Semantic Pregroup Grammars Handle Long Distance Dependencies in French

Anne Preller

Semantic pregroup grammars handle long distance dependencies in French

Anne Preller
Lirmm - CNRS, 161, rue Ada, 343924 Montpellier Cedex 5

Résumé. Nous présentons une grammaire de prégroupe traitant l'accord entre le sujet ou l'objet antéposé avec le participe passé, actif ou passif, comprenant les verbes modaux. La grammaire est munie d'une interprétation sémantique respectant les dépendances non-bornées.

Abstract. A pregroup grammar is presented which handles distant agreement of features in French, including modal verbs, clitics, relative pronouns, the compound past and the passive mode. The grammar has a semantic interpretation into predicate logic which captures the unbounded dependencies.

Mots-clés : Grammaires catégorielles, grammaires de prégroupe, dépendances non-bornées, clitiques, pronoms relatifs, interprétation sémantique.

Keywords: Categorial grammars, pregroup grammars, distant dependencies, agreement of features, French clitics, French pronouns, semantic interpretation.

1 Interpretation in predicate logic

Pregroup grammars belong to the family of categorial grammars and were introduced in (Lambek 1999) as a simplification of the earlier syntactic calculus, now known as Lambek Calculus. Though categorial grammars based on Lambek calculus can be translated into pregroup grammars, the translated pregroup grammar may be stronger than the original one and therefore overgenerate. Moreover, the inherent higher order semantical interpretation of categorial grammars is lost for pregroup grammars. Here, we want to show that the meaning of a sentence with long distance agreement based on an analysis by pregroup grammars can be defined in two-sorted predicate logic. The semantical interpretation used here was introduced in (Preller). The main idea is to accompany the lexical entries in the dictionary by one or more logical expressions translating the entry. The translation of a sentence is computed from the translation of the words and from a reduction to the sentence type.

A pregroup grammar consists of a dictionary, associating to each word a finite number of *types*. Types are strings of *simple types*, i.e. of the form

$$a_1^{(z_1)} \dots a_k^{(z_k)},$$

where $a_1, \dots, a_k$ are basic types and $z_1, ..., z_k \in \mathbb{Z}$. The set of *basic types* B is partially ordered by $\rightarrow$ and includes the syntactical types, e.g. the sentence type $\boldsymbol{s}$. When parsing a sentence with a pregroup grammar one assigns to each word a type from the dictionary and constructs a derivation to the sentence type $\boldsymbol{s}$ by the following rules

$$\begin{array}{ll} \text{(Induced step)} & Xa^{(z)}Y \rightarrow Xb^{(z)}Y \\ \text{(Generalized contraction)} & Xa^{(z)}b^{(z+1)}Y \rightarrow XY \end{array},$$

where X and Y are arbitrary types, a and b are basic and either z is even and $a \rightarrow b$ or z is odd and $b \rightarrow a$. In the following, we write 1 for the empty string a^ℓ for $a^{(-1)}$and a^r for $a^{(1)}$ and refer to them a *adjoints* of a, whereas $a^{\ell\ell} = a^{(-2)}$, $a^{rr} = a^{(2)}$ are *iterated* adjoints.

For example, consider the dictionary

$$\begin{array}{ll} \textit{Marie} & : \pi_{3\mathtt{fs}}, o \\ \textit{Jean} & : \pi_{3\mathtt{ms}}, o \\ \textit{examine} & : \pi^r_{3\mathtt{s}} \boldsymbol{s} o^\ell \end{array}.$$

The basic type $\pi_{3\mathtt{fs}}$ corresponds to 'subject third person feminine singular', or more generally, π_{pgn} to 'subject of person p, gender g and number n', where $p \in \{1, 2, 3\}$, $g \in \{\mathtt{m}, \mathtt{f}\}$ and $n \in \{\mathtt{s}, \mathtt{p}\}$. Here, m stands for 'masculine', f for 'feminine', s for 'singular' and p for 'plural'. We also have the basic types π_{pn} for the subject when only the person and the number matter and π when person, gender and number do not matter. The basic types o and $\boldsymbol{s}$ stand for 'direct object' respectively for 'sentence in the present'. It is assumed that

$$\pi_{pgn} \rightarrow \pi_{pn} \rightarrow \pi, \ \text{ for } p \in \{1, 2, 3\}, \ g \in \{\mathtt{m}, \mathtt{f}\} \text{ and } n \in \{\mathtt{s}, \mathtt{p}\}.$$

To analyze a string of words, choose types from the dictionary and concatenate them in the order of the words. The string of words is a sentence of the grammar if and only if the concatenated type has a derivation to the sentence type. For example,

Marie examine Jean

(MARY EXAMINES JOHN)

$$(\pi_{3\mathtt{fs}})\ (\pi^r_{3\mathtt{s}}\ \boldsymbol{s}\ o^\ell)\ (o) \quad \rightarrow \boldsymbol{s}$$

This derivation is justified by the generalized contractions $\pi_{3\mathtt{fs}}\pi_{3\mathtt{s}}^r \to 1$ and $o^\ell o \to 1$. As customary, the types have been written under the words and the generalized contractions are indicated by under-links.

We illustrate the semantical interpretation by a few examples, following (Preller). The sentence

$$Marie\ examine\ Jean\ (\text{MARY EXAMINES JEAN})$$

is usually rendered in predicate logic by

$$\mathtt{examiner}(\mathtt{marie}, \mathtt{jean}).$$

As usual, transitive verbs like *examiner* are interpreted by binary relations, here embodied by the binary relational symbol $\mathtt{examiner}(x_1, x_2)$. Looking at the type $\pi_{3\mathtt{s}}^r \boldsymbol{s} o^\ell$ of *examine*, we may argue that the basic type $\boldsymbol{s}$ corresponds to the relational symbol and that the non-basic types determine the argument places. We may even go further and make correspond a particular non-basic type to a particular argument place, here ${\pi_{3\mathtt{s}}}^r$ to the first argument place, x_1, and o^ℓ to the second, x_2. Accordingly, the types for proper names, which are just single basic types, do not introduce argument places and are translated by individual constants. Hence we may add a semantic *translation* for each entry in the dictionary above

$$\begin{array}{lll} Marie & : \pi_{3\mathtt{fs}}, o & \mathtt{marie} \\ Jean & : \pi_{3\mathtt{ms}}, o & \mathtt{jean} \\ examine & : \pi_{3\mathtt{s}}^r \boldsymbol{s} o^\ell & \mathtt{examiner}(x_1, x_2) \end{array} .$$

The translation depends both on the word and its chosen type. For each lexical entry we can create new non-logical symbols or reuse others, introduced earlier.

The reduction of the sentence

$$\begin{array}{lllll} \textit{Marie} & \textit{examine} & \textit{Jean} & & \\ (\pi_{3\mathtt{fs}}) & (\pi_{3\mathtt{s}}^r\ \boldsymbol{s}\ o^\ell) & (o) & \to \boldsymbol{s} \end{array}$$

suggests that the translating formula $\mathtt{examiner}(\mathtt{marie}, \mathtt{jean})$ can be computed by substitution according to the links. The under-link from $\pi_{3\mathtt{fs}}$ to $\pi_{3\mathtt{s}}^r$ tells us that the constant $\mathtt{marie}$ translating the basic type $\pi_{3\mathtt{fs}}$ occupies the first argument place x_1. Similarly, the under-link from o^ℓ to o puts the second constant $\mathtt{jean}$ into the second argument place.

Generalizing these heuristic considerations, we may agree that a translation of a given lexical entry $word : t_1 \dots t_n$ respects the following rules :

- each basic type t_i is translated by a functional or relational symbol,

- each non-basic type t_i corresponds to an argument-place of at least one functional or relational symbol of the entry,

-the translation of a sentence, via a reduction to the sentence type, is computed by substituting according to the links of the reduction.

Computing the translation of a sentence from the translation of its words makes the translation clearly compositional. Only one rule is needed to explain how the parts are to be composed, namely substitution. We will illustrate this translation mechanism by our sample sentences, beginning with the compound past of transitive words. A warning to the reader : the type assigned below to the past participle of transitive words corresponds to the case when it is used to form the compound past of the active form. In later examples concerning the passive, it will be assigned a different type. No claim to be exhaustive is made of course, new types may always be added without undoing the already recognized sentences due to the conservativity of extensions of pregroup grammars.

Syntax without translation

Suppose we added a new basic type $\boldsymbol{p}$ standing for the past participle and the lexical entries *examiné* : $\boldsymbol{p}\, o^\ell$ and *a* : $\pi^r_{3s}\boldsymbol{s}\boldsymbol{p}^\ell$ to our dictionary. The augmented dictionary would recognize the sentence *Jean a examiné Marie* (JOHN HAS EXAMINED MARY) by the reduction

$$\begin{array}{cccc} \textit{Jean} & \textit{a} & \textit{examiné} & \textit{Marie} \\ (\pi_{3ms}) & (\pi^r_{3s}\,\boldsymbol{s}\,\boldsymbol{p}^\ell) & (\boldsymbol{p}\,o^\ell) & (o) \end{array} \quad .$$

However, the entry *examiné* : $\boldsymbol{p}\, o^\ell$ would correspond to a unary relation. That means that the relation translating a transitive verb in the past would depend only on the object. Moreover, the semantic role of the auxiliary would - correctly - provide the temporal aspect, but depend on the acting individual(s). Therefore, the translation would be a temporal operator that depends on individuals in opposition to the usual formalizations and interpretations of temporal operators.

Syntax with translation

Adopting the following types 'with translation'

$$\begin{array}{lll} \textit{examiné} & : \pi^r\boldsymbol{p}o^\ell & \mathtt{examiner}(x_1, x_2) \\ \textit{a} & : \pi^r_{3s}\boldsymbol{s}\boldsymbol{p}^\ell\pi_{3s} & \mathtt{avoir}(y)\ \ \mathtt{id}(x) \end{array} \quad ,$$

we get the reduction

$$\begin{array}{cccc} \textit{Jean} & \textit{a} & \textit{examiné} & \textit{Marie} \\ (\pi_{3ms}) & (\pi^r_{3s}\ \boldsymbol{s}\ \boldsymbol{p}^\ell\,\pi_{3s}) & (\pi^r\ \boldsymbol{p}\ o^\ell) & (o) \end{array} \quad .$$

Now we can correctly interpret the past participle by a binary relation, the same as for other forms of the verb. Next, the type $\pi^r_{3s}\boldsymbol{s}\boldsymbol{p}^\ell\pi_{3s}$ for the auxiliary *a* is now a string of four simple types two of which are basic types, namely $\boldsymbol{s}$ and π_{3s}. The other two are the right adjoint π^r_{3s} and the left adjoint $\boldsymbol{p}^\ell$, which correspond to an argument-place x and to an argument place y in this order. Each of the two basic types $\boldsymbol{s}$ and π_{3s} of the entry is associated to a non-logical symbol, namely $\boldsymbol{s}$ to the predicate symbol `avoir` and π_{3s} to the functional symbol `id`. The latter depends on the argument-place x given by π^r_{3s} , whereas `avoir` depends on the argument-place y (corresponding to $\boldsymbol{p}^\ell$) . This means that the semantic translation of a single word may comprise several logic expressions.

The correspondence between simple types and non-logical symbols is

$$\begin{array}{llllll} \textit{a} & : & \pi^r_{3s} & \boldsymbol{s} & \boldsymbol{p}^\ell & \pi_{3s} \\ & & x & \mathtt{avoir} & y & \mathtt{id} \\ \textit{examiné} & : & \pi^r & \boldsymbol{p} & o^\ell & \\ & & x_1 & \mathtt{examiner} & x_2 & \end{array} \quad .$$

The reduction

$$\begin{array}{cccc} \textit{Jean} & \textit{a} & \textit{examiné} & \textit{Marie} \\ \mathtt{john} & x\ \mathtt{avoir}\ y\ \mathtt{id} & x_1\ \mathtt{examiner}\ x_2 & \mathtt{marie} \\ (\pi_{3ms}) & (\pi^r_{3s}\ \ \boldsymbol{s}\ \ \boldsymbol{p}^\ell\,\pi_{3s}) & (\pi^r\ \ \boldsymbol{p}\ \ o^\ell) & (o) \end{array} \quad ,$$

defines the substitutions

$$[x \mid \texttt{jean}],\ [y \mid \texttt{examiner}],\ [x_1 \mid \texttt{id}] \text{ and } [x_2 \mid \texttt{marie}]\,.$$

The translation of the sentence above is now obtained by substituting according to the underlinks, i.e.

$$\texttt{avoir}(\texttt{examiner}(\texttt{id}(\texttt{jean}), \texttt{marie})).$$

The functional symbol `id` only serves to push the subject from the left side of the auxiliary to the right, i.e. it behaves like the identity function. This is expressed by the non-logical axiom

$$\texttt{id}(x) = x$$

Using the equality $\texttt{id}(\texttt{jean}) = \texttt{jean}$, we derive

$$\texttt{avoir}(\texttt{examiner}(\texttt{jean}, \texttt{marie}))$$

The structure of this expression suggests to read `avoir` as a modal operator applied to an atomic formula. Axioms could be added to the logic to express its temporal meaning, but this goes beyond the scope of our endeavor here. The example of the auxiliary shows that the translation of certain words may comprehend more than one expression of the logic. The predicate symbol `avoir` renders the fact that a sentence is translated by a formula. The functional symbol `id` serves a purpose similar to that of an index in HPSG's : it is used to handle unbounded dependencies.

More generally, the infinitive of a verb will have the same number of arguments as its finite forms, but its type does not depend on the person, gender or number. Hence in the case of the infinitive, we choose the following types and corresponding translations

$$\begin{array}{ll} \textit{avoir} \quad : \pi^r \boldsymbol{ip}^\ell \pi & \texttt{avoir}(y)\ \ \texttt{id}(x) \\ \textit{examiner} : \pi^r \boldsymbol{io}^\ell & \texttt{examiner}(x_1, x_2) \end{array}.$$

The logic underlying this semantic interpretation is two-sorted first order logic, one sort for individuals and the other one for sets of individuals, with two primitive relational symbols, namely $\in$ and $=$. It is equivalent to Henkin's system of second order logic with general models, see (v. Benthem 2005).

Before continuing the presentation of the grammar, we want to connect the subject types $\pi^r, \pi^r_{3\mathrm{s}}$ in the past participle or the infinitive with the higher order types of categorial grammars. The usual translation from categorial grammars to pregroup grammars is defined by $A/B \longmapsto AB^\ell$ and $B\backslash A \longmapsto B^r A$. An example why this translation can lead to an overgenerating pregroup grammar is studied in (Moortgat-Oehrle). It concerns the type of the relative pronoun *that*, for example in the expression *book that Alice found*. The types in Non-associative Lambek calculus with modal operators and their translation into pregroup calculus are

$$\begin{array}{lll} \textit{book} & : n & n \\ \textit{that} & : (n\backslash n)/(\boldsymbol{s}/\Diamond\Box np) & n^r n\, (o*)^{\ell\ell} \boldsymbol{s}^\ell \\ \textit{Alice} & : np & o \\ \textit{found} & : (np\backslash \boldsymbol{s})/np & o^r \boldsymbol{s} o^\ell \end{array},$$

where we have written $o*$ for $\Diamond\Box np$ and o for np. The basic types are identical in both grammars. The compound type $\Diamond\Box np$ of the NL-grammar is assimilated to a basic type $o* = \Diamond\Box np$

in the pregroup grammar. Moreover, $o* \to o$ is postulated in (Lambek 2004). However, this pregroup grammar overgenerates. It recognizes both the grammatical

$$\begin{array}{l} \textit{book} \quad \textit{that} \quad \textit{Alice found} \\ (n)\ (n^r\ n\ o*^{\ell\ell}\ \boldsymbol{s}^\ell)\ (o)\ (o^r\ \boldsymbol{s}\ o^\ell\ ) \to n\,. \end{array} \tag{1}$$

and, incorrectly, the non-grammatical

$$\begin{array}{l} {*}\textit{book} \quad \textit{that} \quad \textit{Alice found} \quad \textit{it} \quad \textit{and} \\ (n)\ (n^r\ n\ o*^{\ell\ell}\ \boldsymbol{s}^\ell)\ (o)\ (o^r\ \boldsymbol{s}\ o^\ell)\ (o)\ (o^r\ o\ o^\ell\ ) \to n\,. \end{array} \tag{2}$$

The derivation responsible for the overgeneration is characterized in *loc.cit.*and used to formulate a rule that excludes this sort of derivations. The result is an enriched grammar with new rules, similar to the constraints on movement in transformational grammars.

A closer look at the difference between the NL-derivation and the pregroup derivation makes it possible to define an ordinary pregroup grammar that recognizes (1), rejects (2) and assigns an appropriate semantic interpretation to (the type of) *that*. In the pregroup derivation (1), the type of *Alice found* is computed directly from the types listed in the dictionary, namely

$$\begin{array}{l} \textit{Alice} \quad \textit{found} \\ (o)\ (o^r\ \boldsymbol{s}\, o^\ell\ ) \to \boldsymbol{s} o^\ell \end{array} \tag{3}$$

The modal operators, on the contrary, transform $(Alice \circ found) \vdash \boldsymbol{s}/np$ to

$$(Alice \circ found) \vdash \boldsymbol{s}/\Diamond\Box np \tag{4}$$

before it is concatenated with the type of *that* :

$$\dfrac{\dfrac{book}{n} \quad \dfrac{\dfrac{that}{(n\backslash n)/(\boldsymbol{s}/\Diamond\Box np)} \quad \dfrac{\dfrac{\dfrac{\dfrac{Alice}{np} \quad \dfrac{\dfrac{found}{(np\backslash \boldsymbol{s})/np} \quad \dfrac{\Box np \vdash \Box np}{\Diamond\Box np \vdash np}}{found \circ \Diamond\Box np \vdash np\backslash \boldsymbol{s}}(\backslash E)}{Alice \circ (found \circ \Diamond\Box np) \vdash \boldsymbol{s}}}{(Alice \circ found) \circ \Diamond\Box np \vdash \boldsymbol{s}}}{Alice \circ found \vdash (\boldsymbol{s}/\Diamond\Box np)}}{that \circ (Alice \circ found) \vdash (n\backslash n)}}{book \circ (that \circ (Alice \circ found)) \vdash n} \tag{5}$$

As pregroup grammars only use types from the dictionary, we must anticipate the type (4), by adding $found\,:\, o^r \boldsymbol{s} o*^\ell$ to our dictionary, but take care to keep the two basic types o and $o*$ unrelated. Hence $o \not\to o*$ in the revised pregroup grammar. The entry $found : o^r \boldsymbol{s} o^\ell$ is retained, so that the pregroup dictionary now lists

$$found : o^r \boldsymbol{s} o^\ell, o^r \boldsymbol{s} o*^\ell \quad .$$

As now $o*$ is isolated in the set of basic types, we may rename $o*^{\ell\ell}$ as $\bar{o}$ and therefore change $o*^\ell$ to $\bar{o}^r$ without changing derivations. The resulting dictionary

$book$	$: n$	n
$that$	$: (n\backslash n)/(\boldsymbol{s}/\Diamond\Box np)$	$n^r n\, \bar{o} \boldsymbol{s}^\ell$
$Alice$	$: np$	o
$found$	$: (np\backslash \boldsymbol{s})/np$	$o^r \boldsymbol{s} o^\ell,\ o^r \boldsymbol{s} \bar{o}^r$

is strongly equivalent to the dictionary before the replacement.[1] More generally it can be shown that due to compactness an arbitrary pregroup dictionary is strongly equivalent to one with no iterated adjoints. The latter has a semantic interpretation, for example the 'dummy' $\bar{o}$ in the type of *that* will play the role of a temporary name for the set of entities satisfying the following relative clause. If the auxiliary *avoir* is seen as a map from predicates to predicates, a 'dummy' is introduced in its pregroup type by the translation as indicated above.

2 Distant agreement in French

French clitics have already been studied with pregroup grammars in (Bargelli-Lambek), but without agreement. Our analysis differs from that given in the latter for two reasons. First of all, we want to avoid the meta-rule used there and base the analysis inside an ordinary pregroup grammar. The other reason is that we prefer to think of clitics as designating individuals or sets of individuals, not operators on relations. Due to the restricted space, only agreement with the preverbal personal pronoun in the role of a direct object is presented in some detail.

In the compound past of the active form, the past participle agrees in gender and number with the direct object clitic. If the verb is in passive mode or forms its compound past with the auxiliary *être*, the past participle agrees in gender and number with the subject. Reflexive pronouns, which are perverbal in French, agree with the subject.

We add to the basic types of the preceding section new basic types for direct object clitics o_{pgn}, depending on the features of person $p = 1, 2, 3$, gender $g = \mathtt{m}, \mathtt{f}$ and number $n = \mathtt{s}, \mathtt{p}$. The 'dummies' $\hat{o}_{pgn}$ and $\hat{o}$ will capture distant dependencies. The types $\hat{o}_{gn}$ are used if only gender and number matter, but not the person. The dependence of the type of the clitic on the person makes it possible to avoid non grammatical combinations of two clitics, like **me lui*, but we do not pursue this topic here. We assume

$\hat{o}_{pgn} \to \hat{o}_{gn} \to \hat{o}$, $o_{pgn} \to \hat{o}_{gn} \to \hat{o}$, for $p = 1, 2, 3$, $g = \mathtt{m}, \mathtt{f}$ and $n = \mathtt{s}, \mathtt{p}$.

We use the following sentences to illustrate how pregroup grammars can handle syntactical and semantical agreement :

Marie les examine	*Marie s'examine*
(MARY EXAMINES THEM)	(MARY EXAMINES HERSELF)
Marie les a examinés	*Marie s'est examinée*
(MARY HAS EXAMINED THEM)	(MARY HAS EXAMINED HERSELF)
Marie doit les examiner	*Marie doit s'examiner*
(MARY MUST EXAMINE THEM)	(MARY MUST EXAMINE HERSELF)
Marie doit les avoir examinés	*Marie doit s'être examinée*
(MARY MUST HAVE EXAMINED THEM)	(MARY MUST HAVE EXAMINED HERSELF)
Marie est examinée par Jean	*Marie est examinée*
(MARY IS EXAMINED BY JOHN)	(MARY IS EXAMINED)

The lexical entries for the personal pronoun *les* and the reflexive pronoun *s'* are

$$\begin{array}{lll} \textit{les}: & o_{3g\mathtt{p}} & \mathtt{Les} \\ \textit{s'}: & \pi^r_{3gn}\pi_{3gn}\hat{o}_{3gn} & \mathtt{id}(x)\ \mathtt{id}(x) \end{array}, \text{ where } g \in \{\mathtt{m}, \mathtt{f}\},\ n \in \{\mathtt{s}, \mathtt{p}\}.$$

[1]Following the categorial type, we have used here the same symbol for noun phrases, regardless whether they occur in subject or object position. In the rest of the paper, we continue to split the type of a noun phrase into π and o as indicated earlier. Both are interpreted as (subsets of) individuals.

If the context permits, the set of values for the subscripts p, g, n is omitted.

Note that the same subscripts g and n appear both in the right adjoint π^r_{3gn} and in the dummy type $\hat{o}_{3gn}$. Hence, the values of the features of gender and number of the subject are identical to those of the dummy object. The anaphoric content is captured by the occurrence of two basic types, π_{3gn} and $\hat{o}_{3gn}$, which both are translated by the unary functional symbol $\mathtt{id}$. The argument-place x corresponds to π^r_{3gn}. The effect is to repeat the entity which will be substituted for x , because $\mathtt{id}(x) = x$ holds in the logic.

In simple tenses, clitics do not require agreement with the following verb. Their preverbal position makes it necessary to assign a new type to the verb. These new entries are added to the ones given in the preceding section :

$$\begin{array}{lll} \textit{examiner}: & \hat{o}^r\pi^r\boldsymbol{i} & \mathtt{examiner}(z_2, z_1) \\ \textit{examine}: & \hat{o}^r\pi^r_{3\mathtt{s}}\boldsymbol{s} & \mathtt{examiner}(z_2, z_1) \end{array}.$$

Note that the order of the variables and the non-logical symbols in the translation is not arbitrary. It is used to code the correspondence of these symbols with the simple types of the entry. In the new entries, the first variable z_1 corresponds to $\hat{o}^r$ and the second argument place z_2 to π^r respectively $\pi^r_{3\mathtt{s}}$. A simple convention will make the explicit definition of the correspondence between simple types in the entry and non-logical symbols in the translation superfluous. It suffices to count the non-basic types from left right and assign them a new variable in each occurrence. Similarly, the non-logical symbols correspond to the basic types in their order of occurrence.

Then we find the following reductions

Marie les examine	*Marie s' examine*
(MARY EXAMINES THEM)	(MARY EXAMINES HERSELF)
$(\pi_{3\mathtt{fs}})\ (o_{3g\mathtt{p}})\ (\hat{o}^r\,\pi^r_{3\mathtt{s}}\ \boldsymbol{s})\,,\qquad g = \mathtt{m}, \mathtt{f}$	$(\pi_{3\mathtt{fs}})\ (\pi^r_{3\mathtt{fs}}\ \pi_{3\mathtt{fs}}\,\hat{o}_{3\mathtt{fs}})\ (\hat{o}^r\,\pi^r_{3\mathtt{s}}\ \boldsymbol{s})\,.$

As g can take two values, the left hand display corresponds to two different type assignments, differing by $o_{3\mathtt{mp}}$ and $o_{3\mathtt{fp}}$ for the clitic *les*. The reduction itself remains unchanged, the set of links is the same for both type assignments.

Note that the semantic difference between the left and right hand sentences above is correctly captured by the reductions. The left hand reductions define the translation

$$\mathtt{examiner}(\mathtt{marie}, \mathtt{Les})\,,$$

whereas the right hand reduction gives

$$\mathtt{examiner}(\mathtt{id}(\mathtt{marie}), \mathtt{id}(\mathtt{marie}))\,.$$

As $\mathtt{id}(x) = x$, the latter translation is equivalent to

$$\mathtt{examiner}(\mathtt{marie}, \mathtt{marie})\,.$$

The type of the reflexive pronoun depends on the person to avoid non-sentences like **Tu s'examine*(YOU EXAMINE HIMSELF). Indeed, $tu : \pi_{2g\mathtt{s}},\ g = \mathtt{m}, \mathtt{f}$ and $\pi_{2g\mathtt{s}}\pi^r_{3g\mathtt{s}} \not\to 1$.

In the compound past, the clitic is separated from its verb by the auxiliary. The auxiliary does not show the relevant features by its form, but it passes them to the following word(s). The lexical entries below model this behavior by 'remembering' types.

$$
\begin{array}{lll}
\textit{examiné} & : \hat{o}^r_{\mathtt{ms}}\pi^r\boldsymbol{p} & \mathtt{examiner}(x_2, x_1) \\
\textit{examinée} & : \hat{o}^r_{\mathtt{fs}}\pi^r\boldsymbol{p} & \mathtt{examiner}(x_2, x_1) \\
\textit{examinés} & : \hat{o}^r_{\mathtt{mp}}\pi^r\boldsymbol{p} & \mathtt{examiner}(x_2, x_1) \\
\textit{examinées} & : \hat{o}^r_{\mathtt{fp}}\pi^r\boldsymbol{p} & \mathtt{examiner}(x_2, x_1) \\
\textit{avoir} & : {o_{pgn}}^r\pi^r\boldsymbol{i}\boldsymbol{p}^\ell\pi\hat{o}_{gn} & \mathtt{avoir}(y_3) \quad \mathtt{id}(y_2) \quad \mathtt{id}(y_1) \\
\textit{a} & : {o_{3gn}}^r\pi^r_{3\mathtt{s}}\boldsymbol{s}\boldsymbol{p}^\ell\pi\hat{o}_{gn} & \mathtt{avoir}(y_3) \quad \mathtt{id}(y_2) \quad \mathtt{id}(y_1) \\
\textit{être} & : \hat{o}^r_{pgn}\pi^r\boldsymbol{i}\boldsymbol{p}^\ell\pi\hat{o}_{gn} & \mathtt{être}(y_3) \quad \mathtt{id}(y_2) \quad \mathtt{id}(y_1) \\
\textit{est} & : \hat{o}^r_{3g\mathtt{s}}\pi^r_{3\mathtt{s}}\boldsymbol{s}\boldsymbol{p}^\ell\pi\hat{o}_{g\mathtt{s}} & \mathtt{être}(y_3) \quad \mathtt{id}(y_2) \quad \mathtt{id}(y_1) \quad .
\end{array}
$$

Here too, we have followed our convention that the variables respectively non-logical symbols correspond to the non-basic types respectively basic types in their order of occurrence. For example, consider the last four entries above. The variables y_1 and y_2 correspond to the right adjoints o^r and π^r , with the appropriate subscripts, with or without hat. The left adjoint $\boldsymbol{p}^\ell$ corresponds to the variable y_3. The basic type $\boldsymbol{i}$ respectively $\boldsymbol{s}$ is translated by the relational symbol, the basic types π and $\hat{o}_{gn}$ are translated by the functional symbol id. Choosing the value $g = \mathtt{f}$ in the type $o_{3\mathtt{fp}}$ for *les*, the sentence *Marie les a examinées* is recognized by the following reduction

$$
\begin{array}{c}
\textit{Marie} \quad \textit{les} \qquad\qquad \textit{a} \qquad\qquad \textit{examinées} \\
(\pi_{3\mathtt{fs}})\ (o_{3\mathtt{fp}})\ (o^r_{3\mathtt{fp}}\pi^r_{3\mathtt{s}}\ \boldsymbol{s}\ \boldsymbol{p}^\ell\ \pi\ \hat{o}_{\mathtt{fp}})\ (\hat{o}^r_{\mathtt{fp}}\pi^r\ \boldsymbol{p})\ .
\end{array}
$$

If the clitic is a reflexive pronoun, the auxiliary in the compound tense is *être*. The past participle agrees in gender and number with the clitic if the latter is the direct object. Hence the type of *être* is similar to that of *avoir*, except that it is tailored to the reflexive pronoun, and therefore uses the dummy object types.

$$
\begin{array}{c}
\textit{Marie} \qquad\qquad \textit{s'} \qquad\qquad \textit{est} \qquad\qquad \textit{examinée} \\
(\pi_{3\mathtt{fs}})\ (\pi^r_{3\mathtt{fs}}\ \pi_{3\mathtt{fs}}\ \hat{o}_{3\mathtt{fs}})\ (\hat{o}^r_{3\mathtt{fs}}\pi^r_{3\mathtt{s}}\ \boldsymbol{s}\ \boldsymbol{p}^\ell\ \pi\ \hat{o}_{\mathtt{fs}})\ (\hat{o}^r_{\mathtt{fs}}\pi^r\ \boldsymbol{p})\ .
\end{array}
$$

Note that the hat on the direct object in the type of *est* prevents **Marie l'est examinée* and **Marie s'a examiné* as $o_{3\mathtt{fs}} \not\to \hat{o}_{3\mathtt{fs}}$ and $\hat{o}_{3\mathtt{fs}} \not\to o_{3\mathtt{fs}}$.

The translation of the latter sentence is

$$\mathtt{être}(\mathtt{examiner}(\mathtt{marie}, \mathtt{marie}))\ .$$

Whereas the auxiliaries *avoir* and *être* 'remember' the features of the object, the modal verbs 'remember' the features of the subject. The clitic is positioned between the modal verb and the verb of which it is the complement.

$$
\begin{array}{lll}
\textit{devoir} & : \pi^r_{pgn}\boldsymbol{i}\boldsymbol{i}^\ell\pi_{pgn} & \mathtt{devoir}(y)\ \mathtt{id}(x) \\
\textit{doit} & : \pi^r_{3g\mathtt{s}}\boldsymbol{s}\boldsymbol{i}^\ell\pi_{3g\mathtt{s}} & \mathtt{devoir}(y)\ \mathtt{id}(x)
\end{array}
$$

where $p = 1, 2, 3;\ g = \mathtt{m}, \mathtt{f};\ n = \mathtt{s}, \mathtt{p}$. In these entries, the unary relational symbol devoir translates the basic types $\boldsymbol{i}$ and $\boldsymbol{s}$. The unary functional symbol id translates the basic type π_{pgn}. The variable y corresponds to $\boldsymbol{i}^\ell$ and x to π^r_{pgn}. The reason why the type of the modal verbs depends on the gender becomes evident when they are used in combination with the compound past. For example

$$
\begin{array}{c}
\textit{Marie} \qquad \textit{doit} \qquad\qquad \textit{s'} \qquad\qquad \textit{être} \qquad\qquad \textit{examinée} \\
(\pi_{3\mathtt{fs}})(\pi^r_{3\mathtt{fs}}\ \boldsymbol{s}\ \boldsymbol{i}^\ell\ \pi_{3\mathtt{fs}})(\pi^r_{3\mathtt{fs}}\ \pi_{3\mathtt{fs}}\ \hat{o}_{3\mathtt{fs}})(\hat{o}^r_{3\mathtt{fs}}\ \pi^r\ \boldsymbol{i}\ \boldsymbol{p}^\ell\ \pi\ \hat{o}_{\mathtt{fs}})(\hat{o}^r_{\mathtt{fs}}\ \pi^r\ \boldsymbol{p})\ .
\end{array}
$$

The reader may verify that the translation renders the correct meaning and check that the non-sentences **Marie doit s'être examiné*, **Marie doit s'être examinés* and **Marie doit s'être examinées* have no reduction to the sentence type.

3 Conclusion

Pregroup dictionaries use more basic types and entries per word than categorial grammars. This is the price to pay for reducing computation to a single rule, namely generalized contraction. In spite of this 'explosion' of types, dictionaries like the one presented here have parsing algorithms which are linear when given strings of lexical entries, see (Preller 2007). Exploiting certain regularities of features rendering 'lazy type assignment' possible, this result is improved in forthcoming work by (Preller-Prince) : there is a linear algorithm which for a given string of words finds a parsing, i . e. a reduction to the sentence type, if and only if the string of words is a sentence.

Références

Joachim Lambek (1999), Type Grammar revisited, in : Alain Lecomte et al., editors, Logical Aspects of Computational Linguistics, Springer LNAI 1582, pp.1-27.

Joachim Lambek (2004), A computational algebraic approach to English grammar, Syntax 7 :2, pp. 128-147.

Michael Moortgat (2004), Richard Oehrle, Pregroups and Type-Logical Grammar : Searching for Convergence, Language and Grammar Studies in Mathematical Linguistics and Natural Language, P. Scott, C. Casadio, R.A.Seeley (eds.), CSLI Publications, 2004

Anne Preller (2007). Toward Discourse Representation Via Pregroup Grammars, *Journal of Language Logic and Information*, Springer Netherlands, vol. 16, no. 2, pp. 173-194, published on-line 02.02.2007, DOI 10.1007/s10849-0006-9033-y.

Anne Preller (2007). Linear Processing with Pregroup Grammars, Studia Logica, forthcoming.

Anne Preller, Violaine Prince (2006), Pregroup grammars with linear parsing : long distance dependency of clitics in French, rapport de recherche LIRMM N°00137778.

Johan van Benthem (2005), Guards, Bounds and Generalized Semantics. *Journal of Language, Logic and Information*, 14, 263-279.

Remerciements

The author would like to thank Violaine Prince for her insistence to tackle anaphora with pregroup grammars.