Generating French with a Reversible Unification Grammar

Dominique Estival

ISSCO
54 rte des Acacias, CH-1227 Geneva

0. Introduction

In this paper, we describe the linguistic solutions to some of the problems encountered in writing a reversible French grammar. This grammar is primarily intended to be one of the components of a machine translation system built using ELU,[1] an enhanced PATR-II style unification grammar linguistic environment based on the UD system described in Johnson and Rosner (1989), but it is also part of our more general experimentation with fully reversible grammars.

The requirement that it be reversible imposes a stringent criterion of linguistic adequacy on a grammar, since it is not allowed to overgenerate while it must at the same time provide a large coverage for analysis (Dymetman and Isabelle (1988)). Formally, grammars that are fully reversible must be completely declarative, since no reference can be made in the grammar rules to the process (analyzer or synthesizer) which will use them. The unification formalism makes it possible to write such grammar statements, because due to the associativity and commutativity of the unification operation, the result of unifying feature structures is independent of the order in which they are unified (Appelt (1989)).

Writing reversible grammars, however, presents problems which do not arise in the traditional grammars used for either analysis or generation. In addition, the progress accomplished recently in building generators for unification grammars has already revealed some of the problems posed by unification-based reversible grammars.[2] As shown by Russell et al. (1990), even though the grammar rules do not refer to the generation process, the generation algorithm imposes particular constraints on the grammar formalism.[3]

This paper concentrates particularly on the problems encountered in the generation of French, specifically in the analysis to be given to clitics.

We first briefly describe the aspects of the generation algorithm and of the grammar formalism which are relevant to the particular problems under discussion, then present the facts of French syntax which pose those problems and the solutions we have adopted.

1. The Generator

The generation algorithm of ELU is based on the algorithm described in Shieber et al. (1989) and was developed at ISSCO by J. Carroll.[4] Generation is head-driven: each rule has a "semantic head" (see Shieber (1988)), which is specified by the grammar writer, and the head daughter of a rule is generated before its siblings. The depth-first algorithm defines a downward path through the semantic heads of rules.

This algorithm does not require that the grammar be semantically monotonic. Non-monotonicity is obtained by having the generator distinguish two types of rules in the grammar, "chaining" and "non-chaining" rules, and by introducing the notion of "pivot". Following from this distinction, it employs both bottom-up and top-down processing.

The partition of the set of grammar rules into chaining and non-chaining rules is pre-compiled from the specification of what counts as the "semantics" of a feature structure. In a chaining rule, the mother and the head daughter have identical semantics; chaining rules are used bottom-up from the "pivot", which is defined as the lowest point in the path through the head daughters of chaining rules at which the semantics of the feature structure remains unchanged. In a non-chaining rule, the mother and the head daughter have different semantics; non-chaining rules are used top-down from the pivot.

The efficiency of the ELU generator depends in a large part on the restrictors defined by the grammar writer. Computing the pivot, i.e. creating a reachability table for chaining rules, and bottom-up processing

[1] ELU stands for *Environnement Linguistique d'Unification*.

[2] See Dymetman and Isabelle (1988), Shieber (1988), Shieber et al. (1989), van Noord (1988), inter alia.

[3] For instance, we cannot compare our grammar with the one presented in Saint-Dizier (1989), since his grammar is neither reversible nor purely declarative, as the rules are annotated with 'generation points'.

[4] Cf. Russell et al. (1990) for a description of the differences between the two algorithms.

are both controlled by pre-compiled "linking" information, which is encoded as sets of restrictor values. A restrictor is a specification of a value which can be computed from a feature structure (syntactic category, for example, is often defined as a restrictor). Before attempting unification between two feature structures, the values for the restrictors are checked in both of them; if these values are not compatible, unification would be bound to fail and is not tried. As linking information is only relevant for chaining rules, it is only used bottom-up during processing, and since by definition, chaining rules have the same semantics for their heads, linking information must be syntactic. Restrictors are also used heavily in the selection of lexical items, so the attributes chosen as restrictors have to be good discriminants between feature structures.[5]

The generation algorithm by itself guarantees neither the **completeness** nor the **coherence** of the resulting feature structure. The responsibility for preventing the generation of structures which unify with the input, but are incomplete (i.e. ensuring completeness) rests with the grammar writer: any structure which needs to be generated in its entirety should not be represented as an unconstrained feature structure, but must be specified as another data type, i.e. a list, a tree, or a user-defined type expression. The grammar writer and the generator share the responsibility for preventing additions to the input structure (i.e. preserving coherence): the grammar writer must again select the appropriate data types, and the generator "freezes" uninstantiated variables that occur in the input.

The choice of appropriate data types as well as of good restrictors is therefore crucial to ensure that the grammar is not only efficient but usable in generation.

2. The Grammar Formalism.

The syntactic representations built by the parser are trees where each node is a directed acyclic graph consisting of attribute-value pairs (i.e. a feature structure which allows reentrancy). The semantic representations used as input by the generator are feature structures derived from the syntactic trees.

The grammar rules consist of context-free phrase structure rules annotated with constraint equations expressing relations between the categories mentioned in the rule. The ELU formalism provides a generalization of the template facility of PATR-II, the "relational abstractions", which are statements abstracting over sets of constraint equations. These statements may receive multiple and recursive definitions. To give multiple definitions to a relational abstraction permits collapsing what in an unextended PATR-like formalism would be several distinct rules, and is a powerful way to capture linguistic generalizations. Multiple definitions, however, give rise to a high degree of non-determinism during processing. Therefore, while the parser expands multiple definitions whenever they are encountered, the generator uses a lazier approach and only expands them when they are needed. Nevertheless, this strategy is not sufficient, and the problem posed by the non-determinism of relational abstractions is the most complex and severe of the grammar/generator interactions described in Russell et al. (1990), because of its adverse effects on the restriction of top-down generation.

3. French Clitics

Any French grammar must account for the position and ordering of preverbal clitics. While full complement and modifier phrases occur to the right of the main verb of a clause, up to three clitics may occur in front of a verb, as in (1).

(1) Il *m'y en* a fait part.
he me there of it informed
He informed me of it there.

Moreover, the clitics must appear in a fixed order, which, as shown in (2), is independent of the semantics of the clitics.

(2) a. Ils *vous l'y* ont donnée.
they to you it there gave
They gave it to you there.
b. *Ils *leur l'y* ont donnée.
they to them it there gave
They gave it to them there.

This fixed order can be represented by the traditional table given in (3).[6]

(3) **Ordering of French clitics**

me	le	lui	y	en
te	la	leur		
se	les			
nous				
vous				

In most accounts of the distribution shown in (3), the problem is simplified, because only subcategorized complements are dealt with. A French preverbal clitic, however, is not necessarily a subcategorized complement of the verb; adverbials and parts of complement phrases can also cliticize, and the grammatical category of some clitics is that of adverbs or

[5] Restrictors are also used to restrict the search space in parsing (see Shieber (1985)). The use of linking information in generation was first proposed by van Noord (1988).

[6] In (3), *se* stands for any of the so-called 'R-clitics', i.e. the reflexive and reciprocal pronouns, as well as the inherent reflexive and the middle marker, as explained in more detail below.

quantifiers.

The contrast between (4.a) and (4.b) shows that a clitic can be either a full complement, or part of a complement. In (4.a), *en* is the full prepositional object of the verb *parler*, while in (4.b), *en* represents the partitive prepositional phrase which is the complement of the object of *vouloir*.

(4) a. Il *en* parlait souvent.
he often talked about it
[cf. Il parlait souvent *de ce livre*]
[*he often talked about that book*]
b. J'*en* veux deux.
I want two of them
[cf. Je veux deux *de ces pommes*]
[*I want two of these apples*]

The contrast between (5.a) and (5.b) shows that a clitic can either be subcategorized or not. In (5.a), *y* is the subcategorized complement of the verb *aller*, while in (5.b), *y* is a locative adverb, which is not subcategorized by the verb *dormir*.

(5) a. Il *y* allait souvent.
he often went there
[cf. Il allait souvent *dans cette ville*]
[*he often went to that city*]
b. Il *y* dormait souvent.
he often slept there.
[cf. Il dormait souvent *dans cet hôtel*]
[*he often slept in that hotel*]

Besides the personal pronouns and the adverbs given in Table (3), there are other lexical items which are not usually considered in the treatment of French clitics, but whose behavior is closely related.[7] The negative elements *pas*, *plus*, *jamais*, *rien*, and the quantifiers *tant*, *autant*, *plus*, *moins* and *tout*[8] also cliticize and may appear preverbally.

While all the clitics of Table (3) must appear in front of the traditional AUX constituent (i.e. before any of the verbal elements of the VP), the examples in (6) show that the elements of this second set appear inside AUX, more precisely after the first tense-bearing verbal form.

(6) a. Il n'*en* avait *jamais* été persuadé.
he had never been sure of it
b. Il n'*en* avait *jamais rien* cru.
he had never believed any of it
c. Je n'*y en* ai *jamais autant* vu.
I had never seen so many of them there

There are thus at least two slots for clitics inside a French VP, and neither of these slots correlates with argumenthood. The quantifiers *rien* and *autant* which appear inside AUX in (6.b) and (6.c) are (parts) of the argument of the verbs *croire* and *voir*, and so is the quantifier *en*, which is in front of the AUX. On the other hand, the adverb *y* is not an argument in (6.c), nor is *jamais* in (6.a-c).

Therefore, the lexical entry of every clitic element must specify not only that it is a clitic but whether it appears in front of or inside the AUX constituent.

4. Generation

Theoretically, the fundamental problem posed by clitics stems from their dual nature, syntactic and morphological, and partly consists in deciding whether to treat them by syntactic or by morphological processes.[9]

Descriptively, there are three issues to be addressed: argument-binding, linear ordering relations, and categorial status of the clitics. All three give rise to problems in generation due to non-determinism, for which the solution is to ensure that the lexical verb is instantiated as soon as possible.

4.1. Subcategorization

The unification formalism makes it very natural to encode syntactic information in the lexicon and with a lexicalist approach, our treatment of arguments is straightforward: we make the standard use of a subcategorization list to encode the complements a verb requires. Since any complement phrase may be realized as a clitic, this fact is not mentioned in the subcategorization list.[10]

---

[7] They are, however, the subject of work in theoretical linguistics, see e.g. Perlmutter (1971), Emonds (1975), Kayne (1975), and more recently Pollock (1989). Interestingly, though it was developed in a different framework and for different reasons, our treatment of those elements is compatible with the latter's analysis (cf. also fn.9).

[8] The quantifier *tout* has actually several forms, inflected for gender and number: *tout/tous/toute/toutes*.

[9] E.g., restrictions on coordination show that clitics are not independent syntactic constituents.
(i) * Il *me et te* connait.
he knows me and you
Cf. the various analyses presented in Borer (1986). More recently (Rizzi and Roberts (1989), Kayne (1990)), the question has been reformulated in terms of the type of mechanism (adjunction or substitution) involved in cliticization and of whether clitics are phrasal heads or not. With the lexicalist approach adopted in our grammars both types of processes can be referred to in the lexicon, but it is of course still desirable that the two be clearly separated.

[10] This analysis contrast with that of Baschung et al. (1987), or Bès et al. (1989), which treats separately complements appearing to the left and complements appearing to the right of the verb. Their reason for doing so is that they take the variants shown in (i) and (ii) to indicate a relatively free order of complements (subcategorized or not) in French.
(i) il a donné (hier) un livre à Marie (hier).
yesterday he gave a book to Marie
(ii) il a donné (hier) à Marie un livre (hier).
yesterday he gave a book to Marie
While the ordering of full complements inside the VP poses some problems for generation, it is a separate question from that of cliticization, and the two should receive principled solutions of their own.

During analysis, an element found in the VP is checked against *Subcat*, the subcategorization list of the predicate. If it does not unify with any element of *Subcat*, it is treated as a VP modifier and added to *Mods*, the list of modifiers. From the point of view of generation, clitics realize elements from either the *Subcat* list or from the *Mods* list.

For instance, we partly follow the lexicalist analysis of Grimshaw (1982) for the R-clitics represented by *se*. That is, we consider that the R-clitic is not an argument of "inherently reflexive" (7.a,b) and "middle" verbs (7.c), but a morpho-syntactic marker.[11]

(7) a. Il *s'*est évanoui.
he fainted
b. Il *se* le demandait.
he was wondering about it
c. Il *s'*est cassé.
it broke

But in reciprocal and true reflexive constructions, such as (8.a,b), we treat the R-clitic as a pronoun which is an argument of the verb.[12]

(8) a. Ils *se* sont regardés.
they looked at each other/themselves
b. Ils *se* les sont donnés.
they gave them to each other/to themselves

Therefore, because the verbs in the examples of (7) are marked in the lexicon as being inherently reflexive, an R-clitic is generated from *Subcat* without being bound to the list of semantic arguments. In (8) on the other hand, the verbs are respectively transitive and ditransitive: in their case, a semantic argument is both bound to an element of *Subcat* and realized as a reflexive pronoun because of its own semantic features. In (9.a-c) *se* is, as in (7), the inherent reflexive marker and is generated from *Subcat*. In (9.a) *en* is the partitive phrase of a subcategorized argument; *y* in (9.b) is a subcategorized locative argument from *Subcat* and in (9.c), it is a VP adverb from *Mods*.

(9) a. Il *s'en* est cassé deux.
two of them broke
b. Ils *s'y* trouvaient.
they were there
c. Ils *s'y* vendaient.
they were sold there

As described in Russell et al. (1990), problems arise in generation because of non-determinism and because of the unavailability of some syntactic information to the generator. The subcategorization list mechanism typical of unification grammars is a source of both these kinds of problems. Subcategorization lists are relational abstractions with multiple definitions; therefore, they introduce non-determinism in the expansion of the rules in which they are invoked. Moreover, they exemplify the type of syntactic information typically found in lexical entries; this information is not available to the generator until the lexical head has been instantiated, but if it was available at a higher point in the path through the rules it would help constrain the top-down search.

In particular, here, separating the elements found inside the VP into arguments and modifiers can only be done after the lexical head has been instantiated and its subcategorization list is available. As shown by the two meanings of the verb *trouver* given in the lexical entries (10.a,c) and exemplified in (10.b,d), the semantics of the verb (its argument list) may change according to its subcategorization list.

(10) a. trouver * v {+ UN}
!Verb !main !avoir !Nrefl !trans
!Subcat(np,np)
b. Il *l' y* trouve.
he finds it there
[cf. Il *le* trouve dans les Alpes.]
[*he finds it in the Alps*]
c. trouver * v {+ UN}
!Verb !main !être !Refl !intrans
!Subcat(np,pp) !PPsem(loc)
d. Il *s'y* trouve.
it is located there
[cf. Il *se* trouve dans les Alpes.]
[*it is located in the Alps*]

In (10.d) the clitic *y* is an argument (i.e. it is bound to one of the variables in the arguments list), while in (10.b) it is not (i.e. it is added to the modifiers list). Even though the two possiblities are mutually exclusive, if the subcat list is not available at the VP level, the search must proceed top-down and the VP is expanded top-down and non-deterministically. Recall that when the semantics for the head daughter of a rule does not change, the rule is a chaining rule which is used bottom-up, but if the semantics of the head changes, then the rule is a non-chaining rule, which is used top-down and defines a pivot.

4.2. Linear ordering

As was shown by the examples of (2), the linear ordering among preverbal clitics is independent of their semantics; it is also independent of the syntactic features of their dominating clause, i.e. negation, inversion, etc. A perspicuous way to express clitic ordering is to have one relational abstraction with separate definitions stating the different precedence constraints holding between two preverbal clitics. The simplified definitions for *Precede(C1,C2)* given in (11) would account for most of the distribution

[11] In (7.a), there is no non-reflexive verb *évanouir*, and in (7.b), the reflexive verb has a different semantics than the non-reflexive verb from which it is lexically derived.

[12] In this respect, our analysis also differs from that presented in Wehrli (1986).

facts of Table (3) in a natural and elegant way.[13]

(11) Precede(C1,C2)
<C1 head morph pers> = 1/2
<C2 head morph pers> = 3
Precede(C1,C2)
<C1 head morph case> = acc/refl
<C2 head morph case> = dat
Precede(C1,C2)
<C1 head morph case> = refl
<C2 head morph case> = acc/dat
Precede(C1,C2)
<C1 head sem pred> = y
<C2 head sem pred> = en

4.3. Categorial status

A characteristic property of clitics is that they do not have a maximal projection and remain X° constituents, with their own syntactic category feature coming from the lexicon. To express the fact that a dative pronoun or the clitics *y* and *en* actually stand for a PP can be done by building a PP in the lexicon, e.g. with a relational abstraction such as *Make-PP(Cl,PP)*.

(12) Make-PP(Cl,PP)
<Cl head sem pred> = ˜y/en
<Cl head morph case> = dat
<PP head sem pred> = à
<PP head sem args> = [<Cl>]
Make-PP(Cl,PP)
<Cl head sem pred> = y
<PP head sem pred> = à
<PP head sem args> = [<Cl>]
Make-PP(Cl,PP)
<Cl head sem pred> = en
<PP head sem pred> = de
<PP head sem args> = [<Cl>]

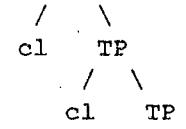
The relational abstractions *Precede* and *Make-PP* constitute an elegant collapsing of syntactic and lexical rules which is useful in analysis: the grammar rules which rewrite VPs containing clitics need not specify all the various possibilities. However, as with the relational abstractions encoding subcategorization facts, its multiple definitions render *Precede* nondeterministic. The non-determinism of *Make-PP*, which is due to the fact that some clitic forms are ambiguous, is no less severe. During generation, the evaluation of the equations is delayed until the semantics for the head has been instantiated, and if the lexical head is not instantiated early enough, rules which involve these relational abstractions are tried repeatedly even if they cannot apply.

In conclusion, for the purpose of generation, we need an analysis where the semantic head of the VP is not necessarily the lexical main verb, but is the element which will be sure to be instantiated as early as possible. In an analysis reminiscent of current work in the Government-Binding framework,[14] where a clause is IP (Inflectional Phrase), the maximal projection of INFL, we take as the semantic head for our rules the element which bears tense. This element, I, may be either the main verb or an auxiliary which takes the main verb as complement.

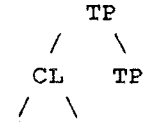
With this analysis of the structure of VP, the semantics of the head daughter I remains the same along the path through the semantic heads so that the pivot of the structure, i.e. the point at which bottom-up generation can start from, is at the end of path. At that point, either I is the main verb (V-raising has applied) and it can be instantiated immediately, or I is an auxiliary (V-raising hasn't applied) and the main verb is its sister, which can be reached through other chain rules.

We can deal with clitics in two ways:

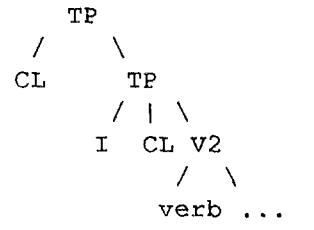
• by successive adjunction to the head:

(13)


• by building a list, itself adjoined to the head:

(14)


Besides being descriptively more adequate, since the ordering constraints hold between the clitics themselves, not between a clitic and a verbal constituent, the second approach is to be preferred because a list ensures completeness of the resulting feature structure. Moreover, the whole list of clitics can be built without instantiating the lexical verbal head. With the two clitic positions and taking I as the head, the syntactic structure for a VP is as in (15).

(15)


[13] There are other constraints not accounted for by (11), e.g. the one requiring that an ambiguous acc/dat form cannot be interpreted as an accusative in front of a dative:

(i) * Elle nous lui présentera.
she us to him will introduce

Similar constraints exist among the clitic elements appearing in post-verbal position.

[14] Cf. fn.6 and 9, and work cited in the references given there.

Clitic elements are marked as to whether they must appear to the left or to the right of I. If V-raising hasn't applied, as in the examples of (6), the two clitic lists will be in front of the main verb, on either side of I. If V-raising has applied, the two clitic lists will still be on either side of I, and of the main verb, as in (16).

(16) a. Il ne *l'en* persuadera *jamais.*
he will never convince her of it
b. Il n'*en* croit *jamais rien.*
he never believes any of it
c. Je n'*y en* voit *jamais autant.*
I never see so many of them there

5. Conclusion

We have shown with the example of French clitics how some problems inherent in the writing of reversible grammars arise, and what aspects of the formalism are responsible for them. The solutions we propose are motivated by internal considerations and provides a coherent syntactic account of the phenomena under consideration, i.e. clitic placement and so-called "adverb climbing" (although space prevents us from showing the details here, they also deal adequately with verbal negation). These solutions make full use of the properties and advantages of the lexicalist approach to grammars while circumventing (some of) the dangers it presents.

** I am grateful to Susan Warwick and Graham Russell for the time they have spent helping me understand ELU and its generator. Neither of them, of course, is responsible for any mistake in this paper.

References

Appelt, D. (1989). "Bidirectional Grammars and the Design of Natural Language Systems". In *Theoretical issues in natural language processing*, edited by Y. Wilks. Hillsdale: Lawrence Erlbaum Associates.

Baschung, K, G. Bès, A. Corluy and T. Guillotin (1987). "Auxiliaries and Clitics in French UCG Grammar". In *Proceedings of the Third Conference of the European Chapter of the Association for Computational Linguistics*, Copenhagen, pp. 173-178.

Bès, G. and C. Gardent (1989). "French Order without Order". In *Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics*, Manchester, pp. 249-255.

Borer, H. (1986). ed. *The Syntax of Pronominal Clitics*, Syntax and Semantics, vol.19. Academic Press.

Dymetman, M. and P. Isabelle (1988). "Reversible Logic Grammars for Machine Translation". In *Proceedings of the Second International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Carnegie-Mellon University, Pittsburgh.

Emonds, J.E. (1975). "A transformational analysis of French clitics without positive output constraints". *Linguistic Analysis*, 1. 3-24.

Grimshaw, J. (1982). "On the lexical Representation of Romance Reflexives". In *The Mental Representation of Grammatical Relations*. edited by J. Bresnan. Cambridge: MIT Press, pp.87-148.

Johnson, R. and M. Rosner (1989). "A rich environment for experimentation with unification grammars". In *Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics*, Manchester, pp. 182-189.

Kayne, R. (1975). *French Syntax*. Cambridge: MIT Press.

Kayne, R. (1990). Seminar. University of Geneva.

van Noord, G. (1988). "BUG: A Directed Bottom Up Generator for Unification Based Formalisms". Dept. of Linguistics, Trans 10, Utrecht University.

Perlmutter, D. (1971). *Deep and Surface Structure Constraints in Syntax*. New York: Holt, Rinehart and Winston.

Pollock, J.-Y. (1989). "Verb Movement, Universal Grammar and the Structure of IP". *Linguistic Inquiry*, 20.3 pp. 365-424.

Rizzi, L. and I. Roberts (1990). "Complex Inversion in French". *Probus*, vol.1.1. pp. 1-30

Russell, G., S. Warwick and J. Carroll (1990). "Asymmetry in Parsing and Generating with Unification Grammars". to appear in the *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, Pittsburgh.

Saint-Dizier, P. (1989). "A Generation Method Based on Principles of Government-Binding Theory". In *Proceedings of the Second European Natural Language Generation Workshop*, Edinburgh.

Shieber, S. (1985). "Using Restriction to Extend Parsing Algorithms for Complex-Feature-Based Formalisms". In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, Chicago, pp. 145-152.

Shieber, S. (1988). "A Uniform Architecture for Parsing and Generation". *Proceedings of the 12th International Conference on Computational Linguistics*, Budapest, pp. 614-619.

Shieber, S., van Noord, G., R.C. Moore and F.C.N. Pereira (1989). "A Semantic-Head-Driven Generation Algorithm for Unification-Based Formalisms". In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, Vancouver.

Wehrli, E. (1986). "On Some Properties of French Clitic *Se*". In Borer (1986).