Auxiliaries and Clitics in French UCG Grammar
Karine Kray-Baschung, Gabriel G. Bès, Annick Corluy, Thierry Guillotin

To cite this version:

HAL Id: halshs-00372903
https://halshs.archives-ouvertes.fr/halshs-00372903
Submitted on 7 Aug 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
AUXILIARIES AND CLITICS IN FRENCH
UCG GRAMMAR


** Universite de Clermont II, 34 Avenue Carnot, F-63037 Clermont-Ferrand (France)
*** Laboratoires de Marcoussis, Route de Nozay, F-91460 Marcoussis

ABSTRACT

French auxilliaries and clitics have been analysed in the flame of U.C.G. (Unification Categorial Grammar). Concatenation of a functor sign and an adjacent argument sign is the basic operation of the model ; unification allows (a) to verify if constraints on concatenation are respected ; (b) to produce a flow of information between the functor sign and the argument sign.

The rules of the grammar and the design structure of the sign allows to express : (a) the concatenation between French auxilliaries (etre and avoir) and the participle verb form within a single pattern, (b) transitions between clitics in a systematic way. Two complex questions of French syntax are thus covered in a fairly simple way.

The UCG Model

U(nification). C(ategorial) G(rammar) is a new grammatical model proposed by an Edinburgh team headed by Ewan Klein in [CALDER 86] and [ZEEVAT 86]. UCG is a feature grammar incorporating some basic insights from GPSG [GAZDAR 85] and HPSG [POLLARD 84]. Functional application applies in UCG as in categorial grammars; it allows for concatenation of a functor with an adjacent argument. Unification is a basic operation which allows (a) to verify if constraints on concatenation are respected; (b) to produce a flow of information between functor and argument. This information together with some defined aspects of the information carried by the functor, will be finally inscribed in the resulting concatenated sign.

The Sign

A UCG sign has the following format :

(1)

\[
\text{sign} \rightarrow \text{category:semantics:order:phonology} \\
\text{category} \rightarrow \text{head:features:catlist} \\
\text{features} \rightarrow [\text{feat, clo, agree, class}] \\
\text{catlist} \rightarrow \text{nil} \\
\text{catlist} \rightarrow \text{catlist/active} \\
\text{active} \rightarrow \text{sign} \\
\text{semantics} \rightarrow \text{index} \\
\text{semantics} \rightarrow \text{index: predicate: arglist} \\
\text{arglist} \rightarrow \text{arg1, ..., argn}
\]

The work reported here was carried out as part of ESPRIT Project 393 (ACORD), "The construction and interrogation of Knowledge Bases using Natural Language Text and Graphics"
sent\textsuperscript{\[fin,v,(,:sg:p3),\]}:nil\textsuperscript{\[nom,n,(:,:sg:p3),\]}:nil\textsuperscript{\[X,pre:\]}:
and\textsuperscript{\[e,at(e,now),aimer(e,X,Y)\]}:
:o:
:aime

(b) Marie

Head\textsuperscript{\[Feat,Clo,Ag,Class\]}:
C/ (Head\textsuperscript{\[Feat,\_Ag,Class\]}:
C/(np\textsuperscript{\[or(nom,acc),Clo,\(\text{fem}:sg:p3),\]ex}\n
:Sem
:Or
:)
:Sem
:o
:marie

Categories

Categories are defined by

(3) (a) A simple category is a category.
(b) If H:C is a category and si is a sign, H:(C/Si)
is a category.

Rules

\cite{ZEEVAT 86} describes 2 grammar rules based on
functional application.

(4) FA (Forward Application)

Functor:

HF
:CF/(HA:CA:SA:pre:\_)
:SF
:OF
:W1

Argument:

HA:CA:SA:pre:W2

\rightarrow HF:CF:SF:OF:[W1,W2]

(5) BA (Backward Application)

Argument:

HA:CA:SA:post:W1

Functor:

HF
:CF/(HA:CA:SA:post:\_)
:SF
:OF
:W2

\rightarrow HF:CF:SF:OF:[W1,W2]

We added two rules to these, inspired by
functional composition as described in \cite{STEEEDMAN 86}.

(6) FC (Forward Composition)

Functor:

HF
:CF/(HA:CA:SA:pre:\_)
:SF
:OF
:W1

Argument:

HA:CA/(np\textsuperscript{\[Fe,\_Ag,\]lex}\textsuperscript{\[-nil:X:\_]})\textsuperscript{\[-:pre:\_]}SA:pre:W2

\rightarrow HF
:CF/(np\textsuperscript{\[Fe,\_Ag,\]lex}\textsuperscript{\[-nil:X:\_]})\textsuperscript{\[-:pre:\_]}
:SF
:OF
:W1,W2

FC is basically designed to deal with np-gaps.

(7) BC (Backward Composition)

Argument:

HA:CA/(np\textsuperscript{\[Feats:nil:X:O:\_\]})\textsuperscript{\[-:post:\_]}SA:post:W1

Functor:

HF
:CF/(HA:CA:SA:post:\_)
:SF
:SO
:W2

\rightarrow HF
:CF/(np\textsuperscript{\[Feats:nil:X:O:\_]})
:SF
:OF
:W1,W2

BC is designed to deal with free-order of np-
arguments of verbs

Forward application must be interpreted as
follows:

\footnotesize

\begin{itemize}
\item (1) where PROLOG conventions are respected: lower
\item case = constant, upper case = variable, _ = anonymous
\item variable
\end{itemize}
If a sign of string $W_1$ and category $HF:CF/(HA:CA)$ unifies with a sign of string $W_2$ and category $HA:CA$, $W_1$ concatenates with $W_2$; the resulting sign, with string $[W_1,W_2]$, is of category $HF:CF$, where $HF:CF$ is the category inherited from the functor as resulting from unification with its argument, and stripping $HA:CA$.

Mutatis mutandis, analogous interpretations must be given to (5) through (7).

By definition (3) $HA:CA$ in $HF:CF/(HA:CA)$ of (4) must be a sign; it is the active part of the functor. The final concatenated sign is obtained by stripping the active part of the functor as instantiated by the argument.

Example

For example: (8) is the instantiation by $BA$ of (2b) as the functor with respect to (2a) as the argument of the rule; (9) is the resulting sign, obtained from (8) by stripping; (10) represents the sign of the whole sentence *Pierre aime Marie*:

(8) *Marie*

\[
\text{sent}^*\left[\text{fin},\text{m},(\_:_sg:p3),\text{Class}\right] \\
\text{:nil/np}^*\left[\text{nom},\text{n},(\text{Gen1 :sg:p3}),\text{Class1}\right] \text{:nil/X:pre:P1} \\
\left/\text{sent}^*\left[\text{fin},(\_:_sg:p3),\text{Class}\right] \\
\text{:nil/np}^*\left[\text{nom},\text{n},(\text{Gen1 :sg:p3}),\text{Class1}\right] \text{:nil/X:pre:P1} \\
\left/\text{np}^*\left[\text{acc},\text{m},(\text{fem :sg:p3}),\text{lex}\right] \text{:nil:marie:post:}_1 \\
\left/\text{and}(e,\text{at}(e,\text{now}),\text{aimer}(e,X,\text{marie})) \\
\left/\text{post} \\
\left/\text{aimer} \\
\left/\text{and}(e,\text{at}(e,\text{now}),\text{aimer}(e,X,\text{marie})) \\
\left/\text{O} \\
\left/\text{marie}
\]

(9) *aime Marie*

\[
\text{sent}^*\left[\text{fin},\text{m},(\_:_sg:p3),\text{Class}\right] \\
\text{:nil/np}^*\left[\text{nom},\text{n},(\text{Gen1 :sg:p3}),\text{Class1}\right] \text{:nil/X:pre:P1} \\
\left/\text{and}(e,\text{at}(e,\text{now}),\text{aimer}(e,X,\text{marie})) \\
\left/\text{O} \\
\left/{\text{aimer,marie}}
\]

(10) *Pierre aime Marie*

\[
\text{sent}^*\left[\text{fin},\text{n},(\_:_sg:p3),\text{Class}\right] \\
\text{:nil} \\
\left/\text{and}(e,\text{at}(e,\text{now}),\text{aimer}(e,\text{pierre},\text{marie})) \\
\left/\text{O} \\
\left/\text{pierre,aimer,marie}
\]

Semantics

The semantics of UCG incorporates the basic insights of Kamp's DRT [KAMP 81] but the introduction of indexes greatly increases the expressive power of semantic representations (cf. [ZEEVAT 86]).

To resume:

- The whole model is based on:
  - one unique operation: concatenation between adjacent constituents.
  - one unique process to control the flow of information and to verify conditions: unification.

- similar ways to combine a functor and its argument to give a resulting sign.

The French sentence

**simple verbs**

They accept left-placed arguments (as clitics) and right-placed ones (as lexical ones).

**composed verbal forms**

No argument can be inserted between the auxiliary and the participle form.

Whereas in English only one auxiliary is used to construct perfect tenses, French uses *avoir* and *être* depending on the main verb. Furthermore, *être* is also used for passive constructions.

The most important problem, however, is due to the agreement of the past participle with the subject of the main verb when used with *être*, but with the object only if it precedes the auxiliary-when used with *avoir*.

However, we succeeded to maintain a single lexical entry for a verb, allowing for the different order of arguments. This is made possible by the introduction of forward and backward composition rules.

**AUXILIARIES**

The following are the main features allowing a correct treatment of auxiliaries in a French UCG grammar.

Features as presented in Figure 1 column (ii): PSPA for past participles of verbs using *avoir* as auxiliary, PSPE for verbs used with *être*, PAS for passive participle. They allow for the distinction between finite and non-finite forms and between participles used with *avoir* or *être*.

Values for the CL label: *v* value denotes the fact that the verb is "virgin" i.e. has not consumed any of its arguments.

Values for GE, NB, PERS allow for correct agreement of the past participle and between auxiliary and subject.

A unique format for perfect tenses with *avoir* and *être* and for passive constructs with *être* was designed as follows:

(12) auxiliary general design

\[
\text{sent}^*\left[\text{fin},\text{v},\text{Ag,Class}\right] \\
\left/\text{C/sent}^*\left[\text{FEAT},\text{v,Ag,Class}\right]\right);\text{:Sem:pre:}_1 \\
\left/\text{Sem} \\
\left/\text{O} \\
\left/\text{STRING}
\]

where STRING and FEAT can take values *avoir* and *pspa* or *être* and *pspe* or *pas*; the agreement of the auxiliary unifying with the agreement of the participle will insert the correct agreement on the nominative argument in the participle and thus will control the agreement of the subject with the auxiliary-participle unit.

One of the main achievements of our French UCG grammar is to have a single lexical entry for a verb, notwithstanding differences in semantics according to
tense, free word order, and constrained word order due to critics.

Standard lexical entries present word order as for non-clitic arguments, and semantics as for the infinitive.

A morphological component allows for a dynamic transformation of these entries according to tense gender and person.

Thus, typical entries look like:

(13) regarder

\[
\text{sent}[^{\text{fn}},v,[\text{Ag},_\text{v}]]
\]
\[
^{\text{nil/np}}[^{\text{nom,n}},[\text{Ag},_\text{v}]]^{\text{nil/X}}{:\text{nil:pre:}}
\]
\[
^{\text{np}}[^{\text{acc,m}},_\text{v}]{:}\text{:nil:Y}^{:\text{post:}}
\]
\[
^{\text{regarder}}(e,X,Y)
\]
\[
^{:\text{O}}
\]

When analysing (morphologycally) the passive participle (13) is transformed (by a special passive lexical rule) into:

(14) regardée

\[
\text{sent}[^{\text{pas}},v,[\text{(fem:sg:Pe)},_\text{v}]]
\]
\[
^{\text{nil/np}}[^{\text{nom,n}},[\text{(fem:sg:Pe)},_\text{v}]]^{\text{nil:Y}}{:\text{nil:pre:}}
\]
\[
^{\text{np}}[^{\text{par,m}},_\text{v}]{:}\text{:nil:X}^{:\text{post:}}
\]
\[
^{\text{regarder}}(e,X,Y)
\]
\[
^{:\text{O}}
\]

(15) être

\[
\text{sent}[^{\text{fn}},v,[\text{(Ge:sg:p3)},_\text{v}]]
\]
\[
^{\text{nil/np}}[^{\text{nom,n}},[\text{(Ge:sg:p3)},_\text{v}]]^{\text{nil:X}}{:\text{nil:pre:}}
\]
\[
^{\text{np}}[^{\text{par,m}},_\text{v}]{:}\text{:nil:Y}^{:\text{post:}}
\]
\[
^{\text{est}}
\]
\[
^{:\text{O}}
\]

(16) est regardée

\[
\text{sent}[^{\text{fn}},v,[\text{(fem:sg:p3)},_\text{v}]]
\]
\[
^{\text{nil/np}}[^{\text{nom,n}},[\text{(fem:sg:p3)},_\text{v}]]^{\text{nil:X}}{:\text{nil:pre:}}
\]
\[
^{\text{np}}[^{\text{par,m}},_\text{v}]{:}\text{:nil:Y}^{:\text{post:}}
\]
\[
^{\text{est}}^{:\text{regardée}}
\]

This can then correctly be combined with the subject Marie (2b) respecting the agreement auxiliary-subject and subject-participle (because it is used with être)

CLITICS

Beside the fact that clitics in French are always placed before the verb or verb-auxiliary unit (as it was said before) there are also restrictions concerning placement between them.

It is thus necessary to specify (17 a) and to exclude (17 b), among others.

The main problem with French clitics is that arguments combine in a different order with the verb according to (a) whether they are clitic or not and (b) whether they are first/second person or third person.

(a) Marie lui\[\text{[dat]}\] a donné un livre\[\text{[acc]}\]

(b) Marie a lui\[\text{[dat]}\] donné un livre\[\text{[acc]}\]

The core of conditions on clitic ordering in French can be found in (19). These transitions are valid for argumental clitics and non-argumental ones (for example, VP modifiers, as y in Il y a apporté un livre), but the present paper is only intended to cover the argumental ones.

(19) ^

The complex information of the matrice are included in a uniform way in the clitics lexical entries.

The basic template for clitic is:

Head\[\text{[Feat, Clo2,A,Class]}\]

Head\[\text{[Feat, Clo1,A,Class]}\]

\[
^{:\text{C}}^{:\text{O}}^{:\text{S}}
\]

where the relation between Clo2 and Clo1 contains the matrice information relevant for each clitic.

IMPLEMENTATION

The UCG French grammar has been implemented at the Laboratoires de Marcoussis (France) on a VAX 780 in C-PROLOG using PIMPLE, a PROLOG implementation of a PATR-II like tool for development of unification grammars, implemented by the Centre for Cognitive Science of Edinburgh University.

Some more examples with auxiliaries and clitics

Entries for the sentence Marie la lui a donné :
Entries for the sentence "Marie lui est donnée":

(24) est

sent'[fin,v,(fem:sg:Pe),_]
:/np[acc,n,(fem:sg:Pe),_]:nil:Y:pre:_
:donner(e,unknown,Z,Y)

:O

[est,donnée]  

(25) donnée

sent'[pas,v,(fem:sg:Pe),_]
:/np[acc,n,(fem:sg:Pe),_]:nil:Y:pre:_
:/np[dat,m,_,_]:nil:Z:post
:donner(e,unknown,Z,Y)

:O

:donnée

est with donnée by FA yielding:

sent'[fin,v,(fem:sg:Pe),_]
:/np[acc,n,(fem:sg:Pe),_]:nil:Y:pre:_
:/np[dat,m,_,_]:nil:Z:post
:donner(e,unknown,Z,Y)

:O

[est,donnée]  

lui with [est,donnée] by FA yielding:

sent'[fin,v,(fem:sg:Pe),_]
:/np[acc,n,(fem:sg:Pe),_]:nil:Y:pre:_
:donner(e,unknown,Z,Y)

:O

[lui,[est,donnée]]

are combined in the following way:

a with donnée by FA yielding:

sent'[fin,v,(_:sg:p3),_]
:nil/np[nom,n,(_:sg:p3),_]:nil:X:pre:_
:/np[acc,n,(_:sg:p3),_]:nil:Y:pre:_
:/np[dat,m,_,_]:nil:Z:post
:donner(e,X,Z,Y)

:O

[a,donnée]  

lui with [a,donnée] by FA yielding:

sent'[fin,prota,(_:sg:p3),_]
:nil/np[nom,n,(_:sg:p3),_]:nil:X:pre:_
:/np[acc,n,(_:sg:p3),_]:nil:Y:pre:_
:donner(e,X,pro(Z),Y)

:O

[lui,[a,donnée]]

lui with [lui,[a,donnée]] by FA yielding:

sent'[fin,prota,(_:sg:p3),_]
:nil/np[nom,n,(_:sg:p3),_]:nil:X:pre:_
:donner(e,X,pro(Z),pro(Y))

:O

[lui,[lui,[a,donnée]]]

marie with [la,[lui,[a,donnée]]] by FA yielding:

sent'[fin,n,(_:sg:p3),_]
:nil
:donner(e,marie,pro(Z),pro(Y))

:O

[marie,[la,[lui,[a,donnée]]]]

"a donner as modified by morphological rules into a past participle:"

(23) donnée

sent'[pspa,v,(_:sg:p3),_]
:C/sent'[or(pas,pspe),v,Ag,Class]:C:Sem:pre:_
:Sem

:O

:a

"donner" as modified by morphological rules into a past participle:

(22) a

sent'[fin,v,(_:sg:p3),_]
:C/sent'[pspa,v,(_:sg:p3),_]:C:Sem:pre:_
:Sem

:O

"a"
Marie with (lui, est donnée) by FA yielding:

```
sent:[fin,v,(fem:sg:Pe),_]
:nil
:donner(e,unknown,pro(Z),marie)
:O
:[Marie,[lui,[est,donnée]]]
```

REFERENCES

[CALDER 86]

[GAZDAR 85]

[KAMP 81]

[POLLARD 84]

[STEEDMAN 86]

[ZEEVAT 86]