AUXILIARIES AND CLITICS IN FRENCH UCG GRAMMAR

by K. BASCHUNG [**], G.G BES [**], A. CORLUY [***], T. GUILLOTIN [***]
[**] 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 frame 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)

```
sign --> category:semantics:order:phonology
category --> head^features:catlist
        features --> [feat,clo,agree,class]
catlist --> nil
catlist --> catlist/active
        active --> sign
semantics --> index
semantics --> index:predicate:arglist
        arglist --> agr1, ... ,argn
order --> pre | post
phonology --> <lexical_item>
```

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"

In graph notation, a UCG sign can be represented (in a slightly simplified form, relevant to this paper) as in figure 1.

In this figure the leaves of vertical branches columns (i) through (viii) - denote the values of the corresponding labels in its upper portion .

We have :
(i) to (vi) Simple categories : sent^_:nil, noun^_:nil.
(ii) Features on (i) (see below)
(iii) values for the CL(itics) label are : *prod* (dialogue pronouns, for *me* , *te*, *nous*, *vous*); *protob* (third person object pronouns : *le*, *la*, *les*); *prota* (third person dative pronouns : *lui*, *leur*) ; *se*, *en* and *y*, (for *se*, *en* and *y* pronouns respectively) ; n is a barrier symbol (see below).
(iv and v) values for morphological aspects of the sign:
(v) categorizes signs in lex(ical) and pron(ominal) ones,
(iv) in masc(ulin) and fem(inin), in sing(ular) and pl(ural), and introduces values for the 3 persons.
(vi) the subcatlist : the label C will denote the typical variable for it.
(vii) the index sort system (which is not exhibited here) allows selection on semantic features while a special field (pr_res) contains information (agreement, class and COR) for the pronoun resolution component.
COR(eference) is intended to prepare the semantic representation for pronoun resolution. The corresponding values are : obl(igatory), for bound anaphora as in *se* ; ind(ependent), for NP nominals of indicative sentences and dep(endent) for NP nominals of subjunctive ones (the algorithm for pronoun resolution will not be presented here, but the semantic representation specified by the proposed grammar is intended to carry all the required relevant information).
(viii) *post* and *pre* are the values for *order* ; they are essential for handling word order and for the application of the grammar rules.

In the unification process and in the generation of the subsequent flow of information, the labels Class, Ge, Nb and Pe denote variables for the corresponding values, Clo the variable for clitic placement value and O, the variable for order values.

The two following are the French UCG signs for *aime* and *Marie* :

(2) (a) [1] *aime*

```
sent^[fin,v,(_:sg:p3),_]
:nil/np^[nom,n,(_:sg:p3),_]:nil:X:pre:_
   /np^[acc,m,_,_]:nil:Y:post:_
:and(e,at(e,now),aimer(e,X,Y))
:O
:aime
```

(b) *Marie*

```
Head^[Feat,Clo,Ag,Class]
:C/ (Head^[Feat,_,Ag,Class]
    :C/(np^[or(nom,acc),Clo,(fem:sg:p3),lex]
         :nil:marie:Ord:_)
    :Sem
    :Ord
    :_)
:Sem
:O
:marie
```

Categories

Categories are defined by

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

Rules

[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
```

```
-> HF:CF:SF:OF:[W1,W2]
```

[1] where PROLOG conventions are respected : lower case = constant, upper case = variable, _ = anonymous variable

(5) BA (Backward Application)

Argument :

```
HA:CA:SA:post:W1
```

Functor :

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

```
-> HF:CF:SF:OF:[W1,W2]
```

We added two rules to these, inspired by functional composition as described in [STEEDMAN 86].

(6) FC (Forward Composition)

Functor :

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

Argument :

```
HA:CA/(np^[Fe,_,Ag,¯lex]:nil:X:_:_):SA:pre:W2
```

```
->  HF
    :CF/(np^[Fe,n,Ag,¯lex]:nil:X:pre:_)
    :SF
    :OF
    :[W1,W2]
```

FC is basically designed to deal with np-gaps.

(7) BC (Backward Composition)

Argument :

```
HA:CA/(np^Feats:nil:X:O:_):SA:post:W1
```

Functor :

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

```
->  HF
    :CF/(np^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 :

If a sign of string W1 and category HF:CF/(HA:CA) unifies with a sign of string W2 and category HA:CA, W1 concatenates with W2; the resulting sign, with string [W1,W2], 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.

Mutadis 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*

```
sent^[fin,m,(_:sg:p3),Class]
:nil/np^[nom,n,(Ge1:sg:p3),Class1]:nil:X:pre:P1
   /(sent^[fin,_,(_:sg:p3),Class]
      :nil/np^[nom,n,(Ge1:sg:p3),Class1]:nil:X:pre:P1
         /np^[acc,m,(fem:sg:p3),lex]:nil:marie:post:_
      :and(e,at(e,now),aimer(e,X,marie))
      :post
      :aime)
:and(e,at(e,now),aimer(e,X,marie))
:O
:marie
```

(9) *aime Marie*

```
sent^[fin,m,(_:sg:p3),Class]
:nil/np^[nom,n,(Ge1:sg:p3),_]:nil:X:pre:P1
:and(e,at(e,now),aimer(e,X,marie))
:O
:[aime,marie]
```

(10) *Pierre aime Marie*

```
sent^[fin,n,(_:sg:p3),Class]
:nil
:and(e,at(e,now),aimer(e,pierre,marie))
:O
:[pierre,[aime,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 constituants.
* 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 rigth-placed ones (as lexical ones).

composed verbal forms

No argument can be inserted between the auxilliary 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) auxilliary general design

```
sent^[fin,v,Ag,Class]
:C/sent^[FEAT,v,Ag,Class]):C:Sem:pre:_
:Sem
:O
:STRING
```

where STRING and FEAT can take values *avoir* and *pspa* or *être* and *pspe or pas*; the agreement of the auxilliary 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, nonwithstanding differences in semantics according to tense, free word order, and constrained word order due to clitics.

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*

```
sent^[fin,v,Ag,_]
:nil/np^[nom,n,Ag,_]:nil:X:pre:_
   /np^[acc,m,_,_]:nil:Y:post:_
:regarder(e,X,Y)
:O
:regarder
```

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

(14) *regardée*

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

to be combined with an auxiliary as

(15) *être*

```
sent^[fin,v,(Ge:sg:p3),_]
:C/sent^[or(pspe,pas),v,(Ge:sg:p3),_]:C:S:pre:_
:S
:O
:est
```

yielding

(16) *est regardée*

```
sent^[fin,v,(fem:sg:p3),_]
:nil/np^[nom,n,(fem:sg:p3),_]:nil:Y:pre:_
   /np^[par,m,_,_]:nil:X:post:_
:regarder(e,X,Y)]
:O
:[est,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-auxilliary 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.

(17) (a) Marie lui$_{[dat]}$ a donné un livre$_{[acc]}$
(b) Marie a lui$_{[dat]}$ donné un livre$_{[acc]}$

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.

(18) (a) Marie donne un livre$_{[acc]}$ à Pierre$_{[dat]}$
(b) Marie lui$_{[dat]}$ donne un livre$_{[acc]}$
(c) Marie le$_{[acc]}$ lui$_{[dat]}$ donne
(d) Marie me$_{[dat]}$ le$_{[acc]}$ donne

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) [2]

-->	prod[acc]	prod[dat]	protob	prota	y	en[acc]	en[de]	se[acc]	se[dat]
prod[acc]	Ø	*	Ø	*	G	Ø	G	Ø	*
prod[dat]	*	Ø	G	Ø	Ø	G	G	*	Ø
protob	Ø	*	Ø	G	G	Ø	G	Ø	*
prota	*	Ø	*	Ø	Ø	G	G	*	Ø
y	*	Ø	*	Ø	Ø	G	*	*	*
en[acc]	Ø	*	Ø	*	*	Ø	*	Ø	*
en[de]	*	*	*	*	*	*	Ø	*	*
se[acc]	Ø	*	Ø	*	G	Ø	G	Ø	*
se[dat]	*	Ø	G	Ø	Ø	G	G	*	Ø

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

The basic template for clitic is :

```
Head^[Feat,Clo2,A,Class]
:C / (Head^[Feat,Clo1,A,Class]
     :C/np^[_,_,_,pro]:nil:pro(X):_:_
     :S
     :pre
     :_)
:S
:O
:String
```

where the relation between Clo2 and Clo1 constains 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 donnée* :

[2] where G = grammatical, * = non grammatical, Ø = impossible (because an argument of a verb cannot be consumed twice)

(20) *la*

```
Head^[Feat,protob,A,Class]
:C / (Head^[Feat,or(prota,y,m,v),A,Class]
      :C/np^[acc,_,(fem:sg:p3),pro]:nil:pro(X):_:_
      :S
      :pre
      :_)
:S
:O
:la
```

(21) *lui*

```
Head^[Feat,prota,A,Class]
:C / (Head^[Feat,or(en,m,v),A,Class]
      :C/np^[dat,_,(_:sg:p3),pro]:nil:pro(X):_:_
      :S
      :pre
      :_)
:S
:O
:l
```

(22) *a*

```
sent^[fin,v,(_:sg:p3),Class]
:C/sent^[pspa,v,(_:sg:p3),Class]:C:Sem:pre:_
:Sem
:O
:a
```

donner as modified by morphological rules into a past participle :

(23) *donnée*

```
sent^[pspa,v,Ag,_]
:nil/np^[nom,n,Ag,_]:nil:X:pre:_
    /np^[acc,n,(fem:sg:_),pro]:nil:Y:pre:_
    /np^[dat,m,_,_]:nil:Z:post
:donner(e,X,Z,Y)
:O
: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,(fem:sg:_),pro]: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,(fem:sg:_),pro]:nil:Y:pre:_
:donner(e,X,pro(Z),Y)]
:O
:[lui,[a,donnée]]
```

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

```
sent^[fin,protob,(_:sg:p3),_]
:nil/np^[nom,n,(_:sg:p3),_]:nil:X:pre:_
:donner(e,X,pro(Z),pro(Y))
:O
:[la,[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]]]]
```

Entries for the sentence *Marie lui est donnée* :

(24) *est*

```
sent^[fin,v,Ag,Class]
:C/sent^[or(pas,pspe),v,Ag,Class]:C:Sem:pre:_
:Sem
:O
:est
```

(25) *donnée*

```
sent^[pas,v,(fem:sg:Pe),_]
:nil/np^[nom,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),_]
:nil/np^[nom,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),_]
:nil/np^[nom,n,(fem:sg:Pe),_]:nil:Y:pre:_
:donner(e,unknown,pro(Z),Y)
:O
:[lui,[est,donnée]]
```

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]
CALDER, KLEIN, MOENS, ZEEVAT, *Problems of Dialogue Parsing*, ACORD deliverable T2.1, Edinburgh, 1986.

[GAZDAR 85]
GAZDAR, KLEIN, PULLUM, SAG, *Generalized Phrase Structure Grammar*, London, Basil Blackwell, 1985.

[KAMP 81]
KAMP, A theory of Truth and Semantic Representation. In Groenendijk, Janssen and Stokhof (eds) *Formal Methods in the Study of Language*, Volume 136, pp277-322. Amsterdam, Mathematical Centre Tracts.

[POLLARD 84]
PROUDIAN and POLLARD, Parsing Head-driven Phrase Structure Grammar. In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, University of Chicago, Chicago, Illinois, 8-12 July, 1985, pp167-171.

[STEEDMAN 86]
STEEDMAN, *Incremental Interpretation in Dialogue*, ACORD deliverable T2.4, Edinburgh, 1986.

[ZEEVAT 86]
ZEEVAT, KLEIN, CALDER, *Unification Categorial Grammar*, Edinburgh, 1986.