PARSING WITH GETARUNS

Rodolfo Delmonte, Denise Dibattista
Section of Linguistic Studies - DSAO
Università Ca' Foscari - Ca' Garzoni-Moro
San Marco, 3417 - 30124 Venezia (Italy)
Tel.:041-2578464/52/19
E-mail:delmont@unive.it website:byron.cgm.unive.it

ABSTRACT

GETARUNS, the system for text and reference understanding which is currently used for summarization and text generation has a highly linguistically sophisticated parser which implements a number of strategies to cope with ambiguity ensuing from PP attachment and other similar problems(see Delmonte & Dolci, 1997). In this paper we present the parser from a linguistic point of view and as such implementing LFG theoretical framework within a DCG, using Xtraposition Grammars to cope with Long Distance Dependencies. The parser is multilingual and contains a lookahead mechanism, which is then used by the Well-Formed-Substring-Table to recover wrongly parser attachment.

1. Introduction

The parser we present was conceived in the middle '80s and started as a Transfer module for a Machine Translation Expert system in a very restricted linguistic domain. Then it became a general parser for Italian and English, to be used with LFG students. German was added later on, beginning of '90s. Since the people working at it were interested in the semantics as much as in the syntax, it was soon enriched with a Quantifier Raising algorithm and an Anaphoric Binding Module. In 1994 the Discourse Model and the Inferential Processes algorithms were developed. Finally in 1996 work on a Situational Semantics interface and on the Discourse Structure was carried out. These experiments were finally enriched - two years ago - with a number of Parsing Strategies procedures like setting up a Lookahead mechanism, a Well-Formed Substring Table and a number of other semantically and/or lexically based triggering lookup procedures.

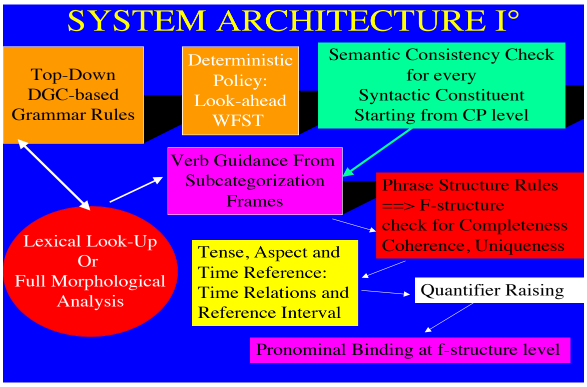
We worked from the very beginning within LFG framework which allowed us to think in terms of a much richer representation, closer to the semantics, already from the start than just a context-free syntactic constituency. In particular, all levels of Control mechanisms which allow coindexing at different levels of parsing gave us a powerful insight into the way in which the parser should be organized.

Yet the grammar formalism implemented in our system differs from the one suggested by the theory, in the sense that we do not use a specific Feature-Based Unification algorithm but a DCG-based parsing scheme. In order to follow LFG theory more closely, unification should have been implemented: but DCG gives us full control of a declarative rule-based system, where information is clearly spelled out and passed on and out to higher/lower levels of computation. The grammar is implemented in Prolog using XGs(extraposition grammars) introduced by Pereira(1981;1983). Prolog provides naturally for backtracking when allowed, i.e. no cut is present to prevent it. Furthermore, the instantiation of variables is a simple way for implementing the mechanism for feature percolation and/or for the creation of chains by means of index inheritance between a controller and a controllee, and in more complex cases, for instance in case of constituent ellipsis or deletion.

Apart from that, the grammar implemented is a surface grammar of the languages chosen. Also functional Control mechanisms – both structural and lexical - have been implemented as close as possible to the original formulation, i.e. by binding an empty operator in the subject position of a propositional like open complement/predicative function, whose predicate is constituted by the lexical head.

Of course there are a number of marked differences in the treatment of specific issues, concerning Romance languages, which were not sufficiently documented in the linguistic literature at the time. In particular,

- we introduced an empty subject pronominal - little pro - for tensed propositions, which had different referential properties from big PRO; this had an adverse effect on the way in which c-structure should be organized. We soon realized that it was much more efficient and effective to have a single declarative utterance-clause level where the subject constituent could be either morphologically expressed or Morphologically Unexpressed. In turn MUS or little pros could be computed as variables in case the subject was realized in postverbal position. At the time LFG posited the existence of a rule for sentence structure which could be rewritten as VP in case there was no subject, MUS, or in case the subject was expressed in postverbal position, an approach that we did not implement;
- we also use functional constituents like CP and IP: CP typically contains Aux-to-Comp and other preposed constituents, adjuncts and others; IP contains negation, clitics, and tensed verbal forms, simple and complex, and expands VPs as complements and postverbal adjuncts;
- each constituent is semantically checked for consistency before continuing parsing; we also check for Uniqueness automatically by variable instantiation. But sometimes, in particular for subject-verb agreement we have to suspend this process to check for the presence of a postverbal NP constituent which might be the subject in place of the one already parsed in preverbal position!!;
- syntactic constituency is replicated by functional constituency: subject and object are computed as constituents of the annotated c-structure, which rewrite NP - the same for ncomp - this is essential to assign the appropriate annotated grammatical function; this does not apply to VP, a typical LFG functional non-substantial constituent;
- our lexical forms diverge from the ones used in the theoretical framework: we introduced aspectual categories, semantic categories and selectional restrictions in the main lexical entry itself;
- we also have semantic roles already specified in the lexical form and visible at the level of syntactic-semantic parsing;
- rather than generating a c-structure representation to be mapped onto the f-structure via an annotated c-structure intermediate level (??), we already generated a fully annotated c-structure representation which was then checked for Grammatical Principles Consistency at the level of number/type of arguments and of Adequacy for adjuncts, with a second pass on the output of the parser, on the basis of lexical form of each predicate and semantic consistency crossed checks for adjuncts.


SYSTEM ARCHITECTURE I°

Top-Down DGC-based Grammar Rules

Deterministic Policy: Look-ahead WFST

Semantic Consistency Check for every Syntactic Constituent Starting from CP level

Verb Guidance From Subcategorization Frames

Phrase Structure Rules ==> F-structure check for Completeness Coherence, Uniqueness

Lexical Look-Up Or Full Morphological Analysis

Tense, Aspect and Time Reference: Time Relations and Reference Interval

Quantifier Raising

Pronominal Binding at f-structure level

TABLE 1. GETARUNS PARSER

All parser rules from lexicon to c-structure to f-structure amount to 1900 rules, thus subdivided:

1. Calls to lexical entries - morphology and lexical forms: 150 rules
2. Syntactic and semantic rules in the parser proper: 550 rules
3. Parsing strageties and other tools: 185 rules

All syntactic/semantic rules: 850 rules

4. Semantic Rules for F-Structure

Lexical Rules for Consistency and Control:

- semantic rules 439

F-structure building, F-command:

- semantic rules 170

Quantifier Raising and Anaphoric Control:

- semantic rules 441

All semantic f-structure building rules: 1050 rules

1.1 Grammar and Ambiguity

The Parser builds c-structure representations, which undergo grammatical wellformedness tests by which lexical semantic information is appended to each constituent. Finally constituent information is dropped and DAGs are built in order to produce f-structure configuration.

Each major constituents may be associated with different functional values:

a. NP --> SUBJect, both in preverbal and postverbal position - VP internally, VP adjoined and IP adjoined (see Delmonte, 1987) - with any kind of verbal category; OBJect, usually in VP internal position, but also in preverbal position at Spec CP in case of reversed transitive structures; NCOMP predicative function - if not proper noun - occuring with copulative, and ECM verbs like "consider, believe"; closed ADJunct with [temporal] value, as the corresponding English example "this morning", which however in Italian can be freely inserted in sentence structure;

b. AP --> Modifier of an NP head, occurring as attribute in prenominal and as predication in postnominal position; ACOMP predicative function occurring with copulative, and ECM verbs; open XADJunct occurring freely at sentence level. Other examples of open adjuncts are: floating quantifiers, which however may only occur VP internally; doubling emphatic pronoun "lui" which also occurs VP internally and is computed as open adjunct;

c. AdvP --> Open or closed Adjuncts according to its selectional properties, occurring anywhere in the sentence according to their semantic nature;

d. PP --> OBLiques, when selected by a given predicate; PCOMP predicative function, when selected by a given predicate - both these two types of argument usually occur VP internally but may be fronted; open XADJunct or closed ADJunct according to semantic compatibility checks;

e. VP' --> VCOMP infinitivals, when selected by a given predicate; SUBJect propositional clauses; closed ADJuncts with semantic markers like "for"; VP' gerundive and participial, which are always computed respectively as closed ADJuncts the former and as open ADJuncts the latter;

f. S' -->or CP as main clauses, or subordinate clauses, as well as sentential complements and SUBJect propositional clauses;

g. Clitics and Pronominal elements are also computed as Nps or PPs, because they are assigned grammatical functions when not associated to NP dislocation in preverbal position: in that case, the clitic is simply erased and TOPic function is associated with the binder NP.

The parser is made up of separate modules:

1. The Grammar, based on DCGs, incorporates Extraposition to process Long Distance Dependencies, which works on annotated c-structures: these constitute the output to the Interpretation Module;

2. The Interpretation Module checks whether f-structures may be associated to the input partially annotated c-structure by computing Functional Uniqueness, Coherence and Completeness. Semantic roles are associated to the input grammatical function labels at this level, after semantic selectional restrictions are checked for membership;

3. The Mapping scheme, to translate trees into graphs, i.e. to map c-strutures onto f-structures. The parser builds annotated c-structure, where the words of the input sentence are assigned syntactic constituency and functional annotations. This is then mapped onto f-structure.

2. Parsing Scheme

The parser looks for syntactic constituents adjoined at CP level: in case of failure, it calls for IP level constituents, including the SUBJect which may either be a clause or an NP. This is repeated until it reaches the Verbal Phrase: from that moment onward, the syntactic category associated to the main verb - transitive, unergative, unaccusative, impersonal, atmospheric, raising, psych, copulative - and the lexical form of the predicate, are both used as topdown guidelines for the surface realization of its arguments. Italian is a language which allows for empty or morphologically unexpressed Subjects, so that no restriction may be projected from the lexicon onto c-structure: in case it is empty, a little pro is built in subject position, and features are left as empty variables until the tensed verb is processed.

The grammar is equipped with a lexicon containing a list of fully specified inflected word forms where each entry is followed by its lemma and a list of morphological features, organized in the form of attribute-value pairs. However, morphological analyzers for Italian and English are also available with big root dictionaries (90,000 for Italian, 25,000 for English) which only provide for syntactic subcategorization, though. The fully specified lexicon has been developed for Italian, English and German and contains approximately 5,000 entries for each language.

Once the word has been recognized, lemmata are recovered by the parser in order to make available the lexical form associated to each predicate. Predicates are provided for all lexical categories, noun, verb, adjective and adverb and their description is a lexical form in the sense of LFG. It is composed both of functional and semantic specifications for each argument of the predicate: semantic selection is operated by means both of thematic role and inherent semantic features or selectional restrictions. Moreover, in order to select adjuncts appropriately at each level of constituency, semantic classes are added to more traditional syntactic ones like transitive, unaccusative, reflexive and so on. Semantic classes are of two kinds: the first class is related to extensionality vs intensionality, and is used to build discourse relations mainly; the second class is meant to capture aspectual restrictions which decide the appropriateness and adequacy of adjuncts, so that inappropriate ones are attached at a higher level.

Grammatical functions are used to build f-structures and the processing of pronominals. They are crucial in defining lexical control: as in Bresnan (1982), all predicative or open functions are assigned a controller, lexically or structurally. Lexical control is directly encoded in each predicate-argument structure, but see below.

Structural information is essential for the assignment of functions such as TOPic and FOCus. Questions and relatives, (Clitic) Left Dislocation and Topicalization are computed with the Left Extraposition formalism presented by Pereira(1981;1983). Procedurally speaking, the grammar is implemented using definite clauses. In particular, Extraposition Grammars allows for an adequate implementation of Long Distance Dependencies: restrictions on which path a certain fronted element may traverse in order to bind its empty variable are very easily described by allowing the prolog variable associated to the element in question - a wh- word or a relative pronoun - to be instantiated in a certain c-structure configuration. Structural information is then translated into functional schemata which are a mapping of annotated c-structures: syntactic constituency is now erased and only functional attribute-value pairs appear. Also lexical terminal categories are erased in favour of referential features for NP's determiners, as well as temporal and modal features. Some lexical element disappears, as happens with complementizers which are done away with and substituted by the functional attribute SCOMP or COMP i.e., complement clause - in Italian FCOMP.

From a theoretical point of view, using Prolog and XGs as procedural formalism we stuck on to LFG very closely (see Pereira(1985)) even though we don't use functional equations: as we noted above, the Fusion mechanism can be performed straightforwardly and the Uniqueness Condition respected thanks to Prolog's unification mechanism. Our approach differs from LFG's algorithm basically for dismissing functional equations: however, functional schemata can encode any kind of information in particular annotated f-structures, keeping a clear record of all structural relations intervening between constituents. In particular, long distance dependencies are treated using XGs, since they can easily encode paths from a controller to its controllee, as well as restrictions to prevent "island violations". In this case, we don't rewrite an empty category by means of a rewriting rule, as in LFG: rather, we activate a procedure as in Pereira(1983). Moreover, the bindee or controllee to be bound by its controller or binder is assigned semantic and functional features by its predicate so that semantic compatibility can be checked when required, or else features transmitted to the controller once binding has taken place.

Italian is a highly structurally ambiguous or underdetermined language (see Delmonte, 1985), so that semantic or thematic checking seems necessary at this level: in particular, long distance dependencies activate all kind of functional restrictions available, since they may be used to prevent backtracking which is time-consuming. We use Case, Gender and Person, as well as semantic categories of the bindee whenever available, to restrict the choice of the binder.

It is worth while reminding that f-structures coincide with lexical forms, i.e. a predicate-argument structure paired with a grammatical function assignment; in other words an fname PRED whose fvalue is a lexical form. Usually clause nuclei are the domain of lexical subcategorization, in the sense that they make available to each lexical form the grammatical functions that are subcategorized by that form (see Bresnan, 1982:304). In case also nouns are subcategorized for, the same requirement of coherence and completeness may be applied. Not all nouns however take arguments.

3. C-structure building

In a language like Italian, at least three clause structural organizations are possible:

5. a canonical organization, corresponding to the standard case in which constituents occupy their canonical positions; subjects come in preverbal position, objects and obliques in postverbal positions and adjuncts may alternate in preverbal or postverbal positions - although they may alternate freely also between verb and object NP;

6. an inverted organization, corresponding to presentative constructions in which the subject occupies postverbal inverted position and an expletive may be present, "ci", or an oblique locative may be preposed in the subject place; or else nothing which relates to the arguments of the predicate be present in preverbal position. The latter case being allowed in Italian but not in other languages;

7. a marked organization, corresponding to a complete reversal of constituents, allowed only in Italian, in which the object NP comes in preverbal position and the subject in postverbal position. The subject in this case, might also be an empty category, thus resembling ergative constructions.

Other structures occur with psychic verbs which subcategorize for an open proposition, an infinitival clause as open complement; copulative constructions with a closed tensed or untensed proposition as subject which might be anaphorically controlled by an adjunct PP headed by "for". Also to this lot, belong left dislocation constructions with clitics as topic variables; topicalized impersonal structures, and other constructions.

Even though LFG does not independently provide the tools to build a richer c-structure configuration, we think it highly important to organize c-structure rules for sentence level representation in line with the chomskyan framework: we extended the X-bar system for the syntactic representation of constituency by the introduction of functional major constituents at the following basic levels:

8. CP --> Spec, C'
   C' --> C, IP
   IP --> Spec=NP(subject), I'
   I' --> Inflected Tensed Verb Form, VP

According to this configuration, adjuncts and constituents like wh- words for questions and topicalized NPs, adjoined at sentence level, will be computed at first in a CP constituent and then passed down to the lower level of analysis. This organization of constituency allows for complementizers, i.e. the head of

CP, to be kept separate in C' level so that a nice interaction may be possible, if needed.

When IP is reached, the NP subject or sentential subject should be computed: at this point there are at least two possible parsing strategies to be followed, both theoretically plausible. The former is in line with LFG traditional view that no empty category should be produced unless it is strictly required by language typology. The latter is in line with Chomsky's assumption of the necessity to pose a basic structural or deep structure configuration which is equal for all languages. In the former case no empty subject NP should arise in case the structure to be analysed is an inverted constuction: this is justified by the fact that the Subject NP is actually to be found in inverted VP internal, or VP adjoined position. Since no NP movement is postulated in LFG there would be no possibility to adequately bind the empty category previously generated in preverbal position. Thus, the sentential structure of inverted, presentational constructions corresponds directly to a VP.

In the latter case, the subject position is filled by an empty category and it should be done away with when parsing the actual lexical subject NP in postverbal position. In case we choose the first strategy, this is how the reasoning proceeds with parsing: since Italian freely allows the subject to be left lexically empty, and since we do not want to produce an empty little pro in case the lexical subject is present in postverbal position, the rule for marked presentational IP must be accessed first. In case the sentence has a canonical structure, failure would have to take place in order to start the second rule for canonical IP. The reason to let the presentational structure come first is due to the fact that in case the sentence starts with a lexical NP before the VP (computed at first as subject), a fail is performed very soon. Here we should note exceptions like bare NPs with a head noun homograph with a verb - which is a common case in English - less so in Italian. In case no lexical NP is present, there are still two possibilities: we either have a canonical structure with an empty little pro as subject, or we have a fully inverted structure.

At first we must assume that no subject is available and try to compute an inverted Subject: clearly this might fail, in case the NP computed in the VP is not interpretable as Subject but as Object of the main predicate. However, we take the marked option to be more frequent and less extendible than the other way round: not every verb class may undergo subject inversion, which is not completely free (see Delmonte, 91). And even if it does, there is quite a number of restrictions that may be made to apply to the inverted subject, as to its referential features (definiteness, etc.), which do not apply to the canonical object NP.

As can be easily gathered, the number of drawbacks from the point of view of parsing strategies is quite high: failure requires backtracking to be performed and this might be very heavy, depending mainly on what has been previously computed as inverted Subject. Not to mention the fact that VP rules should be duplicated in part.

As to the second choice, there will be only one general procedure for parsing grammatical sentence structure, which would postulate the existence of a subject position to be filled either by lexical material or by an empty constituent. In other words, in case the sentence starts with a verb we let typologically determined parameters decide whether it is possible to build an empty subject NP or not: in case we are parsing Italian texts this parameter would be active, but in case we are parsing a text belonging to Germanic languages, it would be disactivated. When we generate an empty category in subject position it remains to be decided what to do with it in case a lexical NP in postverbal position is computed, and this is interpreted as the actual Subject function of the sentence, the trace should be discarded.

C-structure building in our parser corresponds to a partial interpretation of each constituent: in fact, when a parse is completed, we assign a structurally determined grammatical function label which could match semantic checking procedures performed when annotated c-structure is built, or it might be rejected as semantically inappropriate, due to selectional restrictions associated to that constituent. Grammatical functions assignment at a c-structure level is required in all cases in which a presentational construction has been parsed: it is just on the basis of the structural position of a given constituent, the postverbal NP, that we know what is the pragmatic import of the entire utterance. And this will be registered only in the grammatical function assigned to one of the arguments of the predicate, which is computed either as Subj_Foc, or Subj_Top according to whether it is an indefinite or definite NP respectively. The empty NP subject is not bound to the actual lexical NP found in inverted position, and it is simply discarded from the final representation. In this way, the annotated c-structure outputted by the parser is cp rewritten as vp, but the postverbal subject is computed with an adequate grammatical function. Backtracking is thus totally eliminated, and there is only one single procedure which applies to all sentential structures.

At the highest level we want to differentiate between direct speech and other utterances, which are all called by the rule standard_utterance. Only simplex utterances are accepted here and not complex utterances. A simple utterance can either be started by the SPEC of CP containing a ±wh element, i.e. it can be a question, a topicalization or a left dislocation, or a yes-no question. These are fairly general rules applying to all languages: there is a call to adjuncts at cp level, and a call to aux-to-comp elements which however is typologically restricted. It applies to Germanic languages in particular, where auxiliaries may be computed in comp position, as will be discussed below in more detail. In case the call to canonical structures fails, we try topicalized and dislocated constructions.

The first of these calls, is a call to impersonal SI reverse constructions which are usually associated to passive voice. Then we have reverse constructions with transitive verbs which may have the object in sentence initial position: this NP cannot be used to trigger Agreement with the Verb, and must be taken at Top level. Two possibilities exist now: in the first case, we have a typical left dislocation construction, which has the following essential structure: NP Object, NP Subject, resumptive clitic, VP structure, and may be exemplified by the sentence,

1."Il libro Gino lo ha comprato"/The book John it has bought.

In the second case, left dislocation is accompanied by subject inversion, i.e. the essential structure, NP Object, resumptive clitic, tensed verb, NP subject, as in the following example,

2."Il libro lo ha comprato Gino"/The book it has bought.

Thus, when a clitic is present and the Subject is in inverted postverbal position, this is captured by the rule where the topicalized Object NP is linearly followed by a clitic which has accusative case, and no intervening lexical NP can be computed.

From this structural level, either a VP could be straightforwardly computed, or else, an empty NP Subject be postulated and then discarded. We prefer the first option since from structural representation we can already tell that the subject must be empty, owing to the presence of an object clitic. In the former case, the clitic is present but the SUBJect is in preverbal position. Or else, which is the option available in all languages, as in

3."Ski John loves",

we have a Topicalization or focalization, i.e. the OBJect is in Top CP, and the SUBJect in preverbal position. No clitic appears. This is achieved partly by constituent check when building annotated c-structure, and partly by Interpretation at sentence level, when all constituents have been recovered and constructed. The presence of a bound clitic for clitic left dislocation, or else the absence of a clitic and the type of constituent can now be adequately dealt with respectively, as a case of left clitic dislocation with subject focalization in the first case, left clitic dislocation in the second and topicalization in the third case. In the former case, the inverted subject will be interpreted as Foc; in the latter case the preposed object will be interpreted as Top; and in the third case the preposed object as Foc. Notice also that no lexical subject might be present, thus resulting in a simple clitic left dislocated structure with an empty NP subject.

It is interesting to note that all this will follow independently by letting the adequate structure building and constituent check at VP level. After cp has been correctly built, we activate the call to ip where subject NP and negation may be parsed; then a call to i_one_bar, will activate calls to Clitics and Infl, for all inflected verbal forms. The call to Clitics, is allowed both for German and Italian; it also applies exceptionally to English "there", provided no NP subject has been analyzed. Infl is a call which is specialized for different languages and the subsequent typologically marked constructions of Italian.

Parsing the appropriate VP structure requires the instantiation of the appropriate syntactic verb class of the main predicate: in this case, it may either belong to the class of psychic or copulative verbs. Theoretically speaking, c-structure is now represented with a verbal phrase which contains no verb, which has been raised to infl, in case it is a tensed finite verb. In order to let the analysis enter the call for inchoativized verb_phrase, aspectual class is needed; in addition, Subject NP should be an empty pro, in Italian.

All subject inverted constructions at VP level, are constrained by a check on the subject NP: it must be an empty category. This check also applies to impersonal-si constructions and to dislocated constructions. In this way, no backtracking will be allowed. In addition, syntactic category of the main verb should always be checked accordingly. In particolar, inchoative constructions and impersonal-si constructions are also typologically marked, since they are only allowed in Romance languages; also fully inverted transitive constructions and intransitive reflexive structures are only present in Romance languages. The call to intransitive verbal phrases is subsequently further split into the four syntactic classes: {atmospheric, unaccusative, inergative, impersonal}. Transitive structures are differentiated according to the complement type: i.e. adverbial objects require a separate treatment owing to differences in the interpretation of its NP, see

4."John spent three days in Venice"

5."Mary weighs 45 kilos"

and so on. Transitive verbs with open complements are also special in that their object is nonthematic and is interpreted in the open complement, see verbs like

6."believe John stupid"

7."see Mary in the shower",

"consider" and so on. The presence of syntactic classes in verbal entries listed in the lexicon is used as a filter in the construction of VP might be regarded as redundant information, but from a computational point of view it turns out to be a very powerful tool. This is especially so, seen that Italian verbs select auxiliaries according to syntactic class! In particular, unaccusatives require "essere/be" and unergatives "avere/have".

The rule for copulative VPs starts by checking whether a "lo" clitic has been found, in that case this will constitute the open complement, as in

8."Gino lo è" = John it is (happy),

where "lo" is the resumptive invariable clitic for open complements in Italian. In case another clitic has been computed, this can only be treated as a complement or adjunct of the open complement, and is consequently included as first element in the list of constituents passed onto the open complement call. The XCOMP call can be instantiated with any of the allowable lexical heads X=P,A,N,V,Adv, and its associated main constituents. Finally, there is a check on the specifier and referentiality of the preverbal NP computed: in case it is a deictic pronoun, or the Xcomp is a proper noun, this structure will be locally computed as inverted structure as appears in sentences like:

9.The murdered is John,

10.This is a spy story.

Here below we list some of the higher rules of the grammar with one of the interpretation rules for copulative constructions:

utterance --> assertion_direct

```
utterance          --> standard_utterance
standard_utterance--> wh_question
standard_utterance--> yes_no_question
standard_utterance--> assert_cp

assert_cp-->    aux_to_comp
                adjunct_cp
                i_double_bar
assert_cp-->    object
                adjunct_cp
                pro=SI
                verb_phrase_impersonal
assert_cp-->    object
                adjunct_cp
                negat
                pro=CLI, {Case=acc}
                verb_phrase_focalized

assert_cp-->    object
                adjunct_cp
                i_double_bar

i_double_bar-->        subject
                negat
                adjs_preverbal
                parenthetical
                i_one_bar

i_one_bar--> verb_phrase_pass_canonic
i_one_bar--> clitics,
               { germanic_aux,
                 clitics,
                 adjs_post_aux,
                 germanic_vp  ;
                all_languages_vp }

verb_phrase_copulative-->   adv_phrase
                            check_clitic_object
                            xcomp
                            prepositional_phrases
interpret_copulative:-
          lexical-form&             predicate-argument_structure
           interpret_subject
           interpret_xcomp
           assign_control_xcomp
           interpret_adjuncts
```

Notice that i_one_bar rewrites as passive VP and in case of failure as active VP: again this is required by the need to activate the appropriate interpretation rule for transitive verb which in most languages is morphologically determined by the presence of the appropriate auxiliary/ies and the past participle of the main verb. In this way also the Inflection rule is kept separate from that used for active verbs, which is complicated by the presence of germanic languages: in case an auxiliary has already been taken at CP level, it will have to be copied down in the following VP structure to build the adequate verbal compound.

4. Lookahead and the WFST

Lookahead is used in a number of different ways: it may impose a wait-and-see policy on the topdown strategy or it may prevent following a certain rule path in case the stack does not support the first or even second match:

1. to prevent expanding a certain rule
2. to prevent backtracking from taking place by delaying retracting symbols from stack until there is a high degree of confidence in the analysis of the current input string.

It can be used to gather positive or negative evidence about the presence of a certain symbol ahead: symbols to be tested against the input string may be more than one, and also the input word may be ambiguous among a number of symbols. In addition, since in some cases we extend the lookahead mechanism to include two symbols and in one case even three symbols, the possibilities are quite numerous.

Consider now failure and backtracking which ensues from it. Technically speaking, by means of lookahead we prevent local failures in that we do not allow the parser to access the lexicon where the input symbol would be matched against. It is also important to say that all our rules satisfy the requirement to have a preterminal in first position in their right-hand side - almost all rules. There are in fact some wellknown exceptions: simple declarative sentence rule, yes-no questions in Italian. Noun phrase main constituents have a multiple symbols lookahead, adjectival phrase has a double symbol lookahead, adverbial phrase has some special cases which require the match with a certain word/words like "time/times" for instance. Prepositional phrase requires a single symbol lookahead; relative clauses, interrogative clauses, complement clauses are all started by one or more symbols. Cases like complementizerless sentential complements are allowed to be analysed whenever a certain switch is activated.

Suppose we may now delimit failure to the general case that may be described as follows:

- a constituent has been fully built and interpreted but it is not appropriate for that level of attachment:

failure would thus be caused only by semantic compatibility tests required for modifiers and adjuncts or lack of satisfaction of argument requirements for a given predicate.

Technically speaking we have two main possibilities:

A. the constituent built is displaced on a higher level after closing the one in which it was momentarily embedded.

This is the case represented by the adjunct PP "in the night" in the example:

11. The thieves stole the painting in the night.

The PP is at first analysed while building the NP "the painting in the night" which however is rejected after the PP semantic features are matched against the features of the governing head "painting". The PP is subsequently stored on the constituent storage (the WFST, or wellformed substring table) and recovered at the VP level where it is taken as an adjunct.

B. the constituent built is needed on a lower level and there is no information on the attachment site.

In this case a lot of input string has already been consumed before failure takes place and the parser needs to backtrack a lot before constituents may be safely built and interpreted.

Consider the following example taken from one of our texts,

12. Al seguito di Alberti, che era diventato vicepresidente del senato, Franco Avveduti nello immediato dopoguerra si trasferì a Roma.

which might be translated roughly as follows,

"At the suite of Alberti, who has become vicepresident of the Senate, Franco Avveduti in the immediate post-war transferred himself to Rome".

The preposed adjunct PP, modified by a nonrestrictive relative clause, is followed by a noun phrase "Franco Avveduti" which is tentatively computed as apposition of the noun "Senate"; in turn, the following PP headed by "in" is also computed in the wrong place, after taking "Franco Avveduti". In other words, before starting to close any constituent and to interpret it, the parser is situated on the verb "transferred" which causes a local failure, the closing of the PP "in the immediate post-war", and the attempt at interpreting it as adjunct modifier of "Avveduti" a proper noun which is the nonlocal failure causing backtracking.

Consider now what has happened on the lookahead stack: the parser has taken all input symbols, 16 actually, and is now looking at the 17th word "transferred" which is a "verb - v" - in the original version the clitic pronoun "si" also counts as a "v". When backtracking, the length of the first constituent is correctly computed against the lookahead symbol which is still available on top of stack. However, all remaining constituents need the information locally both of starting place and ending place in the input string in order to compute their length.

Consider also the fact that in this situation, backtracking takes the parser back in the input string only up to the point in which there has been a wrong attachment, i.e. after the second comma at the end of the nonrestrictive relative clause. This is so simply because the remaining part of the analysis is correctly interpreted in the position in which it has been analysed as clause initial adjunct PP.

As a result of this situation, there is a mismatch between the input string position of the parser - in front of "Franco", the position of the lookahead stack - as follows, "18-[v-transferred]" - and the constituents on stack for WFST, which has all the NP's and PP's intervening between the verb "transferred" and the first word in the input string "at". Each constituent is built as a term which has the first word as functor or predicate and in its internal representation it has the preterminal symbol associated with the terminal on the stack, then a number indicating the position in the string, another number indicating the length of the constituent built, and finally the constituent itself. The input string is internally represented schematically as follows:

13. LOOKAHEAD-STACK

```
0-[p-al]          1-[n-seguito]   2-[p-di]
        3-[n-alberti]
4-[x-,] 5-[c-che]       6-[v-era]
7-[q-diventato]         8-[n-vicepresidente]
9-[d-del, p-del]        10-[n-senato]   11-[x-,]
12-[n-franco]  13-[n-avveduti]          14-[p-
nello]
15-[a-immediato]        16-[n-dopoguerra]
17-[v-si]               18-[v-trasferì]
19-[p-a]                20-[n-roma]     21-[x-.]
```

14. INPUT TERMINAL SYMBOLS

```
- a(3, 2-pp(PP))        il(2, 2-np(NP))
- di(2, 4-pp(PP))       alberti(1, 4-np(NP))
- vicepresidente(1,9-np(NP))
- di(2,11-pp(PP))       il(2, 11-np(NP))
- franco(2, 14-np(NP))          in(3,          17-
pp(PP))
- lo(3, 17-np(NP))
```

We use the following preterminals on the lookahead stack:

15. PRETERMINAL SYMBOLS

1. v=verb-auxiliary-modal-clitic-cliticized verb
2. n=noun; 3. c=complementizer
4. s=subordinator; 5. e=conjunction

6. p=preposition-particle
7. a=adjective; 8. q=participle/gerund
9. i=interjection 10. g=negation
11. d=article-quantifier-number-intensifier-focalizer
12. r=pronoun 13. b=adverb 14. x=punctuation

The lookahead procedure is used both in presence and in absence of certain local requirements for preterminals, but always to confirm the current choice and prevent backtracking from taking place. As a general rule, one symbol is sufficient to take the right decision; however in some cases, more than one symbol is needed. In particular when building a NP, the head noun is taken at first by nominal premodifiers, which might precede the actual head noun of the NP. The procedure checks for the presence of a sequence of at least two nouns before consuming the current input token. In other cases the number of preterminals to be checked is three, and there is no way to apply a wait-and-see policy.

The following procedure is used to disambiguate multiple preterminals assigned to the same word in order to prevent backtracking. The procedure specifies a set of possible disambiguating categories (Cats) which might follow the current symbol (Y). This procedure is used for an Italian word like "decise" which might be computed both as a verb, thus meaning "decided", past tense, third person singular, or past participle, plural feminine; or else as an adjective, plural feminine.

```
disambiguate_tok(Y,Cats):-
agg(Agg,Gen,Num) --> [X],
  {gr(romance),agg_x(X,Agg,Gen,Num),
      disambiguate_tok(a,[p,c]),
      retraction(agg_x,termin(S-Z))
```

The information gathered from the search on the top of stack also makes available the current location which is then used to compute the length of the PP constituent to be written on the PP stack.

After having analyzed a modal verb Aux, we look for an auxiliary verb, Aux1, in order to support the rule for compound auxiliary which makes up a passive verb in germanic languages: retraction of the modal takes place after the auxiliary is found on the stack. Otherwise backtracking is allowed.

```
auxil_comp(aux(Ainf,Mood,Tense/Ainf1/Ai
nf,Pers,Num,Gen)) -->
  {gr(germanic)},
     [Aux],
modal(Aux,shall,indic,pres,Pers,Num),
[Aux1],
{ aux(Aux1,Ainf,inf,_,Pers,Num)},
(Head=have;Head=be),
extract_sec_head(v, Head, I),
retraction(modal, v).
```

5. References

Bresnan J. (1982), Control and Complementation, in J.Bresnan(ed.*), **The Mental Representation of Grammatical Relations***, The MIT Press, Cambridge Mass.

Bresnan J. (ed) (1982*), **The Mental Representation of Grammatical Relations***, MIT Press, Cambridge Mass.

Bresnan J. (1990), Invited Speech, GLOW, Cambridge.

Bresnan J.; Per-K.Halvorsen; J.Maling (1985), Logophoricity and bound anaphora, MS, Stanford University.

Bresnan J., Mchombo J.M. (1987), Topic, Pronoun and Agreement in Chichewa, *Language* 63, 4, 741-782.

Bresnan J., Kanerva F. (1989), Locative Inversion in Chichewa: A Case Study of Factorization in Grammar, *Linguistic Inquiry* 20, 1, 1-50.

Bresnan J., A.Zaenen (1990), Deep Unaccusativity in LFG, ***Proceedings Fifth Biennal Conference on Grammatical Relations***, San Diego.

Chomsky N. (1986), *Barriers*, MIT Press, Cambridge, Mass.

Delmonte R. (1985), Parsing Difficulties & Phonological Processing in Italian, *Proceedings of the 2nd Conference of the European Chapter of ACL*, Geneva, 136-145.

Delmonte R., R.Dolci(1989), Parsing Italian with a Context-Free Recognizer, ***Annali di Ca' Foscari XXVIII***, 1-2,123-161.

Delmonte R. (1990), Semantic Parsing with an LFG-based Lexicon and Conceptual Representations, *Computers & the Humanities*, 5-6.

Delmonte R. (1991), Empty Categories and Functional Features in LFG, *Annali di Ca' Foscari*, XXX, 1-2, 79-140.

Delmonte R. (1992), *Linguistic and Inferential Processing in Text Analysis by Computer*, Unipress "Studi Linguistici Applicati", Padova.

Delmonte R. (1993), GETA_RUN: A fully integrated system for Reference Resolution by Contextual Reasoning from Grammatical Representations, *ACL-93, Exhibitions and Demonstrations*, Columbus, 1993, p. 2.

Delmonte R., Bianchi D. (1991), Binding Pronominals with an LFG-Based Parser, *II° IWTP91*, Cancun.

Delmonte R., Bianchi D. (1992), Quantifiers in Discourse, in *Proc. ALLC/ACH'92*, Oxford(UK), OUP, 107-114.

Delmonte R., D.Bianchi, E.Pianta (1992), GETA_RUN - A General Text Analyzer with Reference Understanding, in *Proc. 3rd Conference on Applied Natural Language Processing, Systems Demonstrations*, Trento, ACL-92, 9-10.

Delmonte R., R. Dolci, (1996), Parsing Strategies

Pereira F. (1981), Extraposition Grammars, *American Journal of Computational Linguistics* 7, 4, 243-256.

Pereira F. (1983), *Logic for Natural Language Analysis*, Technical Note 275, Artificial Intelligence Center, SRI International.

Pereira F.(1985), A Structure-Sharing Representation for Unification-Based Grammar Formalism, in *Proceedings of the 23rd Annual Meeting of the ACL,* Chicago, 137-144.