b) the number of positive occurrences of $\alpha \in \{\beta_k \mid p_k \in Pr\}$ in $Y_1, Y_2 \vdash Z$ equals the number of negative occurrences of α in $Y_1, Y_2 \vdash Z$

Proof. (a) is a consequence of the *count check property* for q_0 and for q_1, and of the following fact : by construction, in $h(X)$ the number of positive occurrences of q_0 equals the number of negative occurrences of q_1, and the number of negative occurrences of q_0 equals the number of positive occurrences of q_1. (b) is a consequence of the *count check property* for α, and of the following fact : by construction, $h(X)$ has the same number of positive occurrences of $\alpha \in \{\beta_k \mid p_k \in Pr\}$ as its number of negative occurrences.

Note. Thus by (a), the presence of a formula $h(X)$ in a sequent imposes some equality constraints on the counts of q_0 and q_1.

Fact (2) [interactions with new letters]

for $X \in Tp^*(P)$ and $\alpha, \alpha' \in \{q_0, q_1\} \cup \{\beta_k \mid p_k \in Pr\}$:
(a) $h(X), \alpha \vdash_{\leq'} \alpha'$ is impossible when X is not empty, unless $(\alpha, \alpha') = (q_1, q_0)$
(b) $h(X), \alpha \vdash_{\leq'} exp(q_1, \Pi)$ where $\Pi \neq \epsilon$ implies X is empty and $\alpha = q_1$
(c) $h(X), \alpha, exp(q_1, \Pi") \setminus \beta \vdash_{\leq'} \beta$, where $\beta \in \{\beta_k \mid p_k \in Pr\}$ implies X is empty and $\alpha = q_1$

Proof. The proof is technical, see Annex.

Fact (3) [chains of type-raise]

if $exp(q_1, \Pi') \vdash_{\leq'} exp(q_1, \Pi'')$ then $Num(\Pi') \subseteq Num(\Pi'')$

Proof. We show a simpler version when $\Pi' = \beta_{k_1}$ (the general case follows from type-raise properties $A \vdash_{\leq'} exp(A, \Pi)$; also if Π' is empty, the assertion is obvious).

We proceed by induction on the length of Π'' and consider $exp(q_1, \beta_{k_1}) \vdash_{\leq'} exp(q_1, \Pi'')$, that is $\beta_{k_1} / (q_1 \setminus \beta_{k_1}) \vdash_{\leq'} exp(q_1, \Pi'')$. The case Π'' empty is impossible ; we write $\Pi'' = \Pi_2.\beta_{k_2}$; the sequent is $\beta_{k_1} / (q_1 \setminus \beta_{k_1})\,,\ exp(q_1, \Pi_2) \setminus \beta_{k_2} \vdash_{\leq'} \beta_{k_2}$; the end of the derivation has two possibilities:

- $$\frac{\boxed{\beta_{k_1} / (q_1 \setminus \beta_{k_1}) \vdash_{\leq'} exp(q_1, \Pi_2)} \qquad \beta_{k_2} \vdash_{\leq'} \beta_{k_2}}{\beta_{k_1} / (q_1 \setminus \beta_{k_1}),\ exp(q_1, \Pi_2) \setminus \beta_{k_2} \vdash_{\leq'} \beta_{k_2}}$$
  we get in this case the assertion by rec. : $k_1 \in Num(\Pi_2)$ ($\subseteq Num(\Pi$”))
  or
- $$\frac{exp(q_1, \Pi_2) \setminus \beta_{k_2} \vdash_{\leq'} (q_1 \setminus \beta_{k_1}) \qquad \boxed{\beta_{k_1} \vdash_{\leq'} \beta_{k_2}}}{\beta_{k_1} / (q_1 \setminus \beta_{k_1}),\ exp(q_1, \Pi_2) \setminus \beta_{k_2} \vdash_{\leq'} \beta_{k_2}}$$
  From which we get the assertion : $k_1 = k_2$ ($\in Num(\Pi$”)).

Main Lemma

(main) if $h(X) \vdash_{\leq'} h(Y)$ then $X \vdash_{\leq} Y$ (where X and Y in $Tp^+(P)$).

Sketch of Proof. We distinguish several cases, depending on the form of Y and of $h(Y)$, and proceed by (joined) induction on the total number of connectives in X, Y :

- for cases where Y is primitive, we recall that $P = Pr \cup Pr'$, where Pr is a connex component and $\leq'$ has no postulate on Pr ; there are two subcases (detailed later) depending on $p_i \in Pr$ or $p_i \in Pr'$:
  (o) for $p_i \in Pr'$ and $X \in Tp^+(P)$: $h(X) \vdash_{\leq'} p_i$ implies $X \vdash_{\leq} p_i$
  (i) if $h(X) \vdash_{\leq'} q_0 / exp(q_1, \Pi')$ then $\forall k \in Num(\Pi') : X \vdash_{\leq} p_k$
  where Π' is a sequence of β_{k_j} (this corresponds to $Y = p_i \in Pr$)
  we will show (ii) an equivalent version of (i) as follows :
  (ii) if $h(X),\ exp(q_1, \Pi') \vdash_{\leq'} q_0$ then $\forall k \in Num(\Pi') : X \vdash p_k$
  (see proof details after for (o) (i) (ii))
- (iii) if $h(Y)$ is of the form $h(D / C)$ and $Y = D / C$
  $h(X) \vdash_{\leq'} h(Y)$ iff $h(X), h(C) \vdash_{\leq'} h(D)$
  by induction $X,\ C \vdash_{\leq} D$ hence $X \vdash_{\leq} D / C$ by the $/R$ right rule
- (iv) if $h(Y)$ of the form $h(C \setminus D)$, $Y = C \setminus D$, the case is similar to (iii)
- (v) if $h(Y)$ of the form $h(C \bullet D)$ *(see proof details after, partly similar to (o))*

Main Lemma Part (o)
(o) for $p_i \in Pr'$ and $X \in Tp^+(P)$: $h(X) \vdash_{\leq'} p_i$ implies $X \vdash_{\leq} p_i$

Proof Details: we discuss on the derivation ending for $h(X) \vdash_{\leq'} p_i$:

- if this is an axiom $h(X) = p_i = h(p_i) = X$
- if this is inferred from a postulate on Pr', $p_j \leq p_i$ then also $X = p_j \vdash_{\leq} p_i$
- if $/L$ is the last rule, there are two cases
  - if the rule introduces $h(B) \,/\, h(A)$, s. t. X has the form $X{=}\Delta,\ B \,/\, A,\ \Gamma,\ \Delta'$

    $$\frac{h(\Gamma) \vdash_{\leq'} h(A) \qquad h(\Delta)\ h(B)\ h(\Delta') \vdash_{\leq'} p_i}{h(\Delta),\, h(B) \,/\, h(A),\, h(\Gamma),\, h(\Delta') \vdash_{\leq'} p_i} \qquad \frac{\boxed{by\ rec.\ (main{+}(o)):}\ \Gamma \vdash_{\leq} A \qquad \Delta,\, B,\, \Delta' \vdash_{\leq} p_i}{by\ rule\ /L : \Delta,\, B \,/\, A,\, \Gamma,\, \Delta' \vdash_{\leq} p_i}$$
  - if the rule introduces $h(p_i) = q_0 \,/\, exp(q_1, \Pi')$,

    the end is of the form $\dfrac{\boxed{h(\Gamma) \vdash_{\leq'} exp(q_1, \Pi')} \qquad h(\Delta),\, q_0,\, h(\Delta') \vdash' p_i}{h(\Delta),\, q_0 \,/\, exp(q_1, \Pi'),\, h(\Gamma),\, h(\Delta') \vdash_{\leq'} p_i}$

    which is impossible according to $\boxed{\text{Fact (1)}}$
- if $\backslash L$ is the last rule, the case is similar to the first subcase for $/L$ above
- if the last rule is $\bullet L$ introducing $h(A) \bullet h(B)$, we apply rec. (o) to the antecedent, then $\bullet L$
- the right rules are impossible ∎

Main Lemma Part (v) for $X \in Tp^+(P)$: $h(X) \vdash_{\leq'} h(C_1 \bullet C_2)$ implies $X \vdash_{\leq} C_1 \bullet C_2$

Proof Details: we discuss on the derivation ending for $h(X) \vdash_{\leq'} h(Y)$ where $Y = C_1 \bullet C_2$:

- this cannot be an axiom, a postulate, $/R$, or $\backslash R$
- if $/L$ is the last rule, there are two cases
  - if the rule introduces $h(B) \,/\, h(A)$, s. t. X has the form $X = \Delta,\ B \,/\, A,\ \Gamma,\ \Delta'$

    $$\frac{h(\Gamma) \vdash_{\leq'} h(A) \qquad h(\Delta)\ h(B)\ h(\Delta') \vdash_{\leq'} h(Y)}{h(\Delta),\, h(B) \,/\, h(A),\, h(\Gamma),\, h(\Delta') \vdash_{\leq'} h(Y)} \qquad \frac{\boxed{by\ rec.\ (main{+}(v)):}\ \Gamma \vdash_{\leq} A \qquad \Delta\ B\ \Delta' \vdash_{\leq} Y}{by\ rule\ /L : \Delta,\, B \,/\, A,\, \Gamma,\, \Delta' \vdash_{\leq} Y}$$
  - if the rule introduces $h(p_i) = q_0 \,/\, exp(q_1, \Pi')$,

    the end is of the form $\dfrac{\boxed{h(\Gamma) \vdash_{\leq'} exp(q_1, \Pi')} \qquad h(\Delta),\, q_0,\, h(\Delta') \vdash' h(Y)}{h(\Delta),\, q_0 \,/\, exp(q_1, \Pi'),\, h(\Gamma),\, h(\Delta') \vdash_{\leq'} h(Y)}$

    which is impossible according to $\boxed{\text{Fact (1)}}$
- if $\backslash L$ is the last rule, the case is similar to the first subcase for $/L$ above
- if the last rule is $\bullet L$ introducing $h(A) \bullet h(B)$, we apply rec. (v) to the antecedent, then $\bullet L$
- if the last rule is $\bullet R$ introducing $h(C_1) \bullet h(C_2)$ then X has the form Δ, Δ', such that :

$$\frac{h(\Delta) \vdash_{\leq'} h(C_1) \qquad h(\Delta') \vdash_{\leq'} h(C_2)}{h(\Delta),\, h(\Delta') \vdash_{\leq'} h(Y)} \qquad \frac{\boxed{by\ rec.\ (main):}\ \Delta \vdash_{\leq} C_1 \qquad \Delta' \vdash_{\leq} C_2}{by\ rule\ \bullet R : \Delta,\, \Delta' \vdash_{\leq} Y}$$

∎

Main Lemma Part (ii) if $h(X),\ exp(q_1, \Pi') \vdash_{\leq'} q_0$ then $\forall k \in Num(\Pi') : X \vdash p_k$

Proof Details : we show a simpler version when $\Pi' = \beta_{k_1}$ (the general case follows from type-raise properties $A \vdash_{\leq'} exp(A, \Pi)$, and if Π' is empty, the assertion is obvious). The sequent is $h(X),\ \beta_{k_1} \,/\, (q_1 \setminus \beta_{k_1}) \vdash_{\leq'} q_0$:

- if $/\,L$ is the last rule, there are two cases (it cannot introduce $exp(q_1, \Pi')$ being rightmost)
  - if the rule introduces $\boxed{h(B) \,/\, h(A)}$, s. t. X has the form $X = \Delta,\ B \,/\, A,\ \Gamma,\ \Delta'$ there are two subcases :

$$\frac{h(\Gamma) \vdash_{\leq'} h(A) \qquad h(\Delta),\ h(B),\ h(\Delta'),\ exp(q_1,\Pi') \vdash_{\leq'} q_0}{h(\Delta),\ h(B) \,/\, h(A),\ h(\Gamma),\ h(\Delta'),\ exp(q_1,\Pi') \vdash_{\leq'} q_0} \quad \boxed{\text{by global rec + rec (ii)}}: \quad \frac{\Gamma \vdash A \qquad \Delta,\ B,\ \Delta' \vdash p_{k_1}}{\textit{by rule } /L : \Delta,\ B \,/\, A,\ \Gamma,\ \Delta' \vdash p_{k_1}}$$

or

$$\frac{h(\Gamma), h(\Delta'),\ exp(q_1,\Pi') \vdash_{\leq'} h(A) \qquad \boxed{h(\Delta),\ h(B) \vdash_{\leq'} q_0}}{h(\Delta),\ h(B) \,/\, h(A),\ h(\Gamma),\ h(\Delta'),\ exp(q_1,\Pi') \vdash_{\leq'} q_0} \quad \text{impossible, see } \boxed{\text{Fact (1)}}:$$

  - if the rule introduces $\boxed{h(p_i) = q_0 \,/\, exp(q_1, \Pi'')}$, in $h(X)$, s. t. X has the form $X = \Delta,\ p_i,\ \Gamma,\ \Delta'$

- if $\dfrac{\boxed{h(\Gamma) \vdash_{\leq'} exp(q_1,\Pi'')} \qquad h(\Delta),\ q_0,\ h(\Delta'),\ exp(q_1,\Pi') \vdash' q_0}{h(\Delta),\ q_0 \,/\, exp(q_1,\Pi''),\ h(\Gamma),\ h(\Delta'),\ exp(q_1,\Pi') \vdash' q_0}$ impossible, see $\boxed{\text{Fact (1)}}$

- if $\dfrac{\boxed{h(\Gamma), h(\Delta'), exp(q_1,\Pi') \vdash_{\leq'} exp(q_1,\Pi'')} \qquad h(\Delta), q_0 \vdash_{\leq'} q_0}{h(\Delta), q_0 \,/\, exp(q_1,\Pi''), h(\Gamma), h(\Delta'), exp(q_1,\Pi') \vdash_{\leq'} q_0}$ Γ, Δ', Δ are empty by Fact (2) and $Num(\Pi') = \{\beta_{k_1}\} \subseteq Num(\Pi'')$ by Fact (3), we get $X = p_i \leq p_{k_1}$

- if $\setminus\, L$ is the last rule, it introduces $h(A) \setminus h(B)$, similar to the first subcase for $/L$ above

$$\frac{h(\Gamma) \vdash_{\leq'} h(A) \qquad h(\Delta),\ h(B),\ h(\Delta'),\ exp(q_1,\Pi') \vdash_{\leq'} q_0}{h(\Delta),\ h(\Gamma),\ h(A) \setminus h(B),\ h(\Delta'),\ exp(q_1,\Pi') \vdash_{\leq'} q_0} \quad \boxed{\text{by global rec + rec (ii)}}: \quad \frac{\Gamma \vdash A \qquad \Delta,\ B,\ \Delta' \vdash p_{k_1}}{\textit{by rule } \setminus L : \Delta,\ \Gamma,\ A \setminus B,\ \Delta' \vdash p_{k_1}}$$

- if the last rule is $\bullet L$ introducing $h(A) \bullet h(B)$, we apply rec. (ii) to the antecedent, then $\bullet L$
- the right rules and the axiom rule are impossible ■

7 Conclusion and Discussion

Former Work in Pregroups. The order on primitive types has been introduced in PG to simplify the calculus for simple types. The consequence is that PG is not fully lexicalized. We had proven in [1] that this restriction is not so important because a PG using an order on primitive types can be transformed into a PG based on a simple free pregroup using a pregroup morphism, s.t. :

- *its size* is bound by the size of the initial PG times the number of primitive types (times a constant which is approximatively 4),
- moreover, this transformation does not change the number of types that are assigned to a word (a k-valued PG is transformed into a k-valued PG).

The AB case. In constrast to pregroups (and L) rigid AB-grammars with basic postulates are more expressive than rigid AB-grammars as shown by the following language ; let $L = \{a, ab\}$, and $G = \{a \mapsto x, b \mapsto y\}$ where $x, y \in Tp(Pr)$, suppose T_1, T_2 are parse trees using G, for a and ab respectively

- in the absence of postulates, we have from T_1 and T_2 : $y = x \setminus x$ in which case *abb* should also belong to the language, contradiction;
- if basic postulates are allowed, we can take $x = S_1$ and then $y = S_1 \setminus S$, with $S_1 \leq S$, generating $L = \{a, ab\}$.

$L = \{a, ab\}$ cannot be handled by a rigid AB-grammar without postulate, whereas it is with postulates.

A similar situation might hold for extensions based on AB, such as Categorial Dependency Grammars (CDG).

In L and Related Formalisms. The work in this paper shows a result similar to [1], for L extended with an order on primitive types. The result holds for both versions with or without product. *A similar result should hold for NL and some other related calculi*, but it does not hold for AB as shown above.

Such a simulation result aims at clarifying properties of the extended calculus, in particular in terms of generative capacity and hierarchies of grammars. Another interest of the extended calculus is to allow some parallels in grammar design (type assignments, acquisition methods) between both frameworks (pregroups and (L)).

References

1. Béchet, D., Foret, A.: Fully lexicalized pregroup grammars. In: Leivant, D., de Queiroz, R. (eds.) WoLLIC 2007. LNCS, vol. 4576, pp. 12–25. Springer, Heidelberg (2007)
2. Buszkowski, W.: Some decision problems in the theory of syntactic categories. Zeitschrift f. Math. Logik u. Grundlagen der Mathematik 28, 539–548 (1982)
3. Buszkowski, W.: Mathematical linguistics and proof theory. In: van Benthem, J., ter Meulen, A. (eds.) Handbook of Logic and Language, ch. 12, pp. 683–736. North-Holland Elsevier, Amsterdam (1997)
4. Buszkowski, W.: Type Logics in Grammar. Trends in logic. Studia Logica Library, vol. 21, pp. 321–366. Springer (2003)
5. Buszkowski, W.: Lambek calculus with nonlogical axioms. In: Language and Grammar, Studies in Mathematical Linguistics and Natural Language, pp. 77–93. CSLI Publications (2005)
6. Lambek, J.: The mathematics of sentence structure. American Mathematical Monthly 65 (1958)
7. Lambek, J.: Type grammars revisited. In: Lecomte, A., Perrier, G., Lamarche, F. (eds.) LACL 1997. LNCS (LNAI), vol. 1582, pp. 1–27. Springer, Heidelberg (1999)
8. Moot, R., Retoré, C.: The Logic of Categorial Grammars. LNCS, vol. 6850, pp. 23–63. Springer, Heidelberg (2012)

Annex : Details of Proofs

Proof sketch for the Simulation based on Duplication. We write $\vdash_{\leq}$ for $\vdash$ in $L(\Phi_{\leq})$, and we consider a version where axioms $A \vdash A$ are such that A is primitive.

In full L. We show by induction on the length of derivation of $\Gamma \vdash_{\leq} Z$ in $L(\Phi_{\leq})$ without cut, that more generally : if $\Gamma \vdash_{\leq} Z$ and $\Gamma = X_1, \ldots X_n$ then
(a) if Z is primitive then $\exists X'_1 \in Dupl^{\uparrow}_{\leq}(X_1) \ldots \exists X'_n \in Dupl^{\uparrow}_{\leq}(X_n)\ X'_1, \ldots X'_n \vdash Z$ in L (without postulates)
(b) $\exists X'_1 \in Dupl^{\uparrow}_{\leq}(X_1) \ldots \exists X'_n \in Dupl^{\uparrow}_{\leq}(X_n)$ and $\exists Z^- \in Dupl^{\downarrow}_{\leq}(Z)$ such that : $X'_1, \ldots X'_n \vdash Z^-$ in L (without postulates)
We first show (b) separately :
- in the axiom case $p \vdash p$ in $L(\Phi_{\leq})$: we take $p \vdash p$ in L
- in the non-logical axiom case $p \vdash q$, where $p \leq q$, in $L(\Phi_{\leq})$: we take $q \vdash q$ in L
- for a rule introducing $/$, $\backslash$ or $\bullet$ on the right (b) is shown easily by rec. on the antecedent then the same rule in L, because for $A' \in Dupl^{\uparrow}_{\leq}(A)$ and $B^- \in Dupl^{\downarrow}_{\leq}(B)$, we get $A' \backslash B^- \in Dupl^{\downarrow}_{\leq}(A \backslash B)$ and $B^- / A' \in Dupl^{\downarrow}_{\leq}(B / A)$ and for $A^- \in Dupl^{\downarrow}_{\leq}(A)$ and $B^- \in Dupl^{\downarrow}_{\leq}(B)$, we get $A^- \bullet B^- \in Dupl^{\downarrow}_{\leq}(A \bullet B)$
- we detail the $/L$ case :

$$\frac{\Gamma \vdash_{\leq} A \quad \Delta_1, B, \Delta_2 \vdash_{\leq} Z}{\Delta_1, B / A, \Gamma, \Delta_2 \vdash_{\leq} Z}\, /L \qquad \begin{array}{l} \text{by rec. } \exists \Gamma' \in Dupl^{\uparrow}_{\leq}(\Gamma), \\ \exists A^- \in Dupl^{\downarrow}_{\leq}(A), \exists Z^- \in Dupl^{\downarrow}_{\leq}(Z) \\ \exists \Delta'_i \in Dupl^{\uparrow}_{\leq}(\Delta_i), \exists B' \in Dupl^{\uparrow}_{\leq}(B) \\ \dfrac{\Gamma' \vdash_{\leq} A^- \quad \Delta'_1, B', \Delta'_2 \vdash_{\leq} Z^-}{\Delta'_1, B' / A^-, \Gamma, \Delta'_2 \vdash_{\leq} Z^-}\, /L \\ \text{where } B' / A^- \in Dupl^{\uparrow}_{\leq}(B / A) \end{array}$$

- the other cases follow similarly the rule and structure without difficulty.
We now show (a) using (b), we suppose Z is primitive :
- the axiom cases are similar to (b)
- a right rule is not possible
- we detail the $/L$ case :

$$\frac{\Gamma \vdash_{\leq} A \quad \Delta_1, B, \Delta_2 \vdash_{\leq} Z}{\Delta_1, B / A, \Gamma, \Delta_2 \vdash_{\leq} Z}\, /L \quad \begin{array}{l} \text{by rec. (b)} \exists \Gamma' \in Dupl^{\uparrow}_{\leq}(\Gamma), \exists A^- \in Dupl^{\downarrow}_{\leq}(A) \\ \text{by rec. (a)} \exists \Delta'_i \in Dupl^{\uparrow}_{\leq}(\Delta_i), \exists B' \in Dupl^{\uparrow}_{\leq}(B) \\ \dfrac{\Gamma' \vdash_{\leq} A^- \quad \Delta'_1, B', \Delta'_2 \vdash_{\leq} Z}{\Delta'_1, B' / A^-, \Gamma, \Delta'_2 \vdash_{\leq} Z}\, /L \\ \text{where } B' / A^- \in Dupl^{\uparrow}_{\leq}(B / A) \end{array}$$

- the other cases follow similarly the rule and structure without difficulty.

■

Proof of Fact (2) by joined induction for (abc) on the derivation. We consider (for (bc)) a version of the calculus where axioms are on primitives such that for a deduction of $\Gamma \vdash Y \;/\; Z$, there is a deduction of not greater length for $\Gamma, Z \vdash Y$.
Part (2)(a) : we first consider $h(X)\;\alpha \vdash_{\leq'} \alpha'$ and the last rule in a derivation :

- if this is an axiom, then X is empty
- if $/L$ is the last rule, there are two cases (with subcases)
  - if the rule introduces $\boxed{h(B) \;/\; h(A)}$, s. t. X is $\Delta,\ B \;/\; A,\ \Gamma,\ \Delta'$

$$\frac{h(\Gamma) \vdash_{\leq'} h(A) \qquad \boxed{h(\Delta),\ h(B),\ h(\Delta'),\ \alpha \vdash_{\leq'} \alpha'}}{h(\Delta),\ h(B) \;/\; h(A),\ h(\Gamma),\ h(\Delta'),\ \alpha \vdash_{\leq'} \alpha'} \quad \boxed{\text{by rec. (a), } h(B) \text{ being not empty}}$$

or

$$\frac{h(\Gamma, \Delta'),\ \alpha \vdash_{\leq'} h(A) \qquad \boxed{h(\Delta),\ h(B) \vdash_{\leq'} \alpha'}}{h(\Delta),\ h(B) \;/\; h(A),\ h(\Gamma),\ h(\Delta'),\ \alpha \vdash_{\leq'} \alpha'} \quad \text{impossible, see } \boxed{\text{Fact(1)}}$$

  - if the rule introduces $\boxed{h(p_i) = q_0 \;/\; exp(q_1, \Pi'')}$, in $h(X)$ s. t. X is $\Delta,\ p_i,\ \Gamma,\ \Delta'$

$$\frac{\boxed{h(\Gamma) \vdash_{\leq'} exp(q_1, \Pi'')} \qquad h(\Delta),\ q_0,\ h(\Delta'),\ \alpha \vdash_{\leq'} \alpha'}{h(\Delta),\ q_0 \;/\; exp(q_1, \Pi''),\ h(\Gamma),\ h(\Delta'),\ \alpha \vdash_{\leq'} \alpha'} \quad \text{impossible, see } \boxed{\text{Fact (1)}}$$

or

$$\frac{h(\Gamma,\ \Delta'),\ \alpha \vdash_{\leq'} exp(q_1, \Pi'') \qquad \boxed{h(\Delta), q_0 \vdash_{\leq'} \alpha'}}{h(\Delta), q_0 \;/\; exp(q_1, \Pi''), h(\Gamma\ \Delta'), \alpha \vdash_{\leq'} \alpha'} \quad \text{we get : } \alpha \in \{q_0, q_1\} \text{ by } \boxed{\text{Fact (1)}}$$

but then $\boxed{\text{by rec. (2)a}}$ Δ is empty and $\alpha' = q_0$

also $\boxed{\text{by (b) and rec. (2)a}}$ Γ, Δ' is empty and $\alpha = q_1$, thus (a).

- if $\backslash\, L$ is the last rule, it introduces $\boxed{h(A) \backslash h(B)}$, similar to the first subcase for $/\, L$ above

$$\frac{h(\Gamma) \vdash_{\leq'} h(A) \qquad \boxed{h(\Delta),\ h(B),\ h(\Delta'),\ \alpha \vdash_{\leq'} \alpha'}}{h(\Delta),\ h(\Gamma),\ h(A) \backslash h(B),\ h(\Delta')\ \alpha \vdash_{\leq'} \alpha'} \quad \boxed{\text{by rec. (a), } h(B) \text{ being not empty}}$$

- if $\bullet L$ is the last rule, it introduces $\boxed{h(A) \bullet h(B)}$, we apply rec. (a) to the antecedent

Part (2)(bc) : we then consider (b) $h(X), \alpha \models_{\leq'} exp(q_1, \Pi)$, suppose $\Pi = \beta.\Pi$" and its equivalent form (c) if $h(X), \alpha, exp(q_1, \Pi") \setminus \beta \models_{\leq'} \beta$ (where Π" may be ϵ) then X is empty and $\alpha = q_1$. We discuss the last rule in a derivation for (c).

- if this is an axiom, this is impossible. The right rules are also not possible for (c).
- if $/L$ is the last rule, there are two cases (with subcases)
  - if the rule introduces $\boxed{h(B) / h(A)}$, s. t. X is $\Delta,\ B / A,\ \Gamma,\ \Delta'$

$$\frac{h(\Gamma)\models_{\leq'} h(A) \qquad \boxed{h(\Delta),\ h(B),\ h(\Delta'),\ \alpha,\ exp(q_1,\Pi'') \setminus \beta \models_{\leq'} \beta}}{h(\Delta),\ h(B) \,/\, h(A),\ h(\Gamma),\ h(\Delta'),\ \alpha,\ exp(q_1,\Pi'') \setminus \beta \models_{\leq'} \beta}$$

$\boxed{\text{impossible by rec. (c), h(B) being not empty}}$

or

$$\frac{\boxed{h(\Gamma),\ h(\Delta'),\ \alpha\models_{\leq'} h(A)} \qquad \boxed{h(\Delta),\ h(B),\ exp(q_1,\Pi'') \setminus \beta\models_{\leq'} \beta}}{h(\Delta),\ h(B) \,/\, h(A),\ h(\Gamma),\ h(\Delta'),\ \alpha,\ exp(q_1,\Pi'') \setminus \beta\models_{\leq'} \beta}$$

impossible, see $\boxed{\text{Fact(1)}}$

or

$$\frac{h(\Gamma),\ h(\Delta'),\ \alpha,\ exp(q_1,\Pi'') \setminus \beta\models_{\leq'} h(A) \qquad \boxed{h(\Delta),\ h(B)\models_{\leq'} \beta}}{h(\Delta),\ h(B) \,/\, h(A),\ h(\Gamma),\ h(\Delta'),\ \alpha,\ exp(q_1,\Pi'') \setminus \beta\models_{\leq'} \beta}$$

impossible, see $\boxed{\text{Fact(1)}}$

  - if the rule introduces $\boxed{h(p_i) = q_0 \,/\, exp(q_1, \Pi_i)}$, in $h(X)$, where $X = \Delta, p_i, \Gamma, \Delta'$

$$\frac{\boxed{h(\Gamma)\models_{\leq'} exp(q_1,\Pi_i)} \qquad h(\Delta),\ q_0,\ h(\Delta'),\ \alpha,\ exp(q_1,\Pi'') \setminus \beta\models_{\leq'} \beta}{h(\Delta),\ q_0 \,/\, exp(q_1,\Pi_i),\ h(\Gamma),\ h(\Delta'),\ \alpha,\ exp(q_1,\Pi'') \setminus \beta\models_{\leq'} \beta}$$

impossible, see $\boxed{\text{Fact (1)}}$

or

$$\frac{h(\Gamma,\Delta'),\ \alpha\models_{\leq'} exp(q_1,\Pi_i) \qquad \boxed{h(\Delta),\ q_0, exp(q_1,\Pi'') \setminus \beta\models_{\leq'} \beta}}{h(\Delta), q_0 \,/\, exp(q_1,\Pi_i), h(\Gamma,\Delta'),\alpha, exp(q_1,\Pi'') \setminus \beta\models_{\leq'} \beta}$$

impossible, by rec. (c)

or

$$\frac{h(\Gamma,\Delta'),\ \alpha,\ exp(q_1,\Pi'') \setminus \beta\models_{\leq'} exp(q_1,\Pi_i) \qquad \boxed{h(\Delta),\ q_0\models_{\leq'} \beta}}{h(\Delta), q_0 \,/\, exp(q_1,\Pi_i), h(\Gamma,\Delta'),\alpha, exp(q_1,\Pi'') \setminus \beta\models_{\leq'} \beta}$$

impossible, see $\boxed{\text{Fact (1)}}$

- if $\setminus L$ is the last rule,
  - if it introduces $\boxed{h(A) \setminus h(B)}$ in $h(X)$: similar to the first subcase for $/L$

$$\frac{h(\Gamma)\models_{\leq'} h(A) \qquad \boxed{h(\Delta),\ h(B),\ h(\Delta'),\ \alpha,\ exp(q_1,\Pi'') \setminus \beta\models_{\leq'} \beta}}{h(\Delta),\ h(\Gamma),\ h(A) \setminus h(B),\ h(\Delta'),\ \alpha,\ exp(q_1,\Pi'') \setminus \beta\models_{\leq'} \beta}$$

$\boxed{\text{by rec. (c)}}$, $h(B)$ being not empty

  - if the rule introduces $\boxed{exp(q_1, \Pi'') \setminus \beta}$ with $X = \Delta, \Gamma$

$$\frac{\boxed{h(\Gamma),\ \alpha\vdash' exp(q_1,\Pi'')} \qquad h(\Delta),\ \beta\models_{\leq'} \beta}{h(\Delta),\ h(\Gamma),\ \alpha,\ exp(q_1,\Pi'') \setminus \beta\models_{\leq'} \beta}$$

$\boxed{\text{by rec.(2bc)}}$ Γ is empty, and $\alpha = q_1$

$\boxed{\text{by rec (2a)}}$, Δ is also empty

- if $\bullet L$ is the last rule, it introduces $\boxed{h(A)\bullet h(B)}$, we apply rec. (c) to the antecedent, the case is impossible

■