

Treating the Semantics of French Clitics with Minimalist Grammars

Maxime Amblard

January 12, 2006

Logic of variation, Utrecht

Signes,
University of Bordeaux1 & LaBRI-C.N.R.S.
INRIA-Futurs, Bordeaux

<http://www.labri.fr/~amblard>

Contents

1	Minimalist Grammars - MG	4
2	The lexicon as automaton	9
3	French clitics - Stabler's version	11
3.1	the two different parts of clitics interaction in the analysis	11
3.2	Stabler's version	12
4	extension for French clitics	13
4.1	clitics after	13
4.2	Subject pronoun and negation	15
4.2.1	negation	15
4.2.2	Subject pronoun	16
5	exemple of derivation	17
5.1	Lexical representation	17
5.2	analysies of a sentence	18
6	A semantics for minimalist grammars	22
7	semantic of french clitics	25

1. Minimalist Grammars - MG

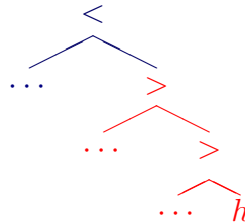
We use the standard definition of Minimalist Grammars give by Ed Stabler since 1997.

MG is a lexicalised formalism.

- representation structure
- features
- lexicon
- generative rules

Representation structure :

- binary tree
- leaf: sequence of features.
- internal node "<" or ">" which indicates the path to the head.
- maximal projection : largest tree whose head is h.



Features

- Base is a set of basic features

$$\{d, n, v, \dots\}$$

- Selector features :

$$\{=x \mid x \in Base\}$$

- Move Features is a subset of Base features: $MF \subset Base$.

- Licensees is a set of licensee feature :

$$\{-x \mid x \in MF\}$$

- Licensors is a set of licensor feature :

$$\{+x \mid x \in MF\}$$

Lexicon : every entry follows the schema :

sequence of features /Phonological Form/ (Logical Form)

Generative rules

There are two main operations :

- merge - a kind of concatenation of two trees.

$$Tree \times Tree \rightarrow Tree$$

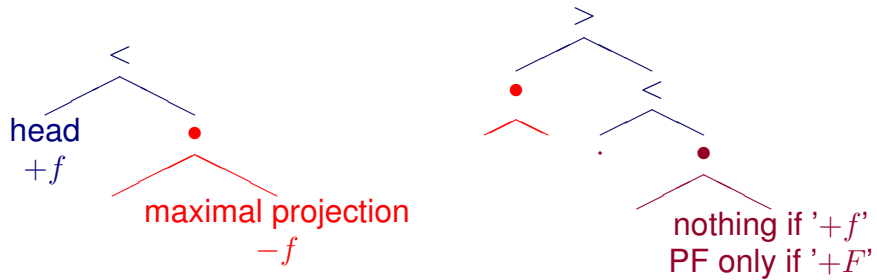
$$merge(t_1[= c], t_2[c]) = \begin{cases} \begin{array}{c} < \\ t_1 \quad t_2 \end{array} & \begin{array}{c} > \\ t_2 \quad t_1 \end{array} \\ \text{if } t_1 \in Lex & \text{otherwise} \end{cases}$$

Other type of Merge : Head Movement : "<=" and ">=" operation.

$$merge(t_1[=> c], t_2[c]) = \begin{array}{c} < \\ t_1 \quad t_2 \\ /PF(t_2)//PF(t_1)/ \end{array} \quad merge(t_1[c <=], t_2[c]) = \begin{array}{c} < \\ t_1 \quad t_2 \\ /PF(t_1)//PF(t_2)/ \end{array}$$

- move - transformation of a tree.

$Tree \rightarrow Tree$



2. The lexicon as automaton

We represent a lexicon as a automaton with product arrows.

For Every different kind on lexical entries, we associate, if the entry begin by :

- a base feature : a node.

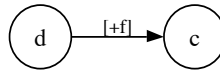
Lexical entry : $[\]::[a ; +f]. \Rightarrow$



The rest of the sequence feature will be place on all the arrow from this node.

- a selector feature : an arrow from the selector feature to the base feature. The rest of the sequence is put on the arrow

Lexical entry : $[\]::[=d ; +f ; c]. \Rightarrow$

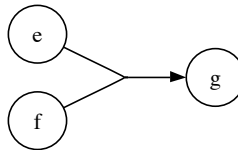


- if the entry contains several selectors, we use a produc arrow. In this case, we lose the automata structure.

Lexical entries :

`[]::[=e ; =f ; g].`

`[]::[f]. ⇒`



3. French clitics - Stabler's version

3.1. the two different parts of clitics interaction in the analysis

We distinguish two parts in treatment of clitics.

- One phonologically empty which take argumental place in the verb.
- The second doesn't add any syntactic component. We use this as a phonological mark of the clitic.

The two different parts are connected by a move operation. If either is not in the sentence, the derivation fails.

Sketch of derivation for clitics :

- (1)
- répare $\epsilon[-F]$
 - $la[+F]$ répare $\epsilon[-F] \Rightarrow la$ répare $\epsilon[la]$
 - Jean $[-k]$ la répare $\epsilon[la]$
 - $t[\epsilon]$ Jean $[-k]$ la répare $\epsilon[la]$
 - Jean $t[\epsilon]$ $t[Jean]$ la répare $\epsilon[la]$

3.2. Stabler's version

The lexicon :

$\epsilon ::= \text{Refl12} +k \text{ T}$	$\epsilon ::= \text{Acc3} +k \text{ T}$	$\epsilon ::= \text{Dat3} +k \text{ T}$	$\epsilon ::= =v +k \text{ T}$
$\text{se} ::= \text{Acc3} +F \text{ Refl12}$	$\text{se} ::= \text{Dat3} +F \text{ Refl12}$	$\text{se} ::= =v +F \text{ Refl12}$	
$\text{le} ::= \text{Dat3} +G \text{ Acc3}$	$\text{le} ::= v +G \text{ Acc3}$		$\text{lui} ::= v +F \text{ Dat3}$
$\epsilon ::= \text{vacc} \leq =D \text{ v}$	$\epsilon ::= \text{vdat} \leq =D +k \text{ vacc}$	$\epsilon ::= V \leq =p \text{ vdat}$	
$\text{montrera}::V$	$\text{donne}::V$	$\epsilon ::= >V =D +k =D \text{ v}$	
$\epsilon ::= P \leq =p$	$\text{a}::=D +k \text{ P}$	$\epsilon ::= p -F$	
$\text{Jean}::D -k$	$\text{Marie}::D -k$	$\text{roi}::N$	$\text{livre}::N$
$\text{le} ::= =N \text{ D} -k$	$\epsilon ::= D -k -F$	$\epsilon ::= D -k -G$	$\epsilon ::= =T \text{ C}$

the equivalent presentation as a pseudo-automata :

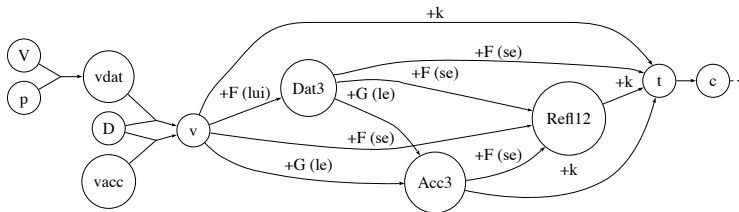


Figure 1: Stabler's version of french clitics treatment

4. extension for French clitics

Order of french clitics :

$$[suj|ne|\{me/te/se/\dots\}|\{le/la/les/\dots\}|\{lui/leur\}|y|en]$$

4.1. clitics after

The treatment of clitics used check them in the reverse order of the one in the sentence.

We must first treat the clitics closest to the verb

to the genitive	:	[en>::[clitic<=, +'EN', genitive].
example from genitive to reflexive	:	[se>::[genitive<=, +'F', refl]. [me>::[genitive<=, +'F', refl].
example from genitive to accusative	:	[le>::[genitive<=, +'G', acc]. [les>::[genitive<=, +'G', acc].
exemple de genitive vers datif	:	[lui>::[genitive<=, +'F', dat]. [leur>::[genitive<=, +'F', dat].
example from genitive to oblique	:	[y>::[genitive<=, +'Y', oblique].
simple exit from genitive	:	[]::[genitive<=, finclitic].

to oblique	:	[y]::[clitic<=, +'Y', oblique]. [y]::[genitive<=, +'Y', oblique].
example exit from oblique to reflexive	:	[se]::[oblique<=, +'F', refl]. [me]::[oblique<=, +'F', refl].
example exit from oblique to accusative	:	[le]::[oblique<=, +'G', acc]. [les]::[oblique<=, +'G', acc].
example exit from oblique to dative	:	[lui]::[oblique<=, +'F', dat]. [leur]::[oblique<=, +'F', dat].
simple exit from oblique	:	[]::[oblique<=, finclitic].

4.2. Subject pronoun and negation

4.2.1. negation

We treat the first part of the negation as a clitic because we consider that in French, clitics are a homogenous sequence placed before the main verb of the sentence.

[ne]:[neg<=, +case, t].	[pas]:[=>clitic, neg].
[ne]:[neg<=, neg2].	[pas]:[=>finclitic, neg].
	[pas]:[=>Big_v, neg].

Remark 1 *This treatment allows to place the clitic before or after the verb. It depends of the language. In French, we consider there is always clitics climbing.*

4.2.2. Subject pronoun

The trace subject clitic :

[ɔ̃]:[d, -Suj, -case].

The passage in the nominative clitic form :

[je]:[=finclitic, + Suj, Nom].	[il]:[=finclitic, + Suj, Nom].	[nous]:[=finclitic, + Suj, Nom].
[tu]:[=finclitic, + Suj, Nom].	[elle]:[=finclitic, + Suj, Nom].	[vous]:[=finclitic, + Suj, Nom].
[on]:[=finclitic, + Suj, Nom].	[ils]:[=finclitic, + Suj, Nom].	[elles]:[=finclitic, + Suj, Nom].

The exit from the nominative clitic to the end of the derivation.

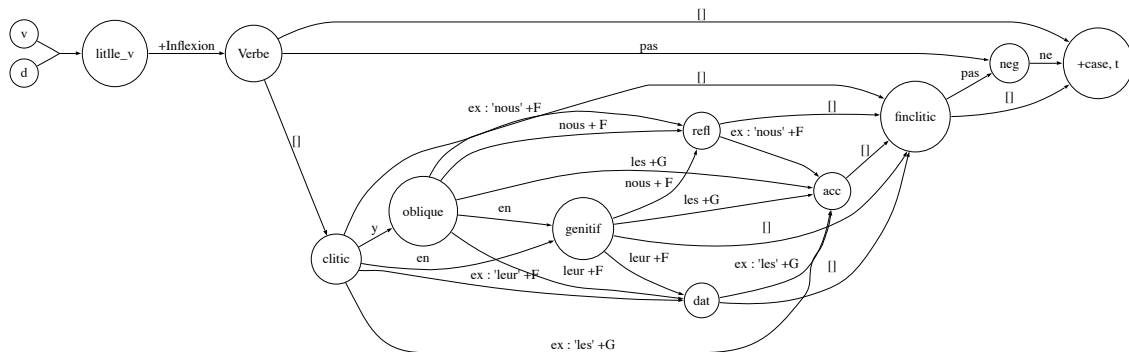
[ɔ̃]:[=nom, +case, t].

We definitively quit the cliticization.

5. exemple de dérivation

5.1. Lexical representation

Lexicon for french clitics :



5.2. analyses of a sentence

We present the derivation of

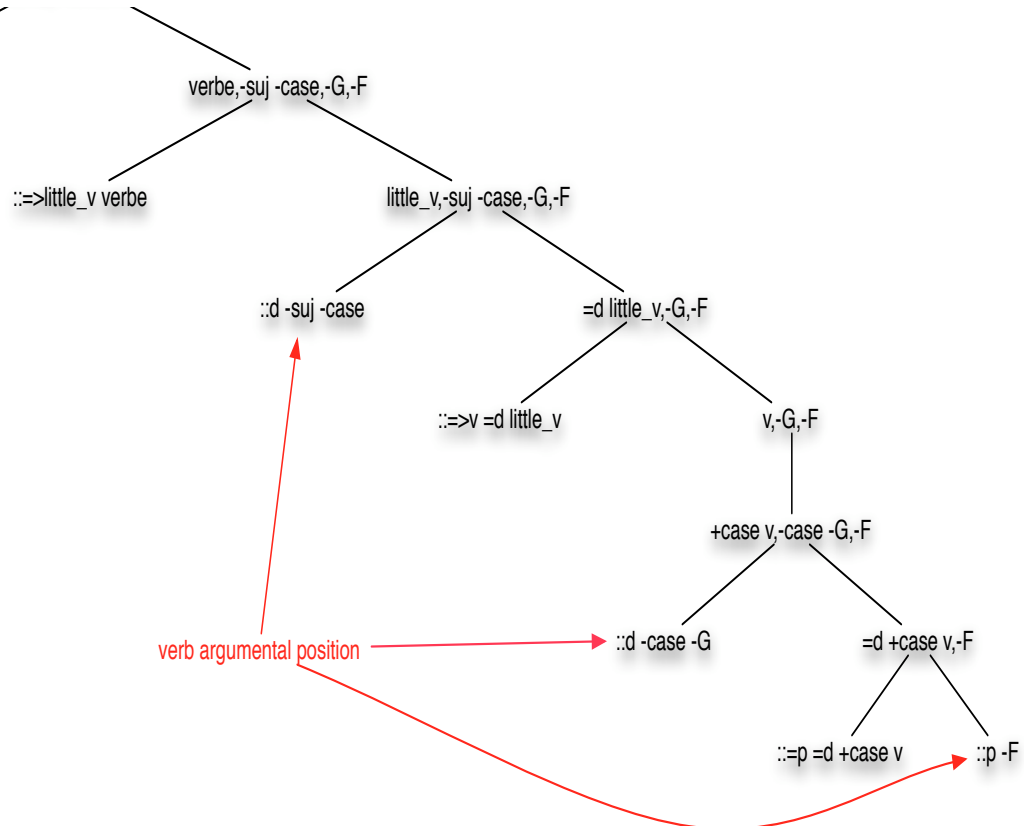
(2) *Je ne la lui donne pas.*
I ne her it give not.
I don't give it to her.

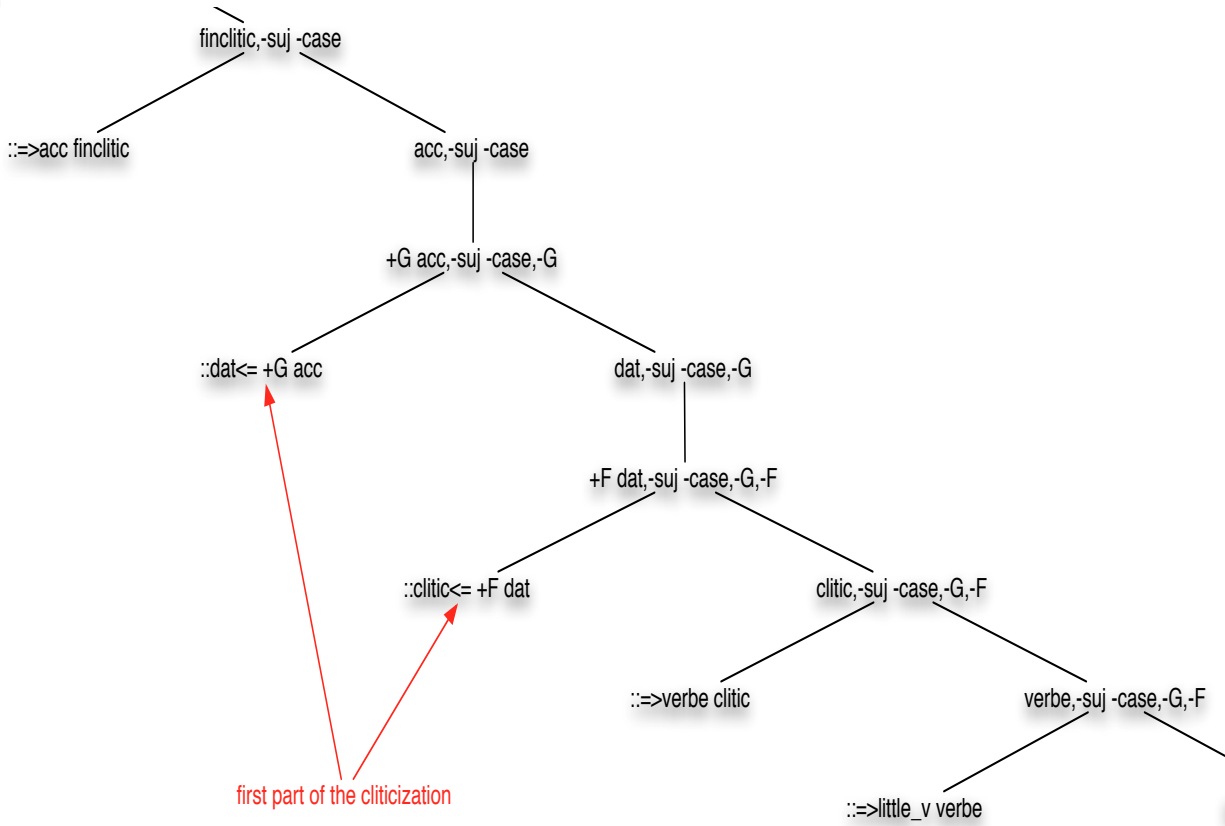
We propose three phases :

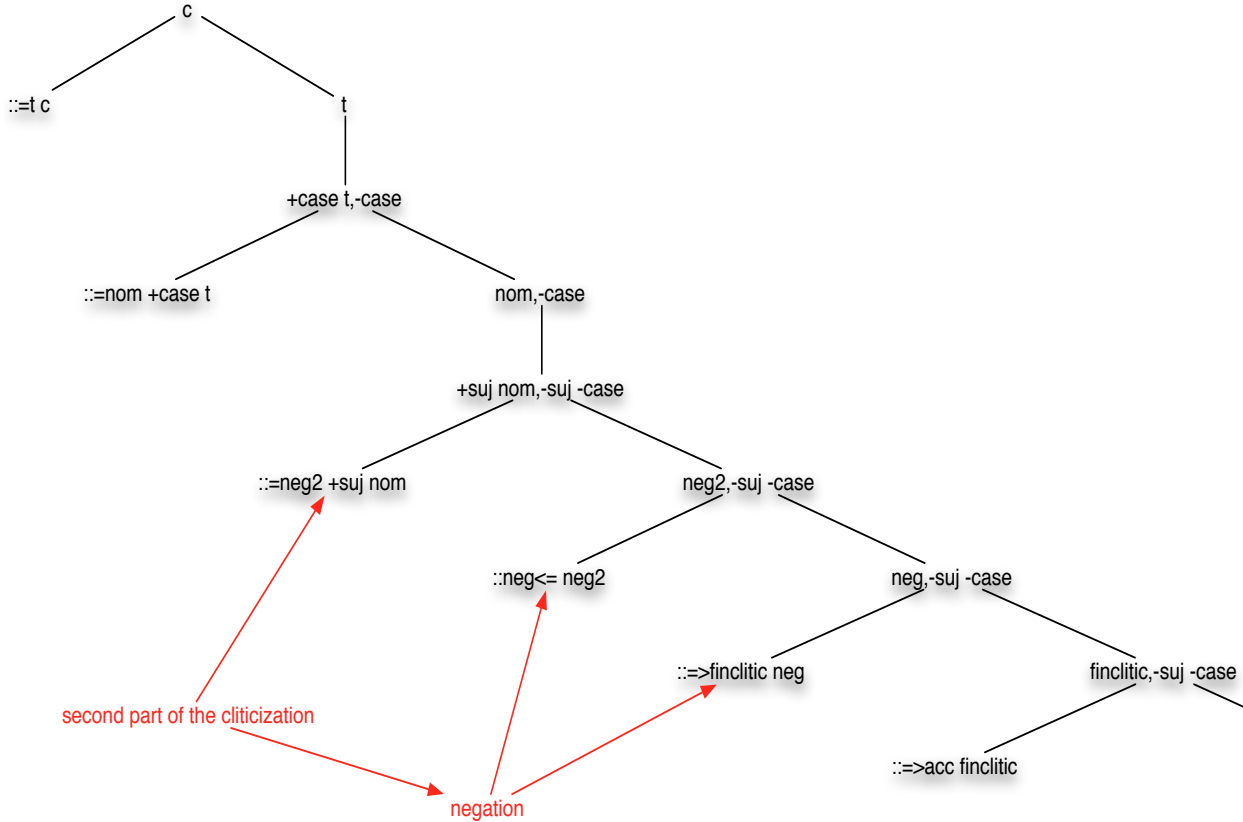
1. verb and argumental places of clitics.
2. first part of clitics.
3. negation and subject treatment

The analyse is given by the parser of John Hale with the lexicon present here.

```
# dessinn "je ne le lui donne pas";;  
accepted as category c in 4.93 seconds
```







6. A semantics for minimalist grammars

To the syntactic calculus, we add a semantic calculus.
For this, we use the λ -calculus.

Définition 1 *For every entry in the lexicon, we add a semantic part : a λ -term.*

For each of syntactic rules, we give a semantic rule.

We define the semantic calculus as the immediate application between two terms when one is completely discharged of its features.

When the application done, we don't have to do anything for the move operation - it is just a rearrangement of the syntactic structure. Normally, it is a λ -abstraction.

- a merge is an application
- a move is a λ -abstraction.

remarks on the semantic calculus :

There are two different options for the semantic calculus using the λ -calculus.

1. We could immediately do the application as we just propose but we could not treat some syntactic rules as "late adjunction".
2. We could wait upon the element has completely discharged its features before do the application. In this case, we could have problems with several sentences, for example question. For this kind of sentence, the object of the question has one additional feature. It must wait until the end of the derivation to be moved to first position. We must have treated all the other applications before, and this element will not take the right argumental place in the verb.

A good solution seems to be with the identification of a subset of Base which block the semantic reduction. The applications can be done when the lexical items are completely discharged of features of this type.

We keep the second option but with the Base set.

Définition 2 We define functions :

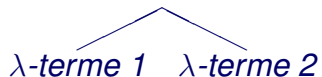
$$\text{feat}(x) = \begin{cases} 1 & \text{if the number of feature of } x = 0 \\ 0 & \text{else} \end{cases}$$

and

$$\text{sem}(x, y) = \begin{cases} 1 & \text{if } \text{feat}(x) = 1 \text{ or } \text{feat}(y) = 1 \\ 0 & \text{else} \end{cases}$$

Définition 3 We define semantic composition as follow :

if $\text{sem}(\lambda\text{-terme } 1, \lambda\text{-terme } 2) = 1$
 $>\vdash \lambda\text{-terme } 1 @ \lambda\text{-terme } 2$



else
 $>\vdash \lambda\text{-terme } 1, \lambda\text{-terme } 2$



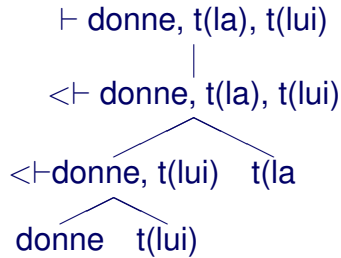
7. semantic of french clitics

Lexicon with semantic form : we add a λ -term to every lexical entries

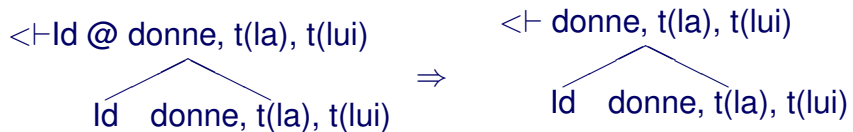
lexical entries	syntactic form	semantic form
<i>je</i>	$d - case$	moi^e
<i>le</i>	$dat \leq +G acc$	Id
<i>lui</i>	$clitic \leq +F dat$	Id
<i>donne</i>	$= p = d + case v$	$\lambda x \lambda y \lambda z. (donne z y x)$
$t(lui)$	$p - F$	x^*
$t(la)$	$p - case - G$	x^*
<i>pas</i>	$\Rightarrow finclitic neg$	Id
<i>ne</i>	$\Rightarrow neg + case t$	$\lambda P. \neg P$
inflexion	$\Rightarrow llittle_v verbe$	$\lambda P.pres(P)$
verbal form complement	$\Rightarrow v = d little_v$	Id
comp	$= t c$	Id
exit of the cliticization	$\Rightarrow acc finclitic$	Id
entry in the cliticization	$\Rightarrow verbe clitic$	Id

* Free variable but determined in the context

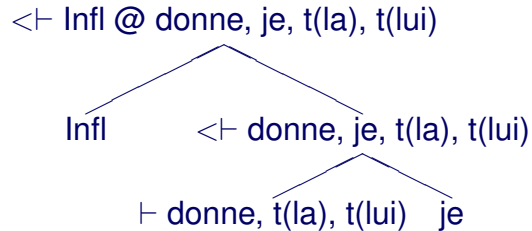
- First part of the calculation in the semantic counterpart : application of the first part of the two clitics.



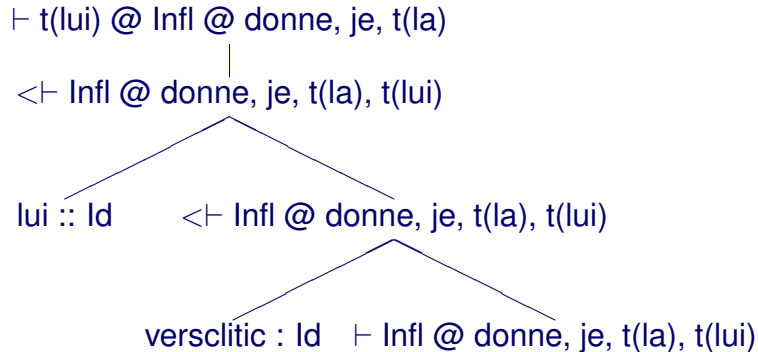
- We add the subject possibility. Here, the semantic part is the identity and $\forall x, \text{sem}(\text{donne}, x) = 1$. We reduce the representation.



- We add the first part of the subject clitic. The next step is the inflexion.



- We pass to the cliticization with a semantic identity which is immediately applied and we treat the phonological form of "lui".
The move operation discharges the clitic lui and $\forall x, sem(t(lui), x) = 1$ so we do the application.



- The next step treat the second clitic : "la". We discharge the clitic "la" with a move and $\forall x, sem(t(la), x) = 1$.

$\vdash t(la) @ t(lui) @ \text{Infl} @ \text{donne, je}$

$\langle \vdash t(lui) @ \text{Infl} @ \text{donne, je, } t(la)$

$la :: \text{Id} \quad \vdash t(lui) @ \text{Infl} @ \text{donne, je, } t(la)$

- After, we exit the first part of the clisization and we pass to the negation.

It is the second part of the negation which add the semantic negation.

$\langle \vdash \text{neg} @ t(\text{la}) @ t(\text{lui}) @ \text{Infl} @ \text{donne, je}$

$\text{neg} \quad \langle \vdash t(\text{la}) @ t(\text{lui}) @ \text{Infl} @ \text{donne, je}$

$\text{pas} :: \text{Id} \quad \langle \vdash t(\text{la}) @ t(\text{lui}) @ \text{Infl} @ \text{donne, je}$

$\text{finclitic} : \text{Id} \quad \vdash t(\text{la}) @ t(\text{lui}) @ \text{Infl} @ \text{donne, je}$



8. conclusion and futur works

- We present here a first extension of the french clitics analysis with Minimalist Grammars.
- In french, clitics are nearly present. It is really necessary to recognise them to increase the coverage of our grammars.
- The aim of this extension is to be used with the synchronisation of semantic calculus.
- From a syntactic analysis, we can find a semantic representation (in first order logic with λ -terms) of the sentence.

We now want extend the treatment of french clitics with raising/control verbs.

(3) *Je la laisse le lui donner*
I her let it (to) him give
I let her give it to him

(4) *Je la fait réparer*
I it let repaire
I'm having it repaired

(5) *Je pense la réparer*
I think it repaire.
I think about repairing

First, we need to make the syntactic analyse of this phenomena and after reconize the semantic form.

There is strange quantifier scope ambiguities with clitics and semi-auxiliary, in sentence like :

(6) *J'en ai vu un.*
I them have saw one.

(7) *Je les ai tous vu.*
I them have all saw