Fully Lexicalized Pregroup Grammars

Annie Foret

joint work with Denis Béchet

Denis.Bechet@lina.univ-nantes.fr
and
foret@irisa.fr
http://www.irisa.fr/prive/foret

LINA – Nantes University, FRANCE

IRISA – University Rennes1 , FRANCE

PLAN

- Introduction
  - Transformation : hopes
  - Background : PG
- First facts
  - A Preliminary Fact
  - Trials in the restricted case
  - Encodings as morphisms
- Construction
  - *what* is the full construction : illustration
  - *how* does it work ? key properties (soundness is easy, completeness is not)
- Conclusion and some related questions

Introduction

The categories are types of a free pregroup generated by a set of primitive types (Pr) with a partial order on Pr. This partial order is not a lexicalized information, not independent of the language that corresponds to a PG.

$q \leq s$ for a yes-or-no question

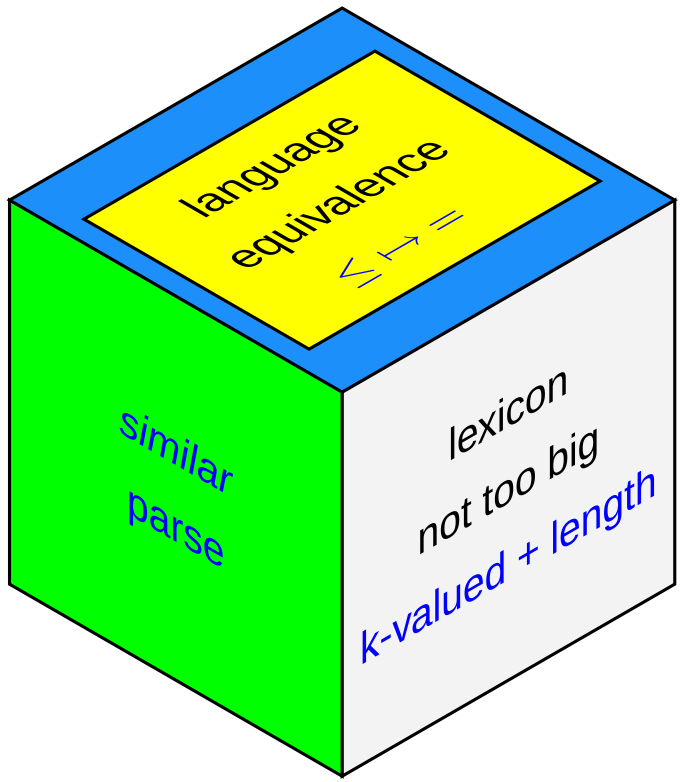
- Question : Is it possible to find a PG that is equivalent to a given PG but where the partial order on primitive types is *universal*? Such that : *(hopes)*
  - the computed PG is not *too big*

    polynomial transformation
  - it works in a *similar* way as the source PG

    a homomorphism,...
  - the same number of types to a word

    k-valued.

Transformation



Pregroup : definitions

A *pregroup* is a structure $(P, \leq, \cdot, l, r, 1)$ s. t. $(P, \leq, \cdot, 1)$ is a partially ordered monoid [a] 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] 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 $a \cdot c \leq b \cdot c$.

Background

– Pregroup–def –

[Free pregroup] Let $(P, \leq)$ be an ordered set of primitive types,

$T_{(P,\leq)} = \{p_1^{(i_1)} \cdots p_n^{(i_n)} \mid 0 \leq k \leq n, p_k \in P$ and $i_k \in \mathbb{Z}\}$ is the set of types.

the empty sequence is denoted by 1.

For X and $Y \in T_{(P,\leq)}$, $X \leq Y$ iff this relation is deductible in the following system

where $p, q \in P$, $n, k \in \mathbb{Z}$ and $X, Y, Z \in T_{(P,\leq)}$:

$$X \leq X \quad (Id) \qquad \frac{X \leq Y \quad Y \leq Z}{X \leq Z}\,(Cut)$$

$$\frac{XY \leq Z}{Xp^{(n)}p^{(n+1)}Y \leq Z}\,(A_L) \qquad \frac{X \leq YZ}{X \leq Yp^{(n+1)}p^{(n)}Z}\,(A_R)$$

$$\frac{Xp^{(k)}Y \leq Z}{Xq^{(k)}Y \leq Z}\,(IND_L) \qquad \frac{X \leq Yq^{(k)}Z}{X \leq Yp^{(k)}Z}\,(IND_R)$$

$q \leq p$ if k is even, and $p \leq q$ if k is odd

Background

– Pregroup grammar –def –

Let $(P, \leq)$ be a finite partially ordered set.

- A *pregroup grammar* based on $(P, \leq)$ is a lexicalized [a] grammar $G = (\Sigma, I, s)$ such that
  - $s \in T_{(P,\leq)}$; ▷
  - G assigns a type X to a string $v_1, \ldots, v_n$ of Σ^* iff for $1 \leq i \leq n$, $\exists X_i \in I(v_i)$ such that $X_1 \cdots X_n \leq X$ in the free pregroup $T_{(P,\leq)}$.
- *The language* $\mathcal{L}(G)$ is the set of strings in Σ^* that are assigned s by G.

[a] a lexicalized grammar is a triple (Σ, I, s): Σ is a finite alphabet, I assigns a finite set of categories (or types) to each $c \in \Sigma$, s is a category (or type) associated to correct sentences.

Background

– Pregroup–example –

Our example is taken from Lambek, with the primitive types:

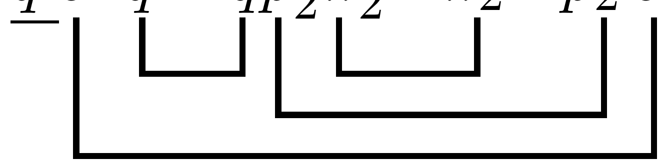
π_2 = second person,
p_2 = past participle,
o = object,
q = yes-or-no question,
q' = question

$q \leq q'$

This sentence gets type q' ($q' \leq s$):

whom have you seen

$\underline{q'} o^{ll} q^l \quad q p_2^l \pi_2^l \quad \pi_2 \quad p_2 \, o^l$



Background

Our example is taken from Lambek, with the primitive types:

π_2 = second person,
p_2 = past participle,
o = object,
q = yes-or-no question,
q' = question

$q \leq q'$

This sentence gets type q' ($q' \leq s$):

whom have you seen

$$\underline{q'} o^{ll} q^l \quad q p_2^l \pi_2^l \quad \pi_2 \quad p_2 \, o^l$$

Background

Our example is taken from Lambek, with the primitive types:

π_2 = second person,
p_2 = past participle,
o = object,
q = yes-or-no question,
q' = question

$$q \leq q'$$

This sentence gets type q' ($q' \leq s$):

whom have you seen

$$\underline{q'} o^{ll} q^l \quad q p_2^l \pi_2^l \quad \pi_2 \quad p_2 \, o^l$$

Background

– Pregroup–example –

Our example is taken from Lambek, with the primitive types:

π_2 = second person,
p_2 = past participle,
o = object,
q = yes-or-no question,
q' = question $\qquad q \leq q'$

This sentence gets type q' ($q' \leq s$):

whom have you seen

$\underline{q'} o^{ll} q^l \quad q p_2^l \pi_2^l \quad \pi_2 \quad p_2 \, o^l$

PLAN

- Introduction
  - Transformation : hopes
  - Background : PG
- First facts
  - A Preliminary Fact
  - Trials in the restricted case
  - Encodings as morphisms
- Construction
  - *what* is the full construction : illustration
  - *how* does it work ? key properties (soundness is easy, completeness is not)
- Conclusion and some related questions

A Preliminary Fact

– from Buszkowski 01 –

Proposition. [PG equivalence and generative capacity]
For any PG G on $(P, \leq)$,
we can construct a PG G' based on $(P', =)$,
s.t. G and G' have the same language.

A Preliminary Fact

– from Buszkowski 01 –

Proposition. [PG equivalence and generative capacity]
For any PG G on $(P, \leq)$,
we can construct a PG G' based on $(P', =)$,
s.t. G and G' have the same language.

Obvious :

- Pregroups Lang. = Context-free Lang. = order 1 AB-Categorial ▷
- Or duplicate the lexical types for each occurrence involved in some an ordering $p_i \leq p_j$
- However this does not preserve the size of the lexicon in general

---

$order(A/B) = order(B \backslash A) = max(order(A, 1 + order(B))), 0$ on Pr

if $X = X' p_0^{2n} Y \mapsto \{X' p_j^{2n} Y' \mid p_j \leq p_0\}$;

if $X = X' p_0^{2n+1} Y' \mapsto \{X' p_j^{2n+1} Y' \mid p_j \geq p_0\}$

Trials in a restricted case

Let $(P, \leq)$ be reduced to $\boxed{p_0 \leq p_1}$, with $\leq'$ as =, (and $q = p_1$).

- Consider the homomorphism from $(P, \leq)$ to $(P', \leq')$ h_γ:

$$
\boxed{
\begin{array}{l|l}
h_\gamma(X^{(n)}) = h_\gamma(X)^{(n)} & h_\gamma(p_0) = q\ \gamma\ \gamma^{(1)} \\
h_\gamma(X.Y) = h_\gamma(X).h_\gamma(Y) & h_\gamma(p_1) = q \\
h_\gamma(1) = 1 & h_\gamma(p_i) = p_i \text{ if } i \notin \{0, 1\}
\end{array}
}
$$

- $h(p_0) \leq' h(p_1)$, since $\gamma\ \gamma^{(1)} \leq' 1$
- $h(p_1) \not\leq' h(p_0)$, since $\gamma\ \gamma^{(1)} \not\leq' 1$
- but... it is not "order-reflecting" :

$$X = p_0^{(2n-2)} p_0^{(2n-1)} p_1^{(2n)} p_1^{(2n-1)}$$

such that $X \not\leq 1$, whereas $h_\gamma(X) \leq' 1$

Trials in a restricted case

Let $(P, \leq)$ be reduced to $\boxed{p_0 \leq p_1}$, with $\leq'$ as =, (and $q = p_1$).

- Consider the homomorphism from $(P, \leq)$ to $(P', \leq')$ h_γ:

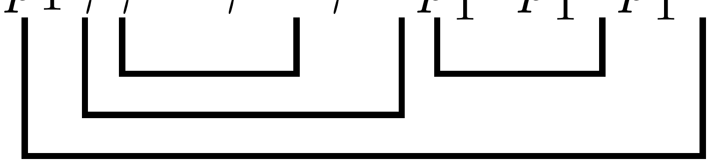
$h_\gamma(X^{(n)}) = h_\gamma(X)^{(n)}$	$h_\gamma(p_0) = q\ \gamma\ \gamma^{(1)}$
$h_\gamma(X.Y) = h_\gamma(X).h_\gamma(Y)$	$h_\gamma(p_1) = q$
$h_\gamma(1) = 1$	$h_\gamma(p_i) = p_i$ if $i \notin \{0, 1\}$

- but... it is not "order-reflecting" :

$$X = p_0^{(2n-2)} p_0^{(2n-1)} p_1^{(2n)} p_1^{(2n-1)}$$

such that $X \not\leq 1$, whereas $h_\gamma(X) \leq' 1$:

$$h_\gamma(X) = p_1 \gamma \gamma^{(1)} \gamma^{(2)} \gamma^{(1)} p_1^{(1)} p_1^{(2)} p_1^{(1)}$$

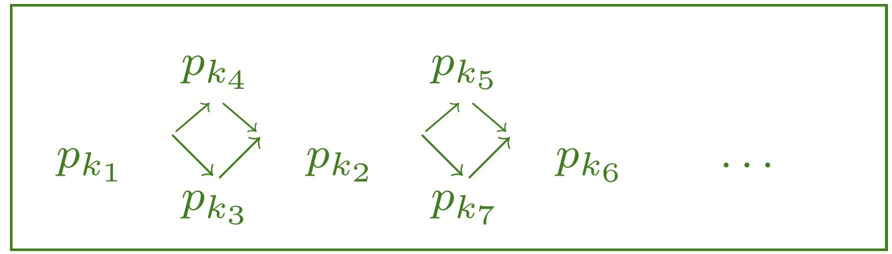


Towards the general case

– Difficulties –

- various partial order schemas

$$p_{i_1} \to p_{i_2} \to p_{i_3} \to p_{i_4} \to \cdots$$
$$p_{j_1} \to p_{j_2} \to p_{j_3} \to p_{j_4} \to \cdots$$



- from group images
  - $[h(X)] = [h(Y)]$ implies $[X] = [Y]$
    thus ensure different equivalence classes (q_R) for different connex components
  - $[X] = [Y]$ implies $[h(X)] = [h(Y)]$
    thus ensure same equivalence class (q) inside a connex component
- involving s, upper and/or lower

Morphism-based encodings

– definitions –

- A mapping h from the free pregroup on $(P, \leq)$ to the free pregroup on $(P', \leq')$, is a *pregroup homomorphism* iff
  1. $\forall X \in T_{(P,\leq)} : h(X^{(n)}) = h(X)^{(n)}$
  2. $\forall X, Y \in T_{(P,\leq)} : h(X.Y) = h(X).h(Y)$
  3. $h(1) = 1$
  4. $\forall X, Y \in T_{(P,\leq)}$: if $X \leq Y$ then $h(X) \leq' h(Y)$ [Monotonicity]
- A mapping from a poset $(P, \leq)$ to a poset $(P', \leq')$ is said *partial-order-preserving* iff
  4b. $\forall p_i, p_j \in P$: if $p_i \leq p_j$ then $h(p_i) \leq' h(p_j)$.

$\Rightarrow$ to ensure, for (4) $\triangleright$

A similar parse

An example revisited

π_2 = second person,
p_2 = past participle,
o = object,
q = yes-or-no question, q' = question
$q \leq q'$

This sentence gets type q' ($q' \leq s$):

whom have you seen

$$\underline{q'}o^{ll}q^{l} \quad qp_2^{l}\pi_2^{l} \quad \pi_2 \quad p_2o^{l}$$

$$\underline{h(q')}h(o)^{ll}h(q)^{l} \quad h(q)h(p_2)^{l}h(\pi_2)^{l} \quad h(\pi_2) \quad h(p_2)h(o)^{l}$$

Simulation :a converse of monotonicity

Def-Prop. [Order-reflecting homomorphism]

Every homomorphism h from the free pregroup on $(P, \leq)$ to the free pregroup on $(P', \leq')$ is such that (1) and (2) are equivalent and define order-reflecting homomorphisms:

(1). $\forall X, Y \in T_{(P,\leq)}$ if $h(X) \leq' h(Y)$ then $X \leq Y$

(2). $\forall X \in T_{(P,\leq)}$ if $h(X) \leq' 1$ then $X \leq 1$

$\Rightarrow$ to ensure (2), rather than (1)

Simulation :a converse of monotonicity

Def-Prop. [Order-reflecting homomorphism]

Every homomorphism h from the free pregroup on $(P, \leq)$ to the free pregroup on $(P', \leq')$ is such that (1) and (2) are equivalent and define order-reflecting homomorphisms:

(1). $\forall X, Y \in T_{(P,\leq)}$ if $h(X) \leq' h(Y)$ then $X \leq Y$

(2). $\forall X \in T_{(P,\leq)}$ if $h(X) \leq' 1$ then $X \leq 1$

Proof. (1) is a corollary of (2) as follows: suppose (2) holds for a pregroup-morphism h with (3) $h(X) \leq' h(Y)$,

we get (4) $h(X\ Y^r) = \underline{h(X)\ h(Y)^r} \leq' h(Y)\ h(Y)^r \leq' \underline{1}$

property (2) then gives (5) $X\ Y^r \leq 1$,

hence $X \leq Y$ (by adding Y on the right of (5)).

$\Rightarrow$ to ensure (2), rather than (1)

PLAN

- Introduction
  - Transformation : hopes
  - Background : PG
- First facts
  - A Preliminary Fact
  - Trials in the restricted case
  - Encodings as morphisms
- Construction
  - *what* is the full construction : illustration
  - *how* does it work ? key properties (soundness is easy, completeness is not)
- Conclusion and some related questions

Construction on one component.

Let $P = Pr \cup Pr'$, Pr a connex component

q and β_k, γ_k, are new letters (no postulate) for each p_k of Pr [a]

We take as poset $P' = Pr' \cup \{q\} \cup \{\beta_k, \gamma_k \mid p_k \in Pr\}$,

$\leq'$ is the restriction of $\leq$ on Pr' ($\leq'$ is identity if Pr' is empty).

Definition. [Simulation-morphism h for Pr]

$h(X^{(n)}) = h(X)^{(n)}$	$h(p_i) = \underbrace{\lambda_i^{(0)} \; q^{(0)} \delta_i^{(0)}}$	$h(1) = 1$
$h(X.Y) = h(X).h(Y)$	for $p_i \in Pr$	$h(p_i) = p_i$ if $p_i \in Pr'$

[a] the symbol q can also be written q_{Pr} if necessary w.r.t. Pr

Construction on one component.

Let $P = Pr \cup Pr'$, Pr a connex component

q and β_k, γ_k, are new letters (no postulate) for each p_k of Pr

Definition. [Simulation-morphism h for Pr]

$h(X^{(n)}) = h(X)^{(n)}$	$h(p_i) = \underbrace{\lambda_i^{(0)} \; q^{(0)} \delta_i^{(0)}}$	$h(1) = 1$
$h(X.Y) = h(X).h(Y)$	for $p_i \in Pr$	$h(p_i) = p_i$ if $p_i \in Pr'$

where

$$\lambda_i = \alpha \left(\prod_{\substack{for\ k'=n..1 \\ if\ p_{k'} \leq p_i}} (\gamma_{k'}^{(1)} \, \gamma_{k'}) \right) \quad \text{and} \quad \delta_i = \left(\prod_{\substack{for\ k=1..n, \\ if\ p_k \geq p_i}} (\beta_k \, \beta_k^{(1)}) \right) \alpha'$$

---

the symbol q can also be written q_{Pr} if necessary w.r.t. Pr

Restricted case : $p_0 \leq p_1 \ldots$

$$h(p_i) = \underbrace{\lambda_i^{(0)} \; q^{(0)} \delta_i^{(0)}} \qquad\qquad \underbrace{\delta_j^{(1)} q^{(1)} \; \lambda_j^{(1)}} = h(p_j)^{(1)}$$

(note the inversion)

- for $\boxed{p_0 \leq p_1}$ where (new symbols):

$$\lambda_0 = \alpha \, (\gamma_0^{(1)} \, \gamma_0) \qquad \delta_0 = \; (\beta_0 \, \beta_0^{(1)}) \, (\beta_1 \, \beta_1^{(1)}) \alpha' \qquad \lambda_0 \leq \lambda_1$$

$$\lambda_1 = \alpha \, (\gamma_1^{(1)} \, \gamma_1) \, (\gamma_0^{(1)} \, \gamma_0) \quad \delta_1 = (\beta_0 \, \beta_0^{(1)}) \, \alpha' \qquad \delta_0 \leq \delta_1$$

we have : $\lambda_0 \leq \lambda_1$, because $(\beta_1 \, \beta_1^{(1)}) \leq 1$

$\delta_0 \leq \delta_1$, because $1 \geq (\gamma_1^{(1)} \, \gamma_1)$

therefore $\quad h(p_0).h(p_1)^{(1)} \leq' 1$

with $\quad h(p_1)h(p_0)^{(1)} \not\leq' 1$

- a subterm per lower atom ($\leq' 1$) in λ, per upper ($\geq' 1$) atom in δ

Restricted case : $p_0 \leq p_1 \ldots$

$$h(p_i) = \underbrace{\lambda_i^{(0)} \; q^{(0)} \delta_i^{(0)}}$$

$$\underbrace{\delta_j^{(1)} q^{(1)} \; \lambda_j^{(1)}} = h(p_j)^{(1)}$$

(note the inversion)

- for $\boxed{p_0 \leq p_1}$ where (new symbols):

$\lambda_0 = \alpha \, (\gamma_0^{(1)} \, \gamma_0)$ $\quad \delta_0 = (\beta_0 \, \beta_0^{(1)}) \, (\beta_1 \, \beta_1^{(1)}) \alpha'$ $\quad \lambda_0 \leq \lambda_1$

$\lambda_1 = \alpha \, (\gamma_1^{(1)} \, \gamma_1) \, (\gamma_0^{(1)} \, \gamma_0)$ $\quad \delta_1 = (\beta_0 \, \beta_0^{(1)}) \, \alpha'$ $\quad \delta_0 \leq \delta_1$

we have : $\lambda_0 \leq \lambda_1$, because $(\beta_1 \, \beta_1^{(1)}) \leq 1$

$\delta_0 \leq \delta_1$, because $1 \geq (\gamma_1^{(1)} \, \gamma_1)$

therefore $\quad h(p_0).h(p_1)^{(1)} \leq' 1$

=

$$\alpha \, (\gamma_1^{(1)} \, \gamma_1) q^{(0)} \, (\beta_1 \, \beta_1^{(1)}) \, (\beta_0 \, \beta_0^{(1)}) \alpha' . \alpha'^{\,(1)} \, (\beta_0^{(1+1)} \, \beta_0^{(1)}) q^{(1)} (\gamma_1^{(1)} \, \gamma_1^{(1+1)}) \, (\gamma_0^{(1)} \, \gamma_0^{(1+1)}) \, \alpha^{(1)} \leq' 1$$

with $\quad h(p_1) h(p_0)^{(1)} \not\leq' 1$

- a subterm per lower atom ($\leq' 1$) in λ, per upper ($\geq' 1$) atom in δ

Restricted case : $p_0 \leq p_1 \ldots$

$$h(p_i) = \underbrace{\lambda_i^{(0)}\ q^{(0)} \delta_i^{(0)}}$$

- for $\boxed{\begin{array}{c} p_0 \leq p_1 \\ p_2 \leq p_1 \end{array}}$ where (new symbols):

$$\lambda_0 = \alpha\,(\gamma_2^{(1)}\,\gamma_2) \qquad \delta_0 = (\beta_0\,\beta_0^{(1)})\,(\beta_1\,\beta_1^{(1)})\alpha'$$

$$\lambda_2 = \alpha\,(\gamma_2^{(1)}\,\gamma_2) \qquad \delta_2 = (\beta_1\,\beta_1^{(1)})\,(\beta_2\,\beta_2^{(1)})\alpha'$$

$$\lambda_1 = \alpha\,(\gamma_2^{(1)}\,\gamma_2)\,(\gamma_1^{(1)}\,\gamma_1)\,(\gamma_0^{(1)}\,\gamma_0) \qquad \delta_1 = (\beta_1\,\beta_1^{(1)})\,\alpha'$$

- a subterm per lower atom ($\leq' 1$) in λ, per upper ($\geq' 1$) atom in δ
- if "reflexive subterm" were dropped : $h(p_0) = h(p_2)$!

The Unrestricted Case on $P = \underbrace{Pr}_{\leq} \cup Pr'$. $-\leq \mapsto \leq' -$

Proposition. [h is order-preserving] (easy by construction)

if $p_i \leq p_j$ then $h(p_i) \leq' h(p_j)$.

It is then a pregroup homomorphism.

Proposition. [Monotonicity of h] (corollary)

$\forall X, Y \in T_{(P,\leq)}$: if $X \leq Y$ then $h(X) \leq' h(Y)$

Proposition. [h is order-reflecting] (hard part)

$\forall X, Y \in T_{(P,\leq)}$: if $h(X) \leq' h(Y)$ then $X \leq Y$

Proposition. [Incrementality]

The construction holds for

a connex component Pr
or several connex components,
possibly whole P

How $h(X) \leq' 1 \mapsto X \leq 1$

– Reasoning with "Left" Derivations –

$h(X) = \dots q^{u_1} \dots q^{u_2} \dots q^{u_3} \dots$

links . . . ?

Derivation $\mathcal{D}$ from $1 \leq' 1$ ending in $h(X) \leq' 1$

Let k denote the step of the leftmost introduction of q, as $q^{(n)} q^{(n+1)}$:

$$\Delta_0 = 1$$

$$\vdots$$

$$\Delta_{k-1} = \Gamma_0 \, \Gamma'_0 \qquad \text{with } \Delta_{k-1} \leq' 1$$

$$\Delta_k = \Gamma_0 \, q^{(n)} \, q^{(n+1)} \, \Gamma'_0 \qquad \text{with } \Delta_k \leq' 1$$

$$\vdots$$

$$\underbrace{\Delta_m}_{=h(X)} = \underbrace{\Gamma_{m-k}}_{\leq' \Gamma_0} \, q^{(n)} \, \underbrace{\Gamma''_{m-k}}_{\leq' 1} \, q^{(n+1)} \, \underbrace{\Gamma'_{m-k}}_{\leq' \Gamma'_0} \qquad \text{with } h(X) = \Delta_m \leq' 1$$

by derivation structure

How $h(X) \leq' 1 \mapsto X \leq 1$

On a kind of parenthesizing in the encodings

Lemma. let

$$X = \underbrace{X_1\, \alpha^{(2u_1)}\, Y_1\, \alpha^{(2u_1+1)}} \ldots \underbrace{X_n\, \alpha^{(2u_n)}\, Y_n\, \alpha^{(2u_n+1)}}\; X_{n+1}$$

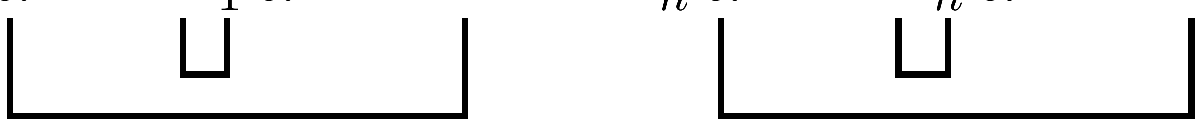
where all X_k, Y_k have no α, and α is not related by $\leq$;

(i) if $X \leq 1$ then :

(1) $\forall k$: $Y_k \leq 1$

(2) $X_1\, X_2 \ldots X_k \ldots X_n\, X_{n+1} \leq 1$

$$X_1\, \alpha^{(2u_1)}\, Y_1\, \alpha^{(2u_1+1)} \ldots X_n\, \alpha^{(2u_n)}\, Y_n\, \alpha^{(2u_n+1)} X_{n+1}$$



variant ▷

How $h(X) \leq' 1 \mapsto X \leq 1$

– Technical Lemmas –

On a kind of parenthesizing in the encodings

Lemma. let

$$X = \underbrace{X_1\, \alpha^{(2u_1)}\, Y_1\, \alpha^{(2u_1+1)}} \ldots \underbrace{X_n\, \alpha^{(2u_n)}\, Y_n\, \alpha^{(2u_n+1)}}\; X_{n+1}$$

where all X_k, Y_k have no α, and α is not related by $\leq$;

(i) if $X \leq 1$ then :

(1) $\forall k$: $Y_k \leq 1$

(2) $X_1\, X_2 \ldots X_k \ldots X_n\, X_{n+1} \leq 1$

(ii) if $Y_0\, \alpha^{(2v+1)}\, X\; \alpha^{(2v+2)} \leq 1$ where Y_0 has no α, then :

(1) $\forall k : Y_k \leq 1$

(2) $X_1\, X_2 \ldots X_k \ldots X_n\, X_{n+1} \leq 1$

variant ▷

Central Lemma

Lemma. Let I_{Pr} denote the set of indices of elements in Pr.

Let $\Gamma = \underbrace{h(X_1)\,Y_1} \ldots \underbrace{h(X_k)\,Y_k} \ldots \underbrace{h(X_m)\,Y_m}\; h(X_{m+1})$ ($m \geq 0$)

where all Y_k have the form: $\lambda_i^{2u}\lambda_j^{2u+1}$ or $\delta_i^{2u-1}\delta_j^{2u}$ with $i, j \in I_{Pr}$

where some X_k may be empty (then considered as 1, with $h(1) = 1$)

(1) If $\Gamma \leq' 1$ then $\forall k_1 \leq m : (\forall k \leq k_1 : h(X_k)$ has no $q) \Rightarrow (\forall k \leq k_1 : Y_k \leq' 1)$

(2) If $\Gamma \leq' 1$ then $X_1 X_2 \ldots X_m X_{m+1} \leq 1$

(3) If $\delta_i^{(2k)}\Gamma\delta_j^{(2k+1)} \leq' 1$, or $\lambda_i^{(2k+1)}\Gamma\lambda_j^{(2k+2)} \leq' 1$

then $\forall k_1 \leq m : (\forall k \leq k_1 : h(X_k)$ has no $q) \Rightarrow (\forall k \leq k_1 : Y_k \leq' 1)$

(4) If $\delta_i^{(2k)}\Gamma\delta_j^{(2k+1)} \leq' 1$, or $\lambda_i^{(2k+1)}\Gamma\lambda_j^{(2k+2)} \leq' 1$ then $X_1 X_2 \ldots X_m X_{m+1} \leq 1$

The proof is technical : a key-point is that some derivations are impossible ("leftmost" introductions + parenthesizing lemma).

Some complications in the formulation (1) and (3) simplify the proof discussion we get later $p_i \not\leq p_j$ in the first form, $p_j \not\leq p_i$ in the second form

Main results

Proposition. [Equivalence property]

$$\forall X, Y \in T_{(P,\leq)} : h(X) \leq' h(Y) \text{ iff } X \leq Y$$

▷

Proposition. [Pregroup Grammar Simulation]

Given a pregroup grammar $G = (\Sigma, I, s)$ on $(P, \leq)$
we define h from a PG on $(P, \leq)$ to a PG on $(P', \leq')$;
we construct a grammar $G' = (\Sigma, h(I), h(s))$, on $(P', \leq')$
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 : $\mathcal{L}(G) = \mathcal{L}(G')$

This proposition applies 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 $P = Pr \cup Pr'$

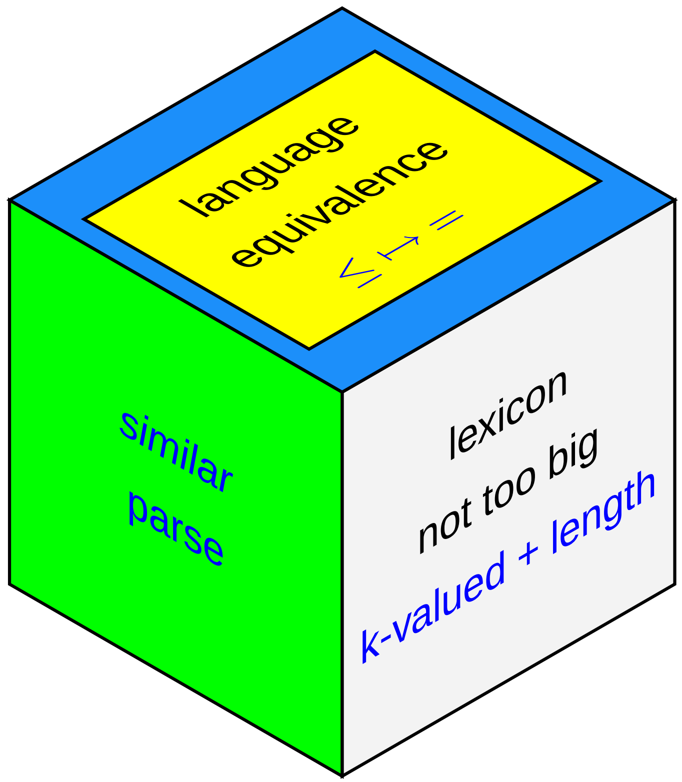
Conclusion

Pregroups were introduced as a simplification of Lambek calculus.
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 have proven 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), the size of the resulting PG is often much less that this bound;
- 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).

Transformation



Related questions

... add a preorder to other calculus, similar property ?

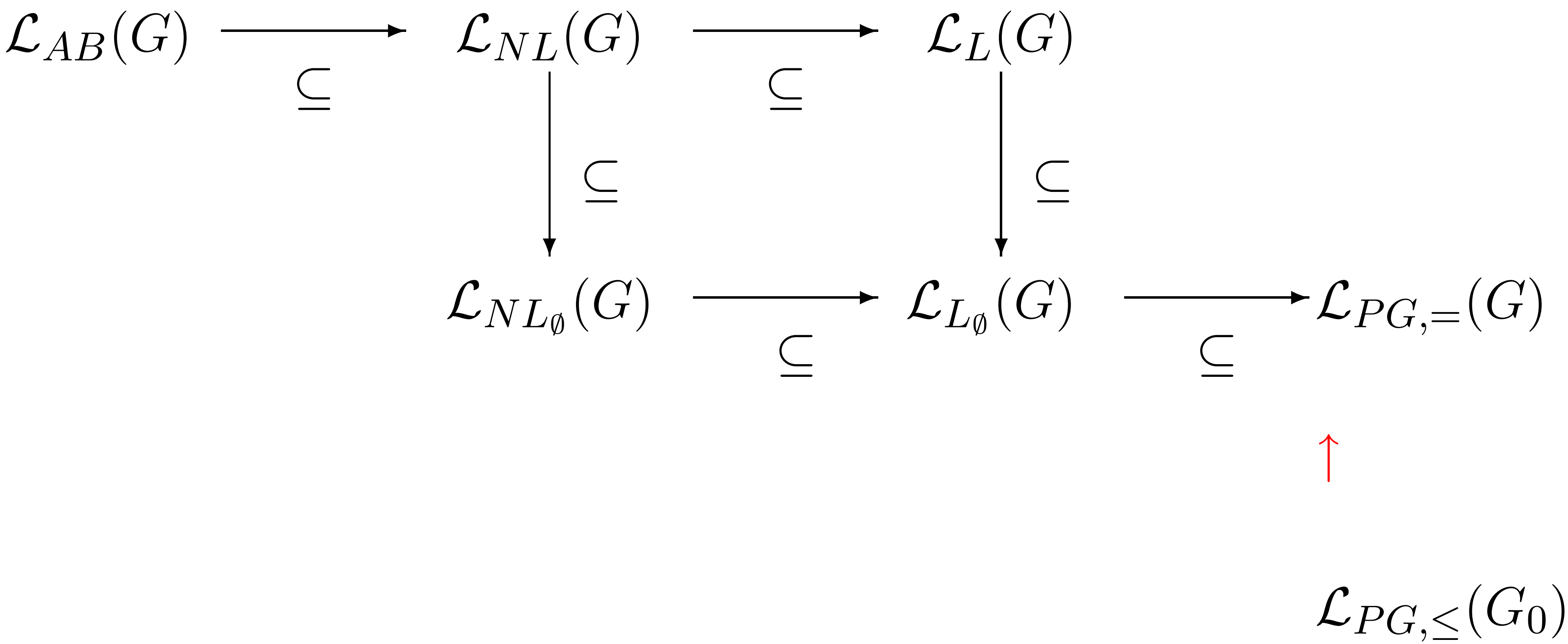
... hierarchies with/without order postulates ?

... learning with/without order postulates ?

Some other slides

for some details and links.

A schema


$\mathcal{L}_{AB}(G)$
$\subseteq$
$\mathcal{L}_{NL}(G)$
$\subseteq$
$\mathcal{L}_{L}(G)$
$\subseteq$
$\subseteq$
$\mathcal{L}_{NL_\emptyset}(G)$
$\subseteq$
$\mathcal{L}_{L_\emptyset}(G)$
$\subseteq$
$\mathcal{L}_{PG,=}(G)$
$\uparrow$
$\mathcal{L}_{PG,\leq}(G_0)$

Detailed writing and properties of the auxiliary types

The Central Lemma uses the following facts on auxiliary types :

- if $p_i \leq p_j$ then $\lambda_i^{2u}\lambda_j^{2u+1} \leq' 1$
- if $p_i \not\leq p_j$ then $\lambda_i^{2u}\lambda_j^{2u+1} \not\leq' 1$
- if $p_j \leq p_i$ then $\delta_i^{2u-1}\delta_j^{2u} \leq' 1$
- if $p_j \not\leq p_i$ then $\delta_i^{2u-1}\delta_j^{2u} \not\leq' 1$

This can be summarized as : the only Y_k, s.t. $Y_k \not\leq 1$ of central Lemma are

$\lambda_i^{2u}\lambda_j^{2u+1}$ for $p_i \not\leq p_j$

and

$\delta_i^{2u-1}\delta_j^{2u}$ for $p_j \not\leq p_i$

Detailed writing and properties of the auxiliary types

The only $Y_k \not\leq 1$ of Central Lemma are $\lambda_i^{2u}\lambda_j^{2u+1}$ for $p_i \not\leq p_j$ and $\delta_i^{2u-1}\delta_j^{2u}$ for $p_j \not\leq p_i$. This can be checked using the following detailed writing

$\lambda_i^{2u}\lambda_j^{2u+1}$ $(i=j)$: $$\alpha^{(2u)} \left(\prod_{\substack{for\ k'=n..1\\ if\ p_{k'}\leq p_i}}(\gamma_{k'}^{(2u+1)}\ \gamma_{k'}^{(2u)})\right)\left(\prod_{\substack{for\ k=1..n\\ if\ p_k\leq p_i}}(\gamma_k^{(2u+1)}\ \gamma_k^{(2u+1+1)})\right)\alpha^{(2u+1)}$$

$\delta_i^{2u-1}\delta_j^{2u}$ $(i=j)$: $$\alpha'^{\,(2u-1)} \left(\prod_{\substack{for\ k'=n..1\\ if\ p_{k'}\geq p_i}}(\beta_{k'}^{(2u-1+1)}\ \beta_{k'}^{(2u-1)})\right)\left(\prod_{\substack{for\ k=1..n,\\ if\ p_k\geq p_i}}(\beta_k^{(2u)}\ \beta_k^{(2u+1)})\right)\alpha'^{\,(2u)}$$

$\lambda_i^{2u}\lambda_j^{2u+1}$ $(i\neq j)$: $$\alpha^{(2u)} \left(\prod_{\substack{for\ k'=n..1\\ if\ p_{k'}\leq p_i}}(\gamma_{k'}^{(2u+1)}\ \gamma_{k'}^{(2u)})\right)\left(\prod_{\substack{for\ k=1..n\\ if\ p_k\leq p_j}}(\gamma_k^{(2u+1)}\ \gamma_k^{(2u+1+1)})\right)\alpha^{(2u+1)}$$

$\delta_i^{2u-1}\delta_j^{2u}$ $(i\neq j)$: $$\alpha'^{\,(2u-1)} \left(\prod_{\substack{for\ k'=n..1\\ if\ p_{k'}\geq p_i}}(\beta_{k'}^{(2u-1+1)}\ \beta_{k'}^{(2u-1)})\right)\left(\prod_{\substack{for\ k=1..n,\\ if\ p_k\geq p_j}}(\beta_k^{(2u)}\ \beta_k^{(2u+1)})\right)\alpha'^{\,(2u)}$$

Morphism-based encodings

– $\leq \mapsto \leq'$ – **Proposition.** Each order-preserving mapping from $(P, \leq)$ to $(P', \leq')$ can be uniquely extended to a unique pregroup homomorphism on $T_{(P,\leq)}$ to $T_{(P',\leq')}$.

i.e. from monotonicity on Pr :

$$p_i \leq p_j \text{ implies } h(p_i) \leq' h(p_j)$$

to monotonicity on $T_{(P,\leq)}$:

$$\text{if } X \leq Y \text{ then } h(X) \leq' h(Y)$$

$\Rightarrow$ to ensure : $p_i \leq p_j$ implies $h(p_i) \leq' h(p_j)$

Morphism-based encodings

$-\leq\mapsto\leq'-$ **Proposition.** Each order-preserving mapping from $(P,\leq)$ to $(P',\leq')$ can be uniquely extended to a unique pregroup homomorphism on $T_{(P,\leq)}$ to $T_{(P',\leq')}$.

i.e. from monotonicity on Pr :

$$p_i \leq p_j \text{ implies } h(p_i) \leq' h(p_j)$$

to monotonicity on $T_{(P,\leq)}$:

$$\text{if } X \leq Y \text{ then } h(X) \leq' h(Y)$$

Proof. The unicity comes from the three first points of the definition of pregroup homomorphism. The last point is a consequence of order-preservation which is easy by induction on a derivation $\mathcal{D}$ for $X \leq Y$

$\Rightarrow$ to ensure : $p_i \leq p_j$ implies $h(p_i) \leq' h(p_j)$

Lemma variant

Variants.

A similar result holds in the odd case (called “Bis version”) for

$$X = \underbrace{X_1\, \alpha^{(2u_1-1)}\, Y_1\, \alpha^{(2u_1)}} \dots \underbrace{X_k\, \alpha^{(2u_k-1)}\, Y_k\, \alpha^{(2u_k)}}$$
$$\dots \underbrace{X_n\, \alpha^{(2u_n-1)}\, Y_n\, \alpha^{(2u_n)}}\; X_{n+1}$$

Order 1

Lemma (Buszkowski LACL'01) If p is atomic and $o(A_i) \leq 1 \;\; for \;\; 1 \leq i \leq n$:

$A_1, \ldots, A_n \vdash p$ is derivable in AB

iff it is derivable in $L1$

iff $T(A_1) \ldots T(A_n) \leq p$ in free pregroups

where

$$T(A/B) = T(A).T(B)^l$$
$$T(B\backslash A) = T(B)^r.T(A)$$

the analogue of : $A/B, B \vdash A$

is $T(A/B).T(B) = T(A).T(B)^l.T(B) \leq T(A)$

and the analogue of : $B, B\backslash A \vdash A$

is $T(B).T(B\backslash A) = T(B).T(B)^r T(A) \leq T(A)$

On Types for Correct Sentences.

Usually, type s associated to correct sentences must be a primitive type ($s \in P$).

- here s can be any type in $T_{(P,\leq)}$.
- not a significant modification of PG because
  $X_1 \cdots X_n \leq X$ is equivalent to $X_1 \cdots X_n X^r \leq 1$.
  if $Y \leq X$ then $Y X^r \leq X X^r \leq 1$;
  if $Y X^r \leq 1$ then $Y \leq Y X^r X \leq X$

On Types for Correct Sentences.

Usually, type s associated to correct sentences must be a primitive type ($s \in P$).

- here s can be any type in $T_{(P,\leq)}$.
- not a significant modification of PG because
  $X_1 \cdots X_n \leq X$ is equivalent to $X_1 \cdots X_n X^r \leq 1$.
  if $Y \leq X$ then $Y X^r \leq X X^r \leq 1$;
  if $Y X^r \leq 1$ then $Y \leq Y X^r X \leq X$

PG G using a composed type S
into a PG G' using s, by a right *wall* $S^r s$

(the type can be seen as the type of the final point of a sentence) because $X \leq S \Leftrightarrow X S^r s \leq s$.

Main Results

Proposition. [Order-reflecting property] The simulation-morphism h from the free pregroup on $(P, \leq)$ to the free pregroup on $(P', \leq')$ satisfies (1) and (2):

(1). $\forall X, Y \in T_{(P,\leq)}$ if $h(X) \leq' h(Y)$ then $X \leq Y$

(2). $\forall X, Y \in T_{(P,\leq)}$ if $h(X) \leq' 1$ then $X \leq 1$ by $(P, \leq)$.
In fact, (1) can be shown from the central lemma (2) (case $m = 0$) and (2) above is equivalent to (1) as explained before .

As a corollary of monotonicity and previous proposition, we get:
Proposition. [Pregroup Order Simulation] The simulation-morphism h from the free pregroup on $(P, \leq)$ to the free pregroup on $(P', \leq')$ enjoys the following property:

$$\forall X, Y \in T_{(P,\leq)}\ h(X) \leq' h(Y) \text{ iff } X \leq Y$$