Pregroup Calculus as a Logic Functor

Annie Foret
foret@irisa.fr
http://www.irisa.fr/prive/foret

IRISA – University Rennes1 , FRANCE

PLAN

- Background
  - Categorial grammars
  - Logic functors
- Pregroups : properties, tools, applications
  - Pregroup grammars
  - Formal Models, Linguistic examples
- Pregroup calculus as a logic component
  - a first attempt towards a Logic functor
  - our proposal : FPG
- Main properties of FPG
  - Lemmas overview
  - Cut elimination, composed calculi
- Conclusion and remarks

Categorial grammars

- Σ = alphabet for words of a natural language

  { John, runs, swims, fast,. . . }
- Pr = primitive types : ($S, N, SN, SV, \cdots$)

  Types = ex: $Tp ::= Pr \mid Tp \backslash Tp \mid Tp / Tp$.

- with derivation rules on types Logical part

AB $\backslash_e : A \backslash B, B \vdash A$ and $\backslash_e : B, B \backslash A \vdash A$

Categorial grammars

- Σ = alphabet for words of a natural language
  { John, runs, swims, fast,. . . }
- Pr = primitive types : ($S, N, SN, SV, \cdots$)
  Types = ex: $Tp ::= Pr \mid Tp\backslash Tp \mid Tp/Tp$.
- A *categorial grammar* on Σ

- associate types of Tp to words in Σ Lexicon part

  { John $\mapsto N$, runs, swims $\mapsto SN \setminus S, \ldots$}

- with derivation rules on types Logical part

  AB $\backslash_e : A \setminus B, B \vdash A$ and $\backslash_e : B, B \setminus A \vdash A$

Categorial grammars

- Σ = alphabet for words of a natural language

  { John, runs, swims, fast,... }
- Pr = primitive types : ($S, N, SN, SV, \cdots$)

  Types = ex: $Tp ::= Pr \mid Tp \backslash Tp \mid Tp/Tp$.
- A *categorial grammar* on Σ

  - associate types of Tp to words in Σ Lexicon part

  { John $\mapsto N$, runs, swims $\mapsto SN \backslash S, \ldots$ }

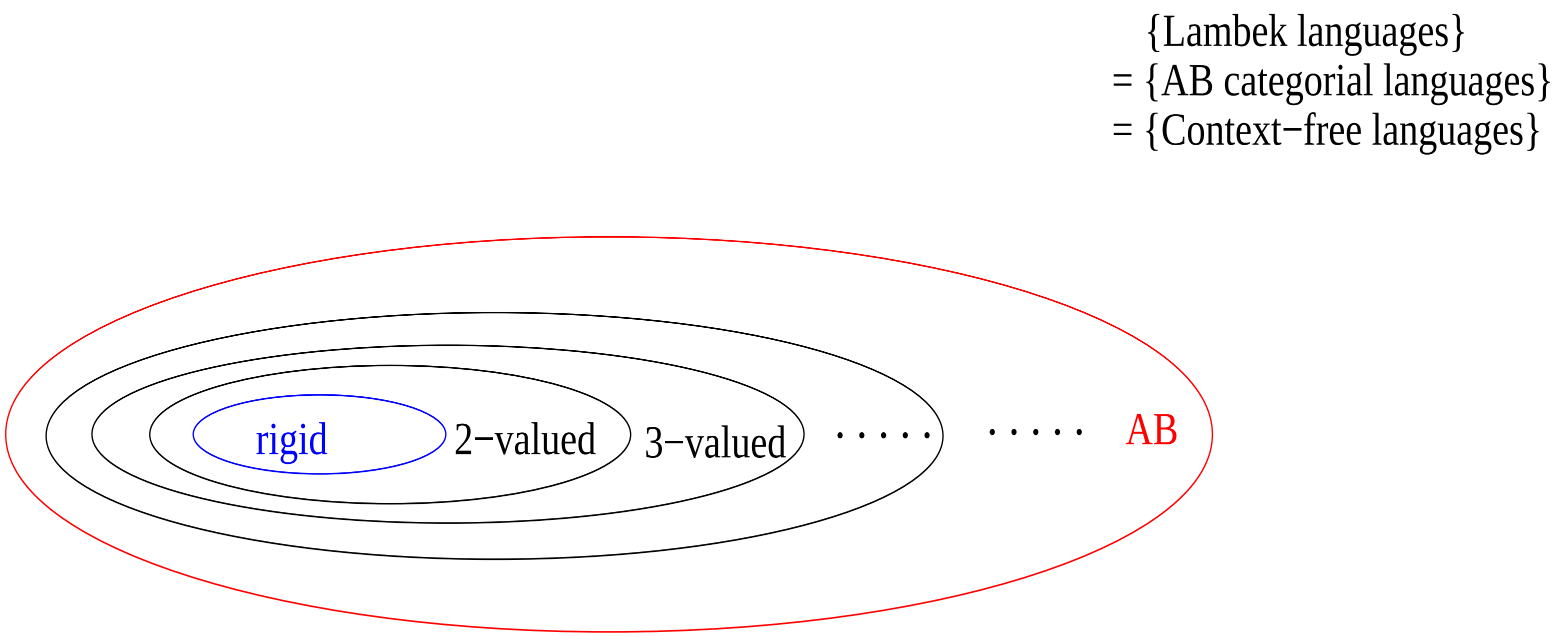
  - with derivation rules on types Logical part

  AB $\backslash_e : A \backslash B, B \vdash A$ and $\backslash_e : B, B \backslash A \vdash A$
- G *generates* a string $c_1 \ldots c_n \in \Sigma^+$ iff

  $\exists A_1, \ldots, A_n \in Tp$:

  $G : c_i \mapsto A_i \; (1 \leq i \leq n)$ and $A_1, \ldots, A_n \vdash S$
- $\mathcal{L}(G)$ = set of strings generated by G (language w.r.t. $\vdash$)

Hierarchy of k-valued categorial grammars



Def: k-valued means at most k types per word (rigid is k=1)

Fact: Class of rigid (k-valued) AB languages learnable "in the limit" (Gold)
In contrast to rigid Lambek or Pregroups

On Logic Functors

- In [Ferré, Ridoux] a *logic* $\mathcal{A}$ is viewed as the association of
  - an *abstract syntax* $AS_{\mathcal{A}}$,
  - a *semantics* $S_{\mathcal{A}}$,
  - operations $P_{\mathcal{A}}$ (and *their implementation*) ,

  including a *subsumption (or entailment) relation*, $\leq_{\mathcal{A}}$.
  - and a *type* $T_{\mathcal{A}}$ made of a set of properties,
- The class of these logics is denoted by $\mathbb{L}$.
- A *logic functor* F takes logics $L_1, \ldots, L_n$ of $\mathbb{L}$ as parameters and returns a logic $F(L_1, \ldots, L_n)$ in $\mathbb{L}$;

  viewed as a tuple (AS_F, S_F, P_F, T_F) of functions s. t.

$$AS_{F(L_1,\ldots,L_n)} = AS_F(AS_{L_1}, \ldots, AS_{L_n}) \qquad P_{F(L_1,\ldots,L_n)} = P_F(P_{L_1}, \ldots, P_{L_n})$$

similarly for S_F, T_F

On Logic Functors

– http://www.irisa.fr/LIS/softwares/ –

- LogFun ToolBox : implemented in Objective Caml
  http://www.irisa.fr/LIS/ferre/logfun/ (see) doc/report/)
- Logical Components, and “Glue”
  - Prop, ...
  - Concrete domains (Atom, Int, Interval,...);
    Structured Data (Product,...)
- Provers (decidable fragments)
- Customized logics
  Prop(Atom), Prop(Interval(Int)), ...
- Querying, Navigating in logical contexts
  http://www.irisa.fr/LIS/ferre/camelis/

PLAN

- Background
  - Categorial grammars
  - Logic functors
- Pregroups : properties, tools, applications
  - Pregroup grammars
  - Formal Models, Linguistic examples
- Pregroup calculus as a logic component
  - a first attempt towards a Logic functor
  - our proposal : FPG
- Main properties of FPG
  - Lemmas overview
  - Cut elimination, composed calculi
- Conclusion and remarks

Pregroup : definitions

A *pregroup* is a structure $(P, \leq, \cdot, l, r, 1)$ s. t. $(P, \leq, \cdot, 1)$ is a partially ordered monoid in which l, r are unary operations on P that satisfy:

$$(PRE) \quad a^l.a \leq 1 \leq a.a^l \quad \text{and} \quad a.a^r \leq 1 \leq a^r.a$$

or equivalently: $a.b \leq c \Leftrightarrow a \leq c.b^l \Leftrightarrow b \leq a^r.c$

Some equations follow from the def. $a^{rl} = 1 = a^{lr}$

we also have: $(a.b)^r = b^r.a^r \ , \ (a.b)^l = b^l.a^l \ , \ 1^r = 1 = 1^l$

but not, in general: $a^{rr} \neq a \neq a^{ll}$

iterated adjoints: $\ldots a^{(-2)} = a^{ll}, a^{(-1)} = a^l, a^{(0)} = a, a^{(1)} = a^r, a^{(2)} = a^{rr} \ldots$

A *monoid* is a structure $< M, \cdot, 1 >$, such that $\cdot$ is associative and has a neutral element 1

A partially ordered monoid is a monoid $(M, \cdot, 1)$ with a partial order $\leq$ that satisfies $\forall a, b, c$: $a \leq b \Rightarrow c \cdot a \leq c \cdot b$ and $a \cdot c \leq b \cdot c$.

Free pregroups

- the set of *atomic types* is : $P^{(Z)} = \{p^{(i)} \mid p \in P, i \in Z\}$
- the *set of types* is
  $Cat_{(P,\leq)} = \{p_1^{(i_1)} \cdots p_n^{(i_n)} \mid p_k \in P, i_k \in Z$ for $0 \leq k \leq n\}$
- $\leq$ on $Cat_{(P,\leq)}$ is the smallest reflexive and transitive relation, s.t. for all $p, q \in Pr$, $X, Y \in Cat_{(P,\leq)}$ and $n \in Z$:

$$Xp^{(n)}p^{(n+1)}Y \leq XY \qquad \text{(contraction),}$$
$$XY \leq Xp^{(n+1)}p^{(n)}Y \qquad \text{(expansion),}$$
$$Xp^{(n)}Y \leq Xq^{(n)}Y, \qquad \text{(induction)}$$

  if $p \leq q$ with n even or $q \leq p$ with n odd

- the free pregroup generated by $(P, \leq)$ is defined on classes $[...]$ modulo $\sim$ s.t. $X \sim Y$ iff $X \leq Y$ *and* $Y \leq X$

Deductions in Free Pregroups

Deduction system (Buszkowski), $\mathcal{S}^{Adj}$

For $X, Y \in Cat_{(P,\leq)}$, we have: $X \leq Y$ iff it is deducible in:

$$X \leq X \quad (Id) \qquad \frac{XY \leq Z}{Xp^{(n)}p^{(n+1)}Y \leq Z}\,(A_L) \qquad \frac{Xq^{(k)}Y \leq Z}{Xp^{(k)}Y \leq Z}\,(IND_L)$$

$$\frac{X \leq Y \quad Y \leq Z}{X \leq Z}\,(Cut) \qquad \frac{X \leq YZ}{X \leq Yp^{(n+1)}p^{(n)}Z}\,(A_R) \qquad \frac{X \leq Yp^{(k)}Z}{X \leq Yq^{(k)}Z}\,(IND_R)$$

with $q \leq p$ if k is even

or $p \leq q$ if k is odd

Cut Elimination

Every derivable inequality has a *Cut-free derivation*

---

symbol corrigendum: in the proceedings permute symbols p and q for IND_R of $\mathcal{S}^{Adj}$.

Free Pregroup Interpretation

- FP = free pregroup on $(Pr, =)$
- Interpretation $[\![\cdot]\!]$ from formulas in L or NL, to FP
  $[\![A]\!] = A$ if A is a primitive type of Pr
  $[\![C_1 \setminus C_2]\!] = [\![C_1]\!]^r [\![C_2]\!]$
  $[\![C_1 \, / \, C_2]\!] = [\![C_1]\!] [\![C_2]\!]^l$
  $[\![C_1 \bullet C_2]\!] = [\![C_1]\!] [\![C_2]\!]$
  The notation extends to sequents by:
  $[\![A_1, \ldots, A_n]\!] = [\![A_1]\!] \cdots [\![A_n]\!]$
- Property FP is a model for L (hence for NL):
  if $\Gamma \vdash_L C$ then $[\![\Gamma]\!] \leq_{FP} [\![C]\!]$
- The converse does not hold:
  $(a.b) \, / \, c \not\vdash a.(b \, / \, c)$ $[\![(a.b) \, / \, c]\!] = [\![a.(b \, / \, c)]\!] = a.b.c^l$
  $(p \, / \, ((p \, / \, p) \, / \, p)) \, / \, p \not\vdash p$ $\underline{p\,p^{ll}\,\underline{p^{ll}\,p^{l}}\,p^{l}} \leq p$

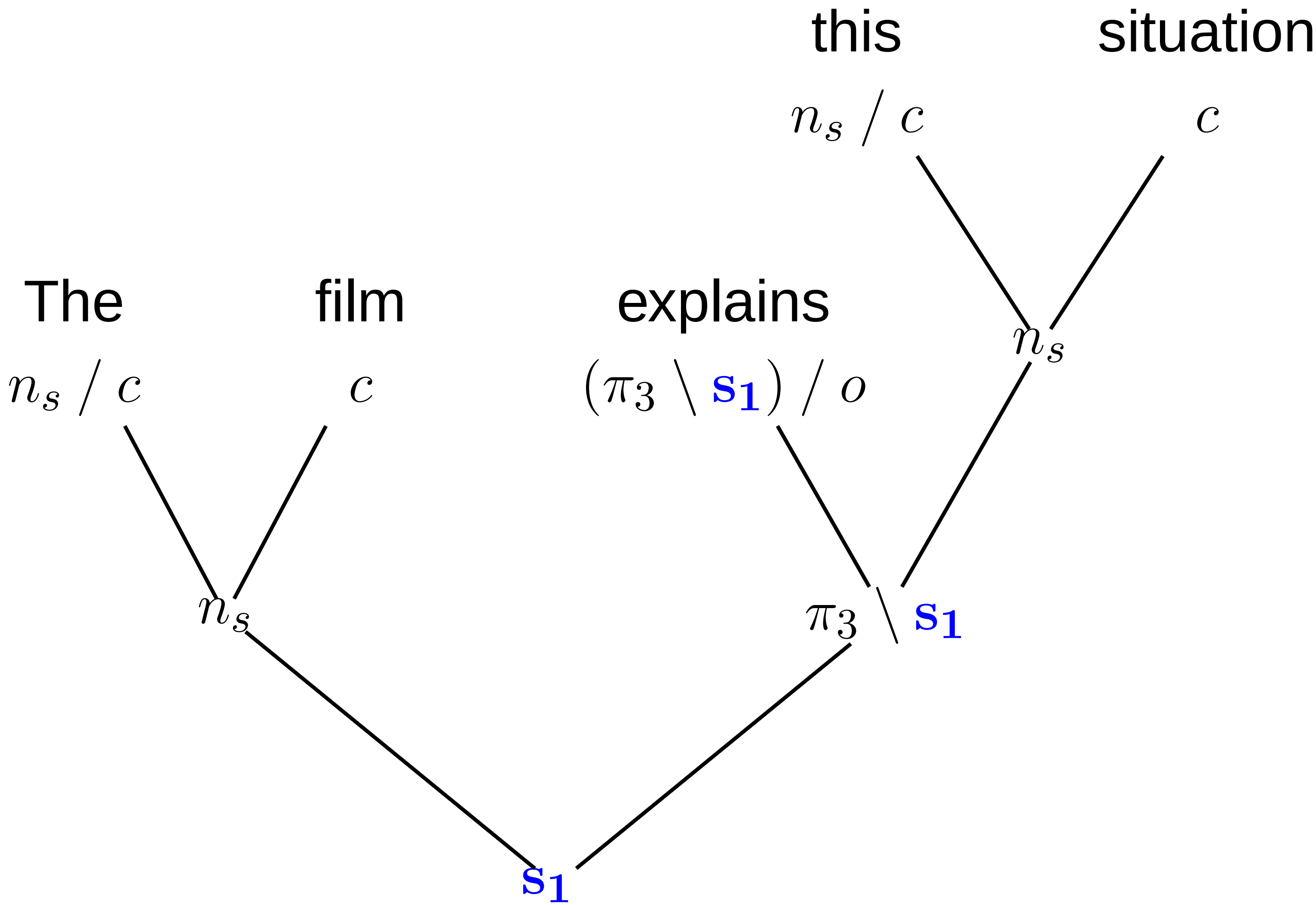
A linguistic example : PG

sentence:	The	film	explains	this	situation
types:	$(n_s\ c^l\ )$	c	$(\pi_3{}^r\ \mathbf{s_1}\ o^l)$	$(n_s\ c^l)$	c

using primitive types and order postulates as follows:

	c = count noun	(film, situation
$n_s \leq \pi_3$	n_s = singular noun phrase	(John)
	π_k = k^{th} personal subject pronoun	(π_3=he/she/it)
$n_s \leq o$	o = direct object	
$s_1 \leq s \leq \overline{s}$	s_1 = statement in present tense	
	s = declarative sentence	(no tense)
	$\overline{s}$=indirect sentence	

A linguistic example : Lambek-like



where types n_s, π_3, o ... are to be replaced with complex types such that:

$n_s \vdash \pi_3$, $n_s \vdash o$, and $s_1 \vdash s \vdash \overline{s}$

PLAN

- Background
  - Categorial grammars
  - Logic functors
- Pregroups : properties, tools, applications
  - Pregroup grammars
  - Formal Models, Linguistic examples
- Pregroup calculus as a logic component
  - a first attempt towards a Logic functor
  - our proposal : FPG
- Main properties of FPG
  - Lemmas overview
  - Cut elimination, composed calculi
- Conclusion and remarks

Functor $FPG(\mathcal{A})$, a first proposal: $\mathcal{S}^{Adj}{}_{[\mathcal{A}]}$

- where p, q are formulas in the logic $\mathcal{A}$ (parameter) , $n, k \in Z$ and $X, Y, Z \in Cat_{[\mathcal{A}]}$ -

$$X \leq X \quad (Id) \qquad \frac{XY \leq Z}{Xp^{(n)}p^{(n+1)}Y \leq Z}\,(A_L) \qquad \frac{Xp^{(k)}Y \leq Z}{Xq^{(k)}Y \leq Z}\,(IND_{L+})$$

$$\frac{X \leq Y \quad Y \leq Z}{X \leq Z}\,(Cut) \qquad \frac{X \leq YZ}{X \leq Yp^{(n+1)}p^{(n)}Z}\,(A_R) \qquad \frac{X \leq Yq^{(k)}Z}{X \leq Yp^{(k)}Z}\,(IND_{R+})$$

with $q \leq_A p$ if k is even

or $p \leq_A q$ if k is odd

for rules $(IND_{L+}), (IND_{R+})$

($\mathcal{A}$ *has* $\leq_A$ *as subsumption)*

This is direct adaptation of $\mathcal{S}^{Adj}$.

However some drawbacks of IND_{L+} and IND_{R+}:

- IND_{L+} , IND_{R+} do not have the *subformula property*
- for the given q in the conclusion, $\{p \in AS_{\mathcal{A}} \mid q \leq_A p\}$ is potentially infinite (in constrast to PG-grammars based on *finite* posets).

---

symbol corrigendum: in the proceedings permute p and q for IND_R^+ of $\mathcal{S}^{Adj}{}_{[\mathcal{A}]}$.

Functor $FPG(\mathcal{A})$, snd proposal: $\mathcal{S}_{[\mathcal{A}]}, \mathcal{S}'_{[\mathcal{A}]}$

- where a, b are formulas of $\mathcal{A}$, $n, k \in Z$ and $X, Y, Z \in Cat_{[\mathcal{A}]}$ -

$$\frac{a \leq_A b, \text{ if } m \text{ is even}}{a^{(m)} \leq b^{(m)}} (Sub) \qquad \frac{XY \leq Z \quad a \leq_A b, \text{ if } m \text{ is even}}{Xa^{(m)}b^{(m+1)}Y \leq Z} (A_{L+})$$

$$\frac{b \leq_A a, \text{ if } m \text{ is odd}}{a^{(m)} \leq b^{(m)}} (Sub) \qquad \frac{XY \leq Z \quad b \leq_A a, \text{ if } m \text{ is odd}}{Xa^{(m)}b^{(m+1)}Y \leq Z} (A_{L+})$$

$$X \leq X \ \ (Id) \qquad \frac{Xa^{(m+1)} \leq Y}{X \leq Ya^{(m)}} (I_R) \qquad \frac{X \leq Y \quad Y \leq Z}{X \leq Z} (Cut)$$

- $\mathcal{S}_{[\mathcal{A}]}$ denotes the same system as $\mathcal{S}'_{[\mathcal{A}]}$ without the cut rule.
- *Pregroup grammars on* $\mathcal{A}$ and their language are defined as before, but using $\mathcal{S}_{[\mathcal{A}]}$ instead.

Functor $FPG(\mathcal{A})$, snd proposal: $\mathcal{S}_{[\mathcal{A}]}, \mathcal{S}'_{[\mathcal{A}]}$

- where a, b are formulas of $\mathcal{A}$, $n, k \in Z$ and $X, Y, Z \in Cat_{[\mathcal{A}]}$ -

$$\frac{a \leq_A b, \text{ if } m \text{ is even}}{a^{(m)} \leq b^{(m)}} \, (Sub) \qquad \frac{XY \leq Z \quad a \leq_A b, \text{ if } m \text{ is even}}{X a^{(m)} b^{(m+1)} Y \leq Z} \, (A_{L+})$$

$$\frac{b \leq_A a, \text{ if } m \text{ is odd}}{a^{(m)} \leq b^{(m)}} \, (Sub) \qquad \frac{XY \leq Z \quad b \leq_A a, \text{ if } m \text{ is odd}}{X a^{(m)} b^{(m+1)} Y \leq Z} \, (A_{L+})$$

$$X \leq X \;\; (Id) \qquad \frac{X a^{(m+1)} \leq Y}{X \leq Y a^{(m)}} \, (I_R) \qquad \frac{X \leq Y \quad Y \leq Z}{X \leq Z} \, (Cut)$$

- $\mathcal{S}_{[\mathcal{A}]}$ denotes the same system as $\mathcal{S}'_{[\mathcal{A}]}$ without the cut rule.
- condensed presentation for rules (Sub) and (A_{L+}):
  where $a^{(m)} \leq_A b^{(m)}$ stands for $a \leq_A b$ if m even, $b \leq_A a$ if m is odd

$$\frac{a^{(m)} \leq_A b^{(m)}}{a^{(m)} \leq b^{(m)}} \, (Sub) \qquad \frac{XY \leq Z \quad a^{(m)} \leq_A b^{(m)}}{X a^{(m)} b^{(m+1)} Y \leq Z} \, (A_{L+})$$

PLAN

- Background
  - Categorial grammars
  - Logic functors
- Pregroups : properties, tools, applications
  - Pregroup grammars
  - Formal Models, Linguistic examples
- Pregroup calculus as a logic component
  - a first attempt towards a Logic functor
  - our proposal : FPG
- Main properties of FPG
  - Lemmas overview
  - Cut elimination, composed calculi
- Conclusion and remarks

Functor $FPG(\mathcal{A})$, lemmas on $\mathcal{S}_{[\mathcal{A}]}, \mathcal{S}'_{[\mathcal{A}]}$

$$\frac{a \leq_A b, \text{ if } m \text{ is even}}{a^{(m)} \leq b^{(m)}}\,(Sub) \qquad \frac{XY \leq Z \quad a \leq_A b, \text{ if } m \text{ is even}}{Xa^{(m)}b^{(m+1)}Y \leq Z}\,(A_{L+})$$

$\ldots$ *if m is odd*

$$X \leq X \;\; (Id) \qquad \boxed{\frac{Xa^{(m+1)} \leq Y}{X \leq Ya^{(m)}}\,(I_R)} \qquad \frac{X \leq Y \quad Y \leq Z}{X \leq Z}\,(Cut)$$

1. rule I_R is reversible in both $\mathcal{S}_{[\mathcal{A}]}$ (without cut) and $\mathcal{S}'_{[\mathcal{A}]}$
   using $\leq_A$ reflexive

Functor $FPG(\mathcal{A})$, lemmas on $\mathcal{S}_{[\mathcal{A}]}, \mathcal{S}'_{[\mathcal{A}]}$

$$\frac{a \leq_A b, \text{ if } m \text{ is even}}{a^{(m)} \leq b^{(m)}} (Sub) \qquad \frac{XY \leq Z \quad a \leq_A b, \text{ if } m \text{ is even}}{Xa^{(m)}b^{(m+1)}Y \leq Z} (A_{L+})$$

$\ldots$ *if m is odd*

$$X \leq X \ (Id) \qquad \boxed{\frac{Xa^{(m+1)} \leq Y}{X \leq Ya^{(m)}} (I_R)} \qquad \frac{X \leq Y \quad Y \leq Z}{X \leq Z} (Cut)$$

1. rule I_R is reversible in both $\mathcal{S}_{[\mathcal{A}]}$ (without cut) and $\mathcal{S}'_{[\mathcal{A}]}$

   using $\leq_A$ reflexive

. IND_L and IND_{R-} (weak form of (IND_R)) hold in $\mathcal{S}_{[\mathcal{A}]}$ and $\mathcal{S}'_{[\mathcal{A}]}$

$$\frac{Xb^{(k)}Y \leq Z}{\overline{Xa^{(k)}Y \leq Z}} (IND_{L+}) \qquad \begin{array}{c} a \leq_A b \text{ if } m \text{ even} \\ b \leq_A a \text{ if } m \text{ is odd} \end{array} \qquad \frac{X \leq Ya^{(k)}}{\overline{X \leq Yb^{(k)}}} (IND_{R-})$$

Functor $FPG(\mathcal{A})$, lemmas on $\mathcal{S}_{[\mathcal{A}]}, \mathcal{S}'_{[\mathcal{A}]}$

$$\frac{a \leq_A b, \text{ if } m \text{ is even}}{a^{(m)} \leq b^{(m)}}\ (Sub) \qquad \frac{XY \leq Z \quad a \leq_A b, \text{ if } m \text{ is even}}{Xa^{(m)}b^{(m+1)}Y \leq Z}\ (A_{L+})$$

$\ldots$ *if m is odd*

$$X \leq X \quad (Id) \qquad \boxed{\frac{Xa^{(m+1)} \leq Y}{X \leq Ya^{(m)}}\ (I_R)} \qquad \frac{X \leq Y \quad Y \leq Z}{X \leq Z}\ (Cut)$$

1. rule I_R is reversible in both $\mathcal{S}_{[\mathcal{A}]}$ (without cut) and $\mathcal{S}'_{[\mathcal{A}]}$

   using $\leq_A$ reflexive

. IND_L and IND_{R-} (weak form of (IND_R)) hold in $\mathcal{S}_{[\mathcal{A}]}$ and $\mathcal{S}'_{[\mathcal{A}]}$

$$\frac{\dfrac{Xb^{(k)}Y \leq Z}{}}{Xa^{(k)}Y \leq Z}\ (IND_{L+}) \qquad \begin{array}{l} a \leq_A b \text{ if } m \text{ even} \\ b \leq_A a \text{ if } m \text{ is odd} \end{array} \qquad \frac{\dfrac{X \leq Ya^{(k)}}{}}{X \leq Yb^{(k)}}\ (IND_{R-})$$

2. if $Xa^{(m+1)}b^{(m)}Y \leq Z$ and $a^{(m)} \leq_A b^{(m)}$ then $XY \leq Z$

   analogue of $1 \leq a^r a$

Cut elimination

– from Lemma 1, 2, $\leq_A$ –

Theorem $X \leq Y$ in $\mathcal{S}_{[\mathcal{A}]}$ iff $X \leq Y$ in $\mathcal{S}'_{[\mathcal{A}]}$ $\quad (\forall X, Y \in Cat_{[\mathcal{A}]})$

$$\dfrac{\gamma_l \left\{ \begin{array}{c} \vdots \\ \dfrac{}{X \leq Y}\, R_l \end{array} \right. \qquad \gamma_r \left\{ \begin{array}{c} \vdots \\ \dfrac{}{Y \leq Z}\, R_r \end{array} \right.}{X \leq Z}\ \mathbf{Cut}$$

induction on the number of **Cut**
and the length of a derivation
$\mathcal{P}$ in $\mathcal{S}'_{[\mathcal{A}]}$, ending in **Cut**:

induction on Y (lemmas 1,2)

when Y is simple,
for R_l (left) and R_r (right):
(no A_{L+} as R_r,
no I_R as R_l, for Y simple)

R_l	R_r	method
Sub	I_R	lemma 1 $[IND_L]$
A_{L+}	Sub	lemma 1 $[IND_{R-}]$
A_{L+}	I_R	permute R_l with cut
Sub	Sub	transitivity of $\leq_A$

Functor $FPG(\mathcal{A})$, lemmas on $\mathcal{S}_{[\mathcal{A}]}, \mathcal{S}'_{[\mathcal{A}]}$

- **Lemma**
  Rule $[IND_R]$, $[A_R]$ ($[A_{R+}]$) are admissible in $\mathcal{S}'_{[\mathcal{A}]}$:
  - $$\frac{\underline{\underline{X \leq Y a^{(k)} Z}}\ and\ a^{(k)} \leq_A b^{(k)}}{X \leq Y b^{(k)} Z}\ (IND_{R+}) \qquad (a, b \in AS_{\mathcal{A}})$$
  - $$\frac{\underline{X \leq Y Z}}{X \leq Y a^{(n+1)} a^{(n)} Z}\ (A_R) \qquad \frac{\underline{\underline{X \leq Y Z}}\ and\ a^{(n)} \leq_A b^{(n)}}{X \leq Y a^{(n+1)} b^{(n)} Z}\ (A_{R+})$$
- **Theorem**
  - $X \leq Y$ is provable in $\mathcal{S}_{[\mathcal{A}]}$ iff it is provable in $\mathcal{S}^{Adj}{}_{[\mathcal{A}]}$.

Functor $FPG(\mathcal{A})$, lemmas on $\mathcal{S}_{[\mathcal{A}]}, \mathcal{S}'_{[\mathcal{A}]}$

- **Lemma**
  Rule $[IND_R]$, $[A_R]$ ($[A_{R+}]$) are admissible in $\mathcal{S}'_{[\mathcal{A}]}$:
  - $$\frac{\underline{\underline{X \leq Y a^{(k)} Z}}\ and\ a^{(k)} \leq_A b^{(k)}}{X \leq Y b^{(k)} Z} (IND_{R+}) \quad (a, b \in A\mathcal{S}_{\mathcal{A}})$$
  - $$\frac{\underline{\underline{X \leq Y Z}}}{X \leq Y a^{(n+1)} a^{(n)} Z} (A_R) \quad \frac{\underline{\underline{X \leq Y Z}}\ and\ a^{(n)} \leq_A b^{(n)}}{X \leq Y a^{(n+1)} b^{(n)} Z} (A_{R+})$$
- **Theorem**
  - $X \leq Y$ is provable in $\mathcal{S}_{[\mathcal{A}]}$ iff it is provable in $\mathcal{S}^{Adj}{}_{[\mathcal{A}]}$.
  - equivalence with Pregroups (for $\leq_A$ as $\leq$)
    as a particular case

Other Properties of composed calculi

- Proposition. Let $\mathcal{A} = \langle \mathcal{AS}_{\mathcal{A}}, \leq_{\mathcal{A}} \rangle$ where $\leq_A$ is a preorder. If $\leq_A$ is a decidable calculus, then $\mathcal{S}_{[\mathcal{A}]}$ and $\mathcal{S}'_{[\mathcal{A}]}$ (applied to $\mathcal{A}$) are decidable.
  This is clear, using the cut elimination theorem, and the subformula property for $\mathcal{S}_{[\mathcal{A}]}$ (with the special case of I_R, having only one possible antecedent).

Other Properties of composed calculi

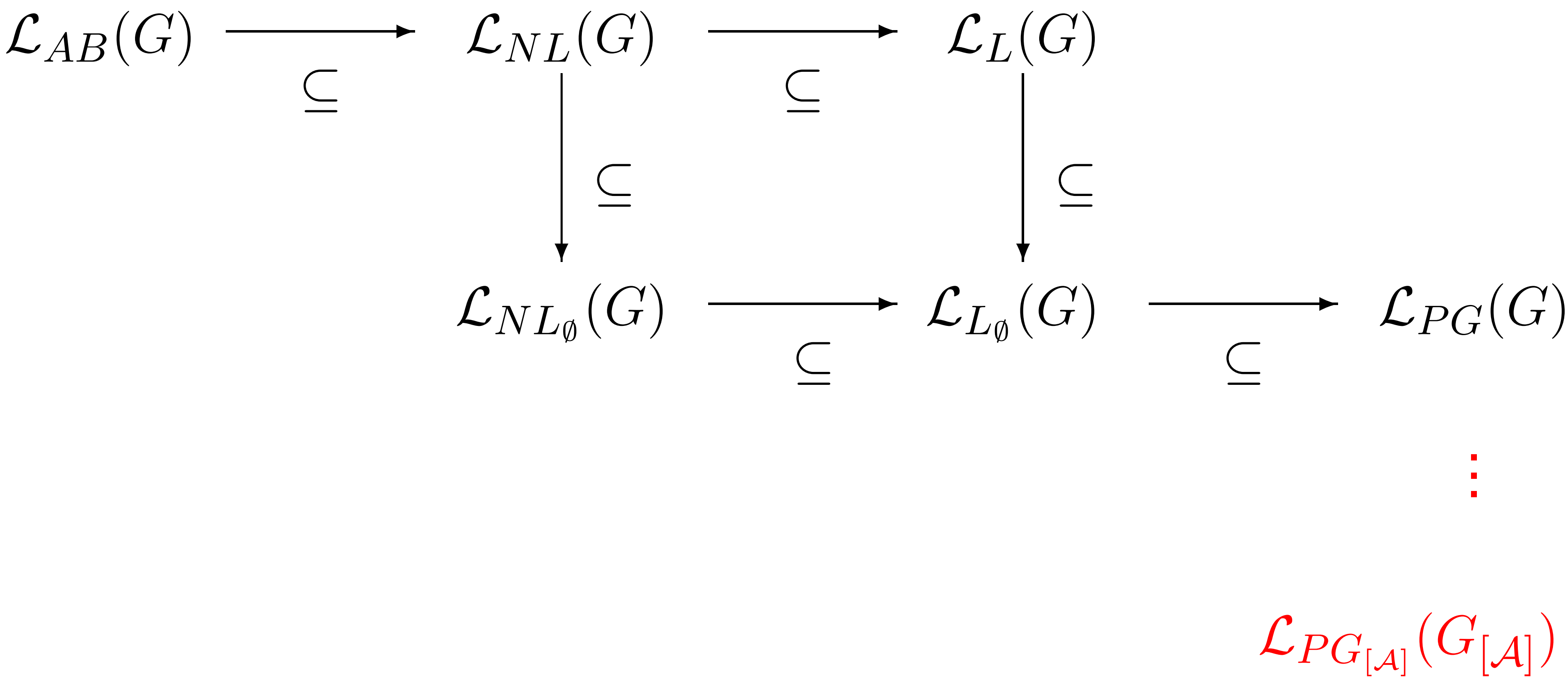
- Proposition. Let $\mathcal{A} = \langle \mathcal{AS}_{\mathcal{A}}, \leq_{\mathcal{A}} \rangle$ where $\leq_A$ is a preorder. If $\leq_A$ is a decidable calculus, then $\mathcal{S}_{[\mathcal{A}]}$ and $\mathcal{S}'_{[\mathcal{A}]}$ (applied to $\mathcal{A}$) are decidable.
  This is clear, using the cut elimination theorem, and the subformula property for $\mathcal{S}_{[\mathcal{A}]}$ (with the special case of I_R, having only one possible antecedent).
- Proposition. The language generated by a PG-grammar G on a Sub-Logic $\mathcal{A} = \langle \mathcal{AS}_{\mathcal{A}}, \leq_{\mathcal{A}} \rangle$ where $\leq_A$ is a preorder (G based on the deduction system $\mathcal{S}_{[\mathcal{A}]}$ or equivalently $\mathcal{S}'_{[\mathcal{A}]}$) is a context-free language.
  This can be shown by associating to G a free PG-grammar G_{PG}, obtained by replacing, in the assignment, all subformulas F that belong to A by a new constant c_F, with $c_F \leq c_{F'}$ whenever $F \leq_A F'$: G_{PG} generates the same language as G.

Conclusion and remarks

We have reformulated and proposed to extend the pregroup calculus, for its composition with other logics and calculi.

- Formal results
  The *cut elimination property* and
  the *decidability* property.
  Equivalence with PG, when $\mathcal{A}$ is reduced to Pr.
- Practical issues of "parameterized pregroups"
  Ready as a decision procedure and a parsing algorithm.
  Customized calculi , ex : structure to the basic types.
- Other perspectives
  PG as argument of another logic functor ;
  other Lambek calculi ;
  enrich types in a clear way, both formally and practically.

A schema


$\mathcal{L}_{AB}(G)$
$\subseteq$
$\mathcal{L}_{NL}(G)$
$\subseteq$
$\mathcal{L}_{L}(G)$
$\subseteq$
$\subseteq$
$\mathcal{L}_{NL_\emptyset}(G)$
$\subseteq$
$\mathcal{L}_{L_\emptyset}(G)$
$\subseteq$
$\mathcal{L}_{PG}(G)$
$\vdots$
$\mathcal{L}_{PG_{[\mathcal{A}]}}(G_{[\mathcal{A}]})$