Parsing pregroup grammars using partial composition

Denis Béchet[(1)], Annie Foret[(2)] and Isabelle Tellier[(3)]

[(1)] LINA, University of Nantes

[(2)] IRISA, University of Rennes

[(3)] LIFL, University of Lilles III

Parsing with word separation

Parsing of “whom have you seen ?” ($q' \leq_{Pr} s$)

whom	have	you	seen
$q'o^{ll}q^l$	$qp_2^l\pi_2^l$	π_2	p_2o^l

$$\boxed{\boxed{q'o^{ll}q^l, \ \boxed{qp_2^l\pi_2^l, \pi_2}^{[1]}}^{[1]} \ , p_2o^l}^{[2]}$$

PLAN

- Introduction
- Background
  - Free pregroup
  - Pregroup grammar and language
  - Parsing
- Parsing with word separation
  - Rewriting
  - Partial composition
  - Majority partial composition
- Conclusion

Free pregroup

$$\frac{}{p^{(n)} \leq p^{(n)}}\,(Id) \qquad \frac{X \leq Y \quad Y \leq Z}{X \leq Z}\,(CUT)$$

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

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

$q \leq_{Pr} p$ if k is even or $p \leq_{Pr} q$ if k is odd

lina LABORATOIRE D'INFORMATIQUE DE NANTES ATLANTIQUE

Free pregroup grammar and language

- A grammar $G = (\Sigma, (Pr, \leq_{Pr}), I, s)$:
  - Σ finite alphabet
  - $(Pr, \leq)$ finite partially ordered set (primitive types) that defines free pregroup $(Tp, \leq_{Tp})$
  - $I \subset \Sigma \times Tp$, a lexicon, assigns a finite set of types to each $c \in \Sigma$
  - $s \in Pr$ is a primitive type for correct sentences
- The language $\mathcal{L}(G) \in \Sigma^+$:

$$v_1 \cdots v_n \in \mathcal{L}(G)$$

iff

for $1 \leq i \leq n$, $\exists X_i \in I(v_i)$ such that $X_1 \cdots X_n \leq_{Tp} s$

Parsing using rewriting (1)

Because s is a primitive type, $v_1 \cdots v_n \in \mathcal{L}(G)$ iff for $1 \leq i \leq n$, $\exists X_i \in I(v_i)$ and $\exists s' \in Pr$ such that:

$$\begin{cases} X_1 \cdots X_n \xrightarrow{(1)^*} s' \\ s' \leq_{Pr} s \end{cases}$$

$\xrightarrow{(1)^*}$: the reflexive and transitive closure of $\xrightarrow{(1)}$:

$$Xp^{(n)}q^{(n+1)}Y \xrightarrow{(1)} XY$$

if $q \leq_{Pr} p$ and n is even or if $p \leq_{Pr} q$ and n is odd

Parsing using rewriting (1) : example

Parsing of "whom have you seen ?" ($q' \leq_{Pr} s$)

whom	have	you	seen
$q'o^{ll}q^l$	$qp_2^l\pi_2^l$	π_2	p_2o^l

$$q'o^{ll}q^lqp_2^l\pi_2^l\pi_2p_2o^l \xrightarrow{(1)} q'o^{ll}p_2^l\pi_2^l\pi_2p_2o^l$$
$$\xrightarrow{(1)} q'o^{ll}p_2^lp_2o^l$$
$$\xrightarrow{(1)} q'o^{ll}o^l$$
$$\xrightarrow{(1)} q'$$

and $q' \leq_{Pr} s$

Parsing using rewriting (1) : "proof net"

Parsing of "whom have you seen ?" ($q' \leq_{Pr} s$)

whom	have	you	seen
$q'o^{ll}q^{l}$	$qp_2^{l}\pi_2^{l}$	π_2	p_2o^{l}

$$q'o^{ll}q^{l}qp_2^{l}\pi_2^{l}\pi_2p_2o^{l} \qquad (q' \leq_{Pr} s)$$

Parsing with word separation (list)

Three rewriting rules ($\Gamma, \Delta \in Tp^*$, $X, Y \in Tp$, $p, q \in Pr$):

- [M] **(merge)**: $\Gamma, X, Y, \Delta \xrightarrow{M} \Gamma, XY, \Delta$.
- [I] **(internal)**: $\Gamma, Xp^{(n)}q^{(n+1)}Y, \Delta \xrightarrow{I} \Gamma, XY, \Delta$, if $q \leq_{Pr} p$ and n is even or if $p \leq_{Pr} q$ and n is odd.
- [E] **(external)**: $\Gamma, Xp^{(n)}, q^{(n+1)}Y, \Delta \xrightarrow{E} \Gamma, X, Y, \Delta$, if $q \leq_{Pr} p$ and n is even or if $p \leq_{Pr} q$ and n is odd.

Parsing with word separation (example)

Parsing of “whom have you seen ?” ($q' \leq_{Pr} s$)

whom	have	you	seen
$q'o^{ll}q^{l}$	$qp_2^{l}\pi_2^{l}$	π_2	p_2o^{l}

$$q'o^{ll}\, q^{l}, q\, p_2^{l}\pi_2^{l}, \pi_2, p_2o^{l} \xrightarrow{E} q'o^{ll}, \boxed{p_2^{l}\pi_2^{l}, \pi_2}, p_2o^{l}$$

$$\xrightarrow{M} q'o^{ll}, p_2^{l}\pi_2^{l}\pi_2, p_2o^{l}$$

$$\xrightarrow{I} q'o^{ll}, p_2^{l}, p_2\, o^{l}$$

$$\xrightarrow{E} q'o^{ll}, \boxed{, o^{l}}$$

$$\xrightarrow{M} q'\, o^{ll}, o^{l}$$

$$\xrightarrow{E} q'$$

and $q' \leq_{Pr} s$

Parsing with word separation (lemma)

Parsing (for a pregroup grammar) can be done using $\xrightarrow{MIE^*}$:
$v_1 \cdots v_n \in \mathcal{L}(G)$ iff
for $1 \leq i \leq n$, $\exists X_i \in I(v_i)$ and $\exists s' \in Pr$ such that:

$$\begin{cases} X_1, \cdots, X_n \xrightarrow{MIE^*} s' \\ s' \leq_{Pr} s \end{cases}$$

lina LABORATOIRE D'INFORMATIQUE DE NANTES ATLANTIQUE

Internal before Merge/External (lemma)

$\xrightarrow{I}$ can be performed before $\xrightarrow{M}$ and $\xrightarrow{E}$:
$v_1 \cdots v_n \in \mathcal{L}(G)$ iff
for $1 \leq i \leq n$, $\exists X_i \in I(v_i)$, $\exists Y_i \in Tp$ and $\exists s' \in Pr$ such that:

$$\begin{cases} \text{for } 1 \leq i \leq n, X_i \xrightarrow{I^*} Y_i \\ Y_1, \cdots, Y_n \xrightarrow{ME^*} s' \\ s' \leq_{Pr} s \end{cases}$$

Internal before Merge/External (example)

$$q'o^{ll}\, q^l, q\, p_2^l\pi_2^l, \pi_2, p_2o^l \xrightarrow{E} q'o^{ll}, \boxed{p_2^l\pi_2^l, \pi_2}, p_2o^l$$

$$\xrightarrow{M} q'o^{ll}, p_2^l\pi_2^l\pi_2, p_2o^l$$

$$\xrightarrow{I} q'o^{ll}, p_2^l, p_2\, o^l$$

$$\xrightarrow{E} q'o^{ll}, \boxed{, o^l} \xrightarrow{M} q'\, o^{ll}, o^l \xrightarrow{E} q'$$

becomes:

$$q'o^{ll}\, q^l, q\, p_2^l\pi_2^l, \pi_2, p_2o^l \xrightarrow{E} q'o^{ll}, p_2^l\, \pi_2^l, \pi_2, p_2o^l$$

$$\xrightarrow{E} q'o^{ll}, \boxed{p_2^l,}, p_2o^l$$

$$\xrightarrow{M} q'o^{ll}, p_2^l, p_2\, o^l$$

$$\xrightarrow{E} q'o^{ll}, \boxed{, o^l} \xrightarrow{M} q'\, o^{ll}, o^l \xrightarrow{E} q'$$

Partial composition

$\xrightarrow{E^*}$ and $\xrightarrow{M}$ corresponding to the same couple of types are joined together in $\xrightarrow{C}$:

- **[C] (partial composition)**: For $k \in \mathbb{N}$,
  $\Gamma, X p_1^{(n_1)} \cdots p_k^{(n_k)}, q_k^{(n_k+1)} \cdots q_1^{(n_1+1)} Y, \Delta \xrightarrow{E} \Gamma, XY, \Delta$, if $q_i \leq_{Pr} p_i$ and n_i is even or if $p_i \leq_{Pr} q_i$ and n_i is odd, for $1 \leq i \leq k$.

lina LABORATOIRE D'INFORMATIQUE DE NANTES ATLANTIQUE

Partial composition (example)

$$
\begin{aligned}
q' o^{ll}\, q^l, q\, p_2^l \pi_2^l, \pi_2, p_2 o^l \quad & \xrightarrow{E} q' o^{ll}, p_2^l\, \pi_2^l, \pi_2\, , p_2 o^l \\
& \xrightarrow{E} q' o^{ll}, \boxed{p_2^l,}\, , p_2 o^l \\
& \xrightarrow{M} q' o^{ll}, \, p_2^l, p_2 \, o^l \\
& \xrightarrow{E} q' o^{ll}, \boxed{, o^l} \xrightarrow{M} q' \, o^{ll}, o^l \xrightarrow{E} q'
\end{aligned}
$$

becomes:

$$
\begin{aligned}
q' o^{ll}\, q^l, q\, p_2^l \pi_2^l, \pi_2, p_2 o^l \quad & \xrightarrow{E} q' o^{ll}, \boxed{p_2^l \pi_2^l, \pi_2}^{[1]} , p_2 o^l \\
& \xrightarrow{C} q' o^{ll}, \, p_2^l, p_2 \, o^l \\
& \xrightarrow{E} q' o^{ll}, \boxed{, o^l}^{[0]} \xrightarrow{C} q' \, o^{ll}, o^l \xrightarrow{E} q'
\end{aligned}
$$

Partial composition (lemma)

Lemma:
For a list of types Γ and $p \in Pr$, $\Gamma \xrightarrow{ME^*} p$ iff $\Gamma \xrightarrow{C^*} p$

Corollary:
$v_1 \cdots v_n \in \mathcal{L}(G)$ iff
for $1 \leq i \leq n$, $\exists X_i \in I(v_i)$, $\exists Y_i \in Tp$ and $\exists s' \in Pr$ such that:

$$\begin{cases} \text{for } 1 \leq i \leq n, X_i \xrightarrow{I^*} Y_i \\ Y_1, \cdots, Y_n \xrightarrow{C^*} s' \\ s' \leq_{Pr} s \end{cases}$$

lina LABORATOIRE D'INFORMATIQUE DE NANTES ATLANTIQUE

Partial composition (example)

$$q' o^{ll}\, q^l, q\, p_2^l \pi_2^l, \pi_2, p_2 o^l \xrightarrow{E} q' o^{ll}, p_2^l\, \pi_2^l, \pi_2\, , p_2 o^l$$

$$\xrightarrow{E} q' o^{ll}, \boxed{p_2^l,}, p_2 o^l$$

$$\xrightarrow{M} q' o^{ll},\, p_2^l, p_2\, o^l$$

$$\xrightarrow{E} q' o^{ll}, \boxed{, o^l} \xrightarrow{M} q'\, o^{ll}, o^l \xrightarrow{E} q'$$

becomes:

$$\boxed{q' o^{ll} q^l, q p_2^l \pi_2^l}^{[1]}, \pi_2, p_2 o^l \xrightarrow{C} \boxed{q' o^{ll} p_2^l \pi_2^l, \pi_2}^{[1]}, p_2 o^l$$

$$\xrightarrow{C} \boxed{q' o^{ll} p_2^l, p_2 o^l}^{[2]}$$

$$\xrightarrow{C} q'$$

Partial composition (parse tree)

$$\left[q' o^{ll} q^l, q p_2^l \pi_2^l\right]^{[1]}, \pi_2, p_2 o^l \xrightarrow{C} \left[q' o^{ll} p_2^l \pi_2^l, \pi_2\right]^{[1]}, p_2 o^l$$

$$\xrightarrow{C} \left[q' o^{ll} p_2^l, p_2 o^l\right]^{[2]}$$

$$\xrightarrow{C} q'$$

corresponds to the following parse tree:

$$\left[\left[\left[q' o^{ll} q^l, q p_2^l \pi_2^l\right]^{[1]}, \pi_2\right]^{[1]}, p_2 o^l\right]^{[2]}$$

Parsing using partial composition

Partial composition does not give a polynomial parsing algorithm because the result of partial composition is not bounded by the lexicon:

$$\Gamma, \boxed{q' o^{ll} q^{l}, q p_2^{l} \pi_2^{l}}^{[1]}, \Delta \xrightarrow{C} \Gamma, q' o^{ll} p_2^{l} \pi_2^{l}, \Delta$$

$$3\ ,\ 3 \qquad\qquad\qquad 4$$

Majority partial composition

A partial composition $\xrightarrow{C}$ is a majority partial composition (noted $\xrightarrow{@}$) if the width of the result is less or equal to the maximum of the widths of the arguments

A non majoritory partial composition:

$$\Gamma, \boxed{q'o^{ll}q^{l}, qp^{l}_{2}\pi^{l}_{2}}^{[1]}, \Delta \xrightarrow{C} \Gamma, q'o^{ll}p^{l}_{2}\pi^{l}_{2}, \Delta$$

A majoritory partial composition:

$$\Gamma, \boxed{q'o^{ll}q^{l}, qo^{l}\pi^{l}_{2}}^{[2]}, \Delta \xrightarrow{@} \Gamma, q'\pi^{l}_{2}, \Delta$$

Parsing using majority composition

Main theorem:

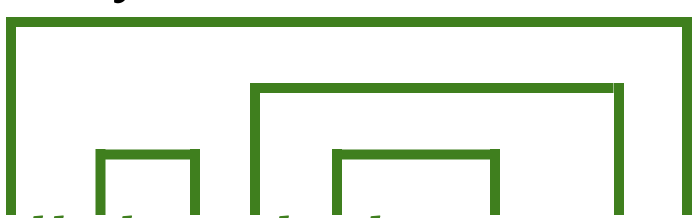
$v_1 \cdots v_n \in \mathcal{L}(G)$ iff

for $1 \leq i \leq n$, $\exists X_i \in \mathcal{R}_{I^*}(I)(v_i)$, and $\exists s' \in Pr$ such that:

$$\begin{cases} X_1, \cdots, X_n \xrightarrow{@^*} s' \\ s' \leq_{Pr} s \end{cases}$$

where $\mathcal{R}_{I^*}(I)$ is the completion of I by $\xrightarrow{I^*}$

Proof: property of (planar) proof nets. there exists a type in Γ that is only linked to its immediate neighour(s)



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

Parsing using @ (example)

$$\boxed{q' o^{ll} q^l, q p_2^l \pi_2^l}^{[1]}, \pi_2, p_2 o^l \xrightarrow{C} \boxed{q' o^{ll} p_2^l \pi_2^l, \pi_2}^{[1]}, p_2 o^l$$

$$\xrightarrow{C} \boxed{q' o^{ll} p_2^l, p_2 o^l}^{[2]}$$

$$\xrightarrow{C} q'$$

is transformed into:

$$q' o^{ll} q^l, \boxed{q p_2^l \pi_2^l, \pi_2}^{[1]}, p_2 o^l \xrightarrow{@} \boxed{q' o^{ll} q^l, q p_2^l}^{[1]}, p_2 o^l$$

$$\xrightarrow{@} \boxed{q' o^{ll} p_2^l, p_2 o^l}^{[2]}$$

$$\xrightarrow{@} q'$$

Polynomial parsing using @

Partial composition gives a polynomial parsing algorithm because the result of partial composition is bounded by the maximum width of the types of the lexicon.

For a grammar G and a list of words $v_1, \cdots, v_n \in \Sigma^+$, we compute for $1 \leq i \leq j \leq n$, $T^G_{v_1,\cdots,v_n}(i,j) \subset Tp$, the possible types associated to the sublist of words $v_i, \cdots, v_j$ using majority partial composition:

$$i = j: \quad T^G_{v_1,\cdots,v_n}(i,j) = \mathcal{R}_{I^*}(I)(v_i)$$

$$i < j: \quad T^G_{v_1,\cdots,v_n}(i,j) = \bigcup_{k=i}^{j-1} \left\{ Z \;\middle|\; \begin{array}{c} \exists X \in T^G_{v_1,\cdots,v_n}(i,k) \\ \exists Y \in T^G_{v_1,\cdots,v_n}(k+1,j) \\ X, Y \xrightarrow{@} Z \end{array} \right\}$$

Polynomial parsing using @

$v_1, \cdots, v_n \in \mathcal{L}(G)$ iff
$\exists s' \in Pr$ such that $s' \in T^G_{v_1,\cdots,v_n}(1,n)$ and $s' \leq_{Pr} s$

Algorithm:

1. Search the types associated by G to each word
2. Add the types deduced by $\xrightarrow{I^*}$
3. Compute recursively the possible types associated to a contiguous segment of words of the string using $\xrightarrow{@}$
4. Look at the primitive types associated to the complete string

Polynomial parsing using @ (example)

Parsing of “whom have you seen ?”

1. Lexicon:
$$\begin{array}{|lcl} \text{whom} & \mapsto & \{q' o^{ll} q^l\} \\ \text{have} & \mapsto & \{q p_2^l \pi_2^l\} \\ \text{you} & \mapsto & \{\pi_2\} \\ \text{seen} & \mapsto & \{p_2 o^l\} \end{array}$$

2. Completion of the lexicon using $\xrightarrow{I}$: nothing to add

lina LABORATOIRE D'INFORMATIQUE DE NANTES ATLANTIQUE

Polynomial parsing using @ (example)

Parsing of "whom have you seen ?"

3. Types associated to segment of words:

- Length = 1:

whom	have	you	seen
$\{q'o^{ll}q^l\}$	$\{qp_2^l\pi_2^l\}$	$\{\pi_2\}$	$\{p_2o^l\}$

- Length = 2:

whom have	have you	you seen
$\emptyset$	$\{qp_2^l\}$	$\emptyset$

- Length = 3:

whom have you	have you seen
$\{q'o^{ll}p_2^l\}$	$\{qo^l\}$

- Length = 4:

whom have you seen
$\{q'$ and $q'o^{ll}o^l\}$

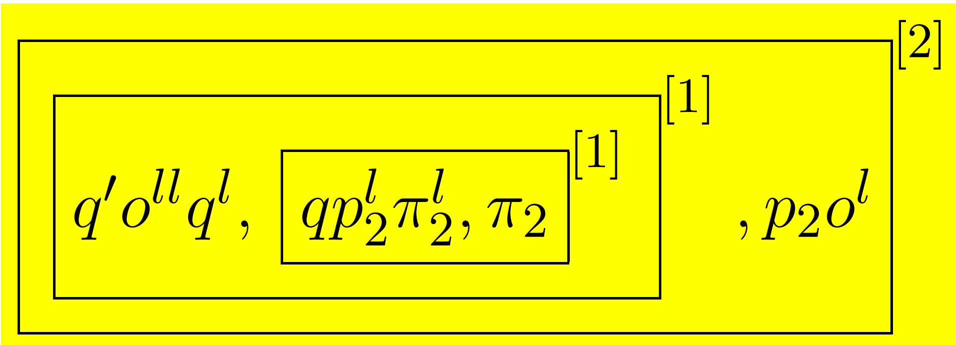
Polynomial parsing using @ (example)

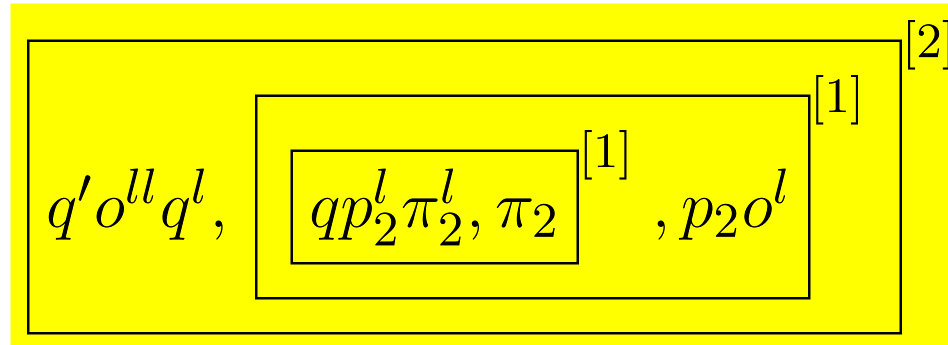
Parsing of “whom have you seen ?”

4. Primitive types for the string: q' and $q' \leq_{Pr} s$
   $\Longrightarrow$ this is a correct sentence

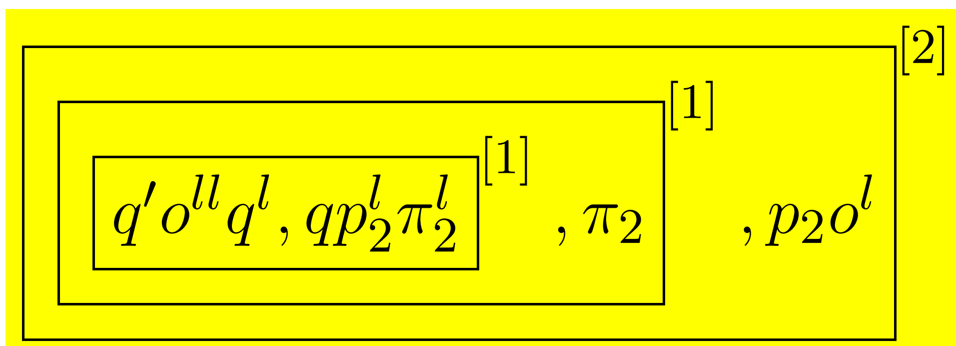
Polynomial parsing using @ (example)

Associativity usually gives several parse trees:





Remark:



is not a parse tree (one partial composition is not a majority partial composition)

Conclusion

- Polynomial parsing algorithm using majority partial composition of types associated to the words of a string
- Need to complete the lexicon with types deduced using “internal” rewriting
- Can be adapted to associative Lambek calculus (using modules)