GETARUN PARSER
A parser equipped with Quantifier Raising and Anaphoric Binding based on LFG

Rodolfo Delmonte
Ca' Garzoni-Moro, San Marco 3417
Università "Ca Foscari"
30124 - VENEZIA
Tel. 39-41-2578464/52/19 - Fax. 39-41-5287683
E-mail: delmont@unive.it - website: project.cgm.unive.it

Proceedings of the LFG02 Conference

National Technical University of Athens, Athens

Miriam Butt and Tracy Holloway King (Editors)

2002

CSLI Publications

http://csli-publications.stanford.edu/

GETARUN PARSER

Rodolfo Delmonte

Abstract

GETARUN, the system for text understanding developed at the University of Venice, is equipped with three main modules: a lower module for parsing where sentence strategies are implemented; a middle module for semantic interpretation and discourse model construction which is cast into Situation Semantics; and a higher module where reasoning and generation takes place. We assume that from a psycholinguistic point of view, parsing requires setting up a number of disambiguating strategies, basically to tell arguments apart from adjuncts and reduce the effects of backtracking. The system is based on LFG theoretical framework and has a highly interconnected modular structure. It is a top-down depth-first DCG-based parser written in Prolog which uses a strong deterministic policy by means of a lookahead mechanism with a WFST to help recovery when failure is unavoidable due to strong attachment ambiguity. It is divided up into a pipeline of sequential but independent modules which realize the subdivision of a parsing scheme as proposed in LFG theory where a c-structure is built before the f-structure can be projected by unification into a DAG. As to multilinguality, the basic tenet of the parser is based on a UG-like perspective, i.e. the fact that all languages share a common core grammar and may vary at the periphery: internal differences are taken care of by parameterized rules. The DCG grammar allows the specification of linguistic rules in a highly declarative mode: it works topdown and by making a heavy use of linguistic knowledge may achieve an almost complete deterministic policy. Parameterized rules are scattered throughout the grammar so that they can be activated as soon as a given rule is entered by the parser.

1. Introduction

GETARUN, the system for text understanding developed at the University of Venice, is equipped with three main modules: a lower module for parsing where sentence strategies are implemented (Delmonte, 1990; Delmonte and Bianchi and Pianta, 1992); a middle module for semantic interpretation and discourse model construction which is cast into Situation Semantics; and a higher module where reasoning and generation takes place (Delmonte, 2000a; Delmonte and Bianch, 2002).

We assume that from a psycholinguistic point of view, parsing requires setting up a number of disambiguating strategies, basically to tell arguments apart from adjuncts and reduce the effects of backtracking.

The system is based on LFG theoretical framework (see Bresnan, J. 2001) and has a highly interconnected modular structure. It is a top-down depth-first DCG-based parser written in Prolog which uses a strong deterministic policy by means of a lookahead mechanism with a WFST to help recovery when failure is unavoidable due to strong attachment ambiguity.

It is divided up into a pipeline of sequential but independent modules which realize the subdivision of a parsing scheme as proposed in LFG theory where a c-structure is built before the f-structure can be projected by unification into a DAG. In this sense we try to apply in a given sequence phrase-structure rules as they are ordered in the grammar: whenever a syntactic constituent is successfully built, it is checked for semantic consistency, both internally for head-spec agreement, and externally, in case of a non-substantial head like a preposition dominates the lower NP constituent; other important local semantic consistency checks are performed with modifiers like attributive and predicative adjuncts. In case the governing predicate expects obligatory arguments to be lexically realized they will be searched and checked for uniqueness and coherence as LFG grammaticality principles require.

Whenever a given predicate has expectancies for a given argument to be realized either optionally or obligatorily this information will be passed below to the recursive portion of the parsing: this operation allows us to implement parsing strategies like Minimal Attachment, Functional Preference and other ones (see Delmonte and Dolci 1989; Delmonte and Dolci 1997).

As to multilinguality, the basic tenet of the parser is based on a UG-like perspective, i.e. the fact that all languages share a common core grammar and may vary at the periphery: internal differences are predicted by parameters. The DCG grammar allows the specification of linguistic rules in a highly declarative mode: it works topdown and by making a heavy use of linguistic knowledge may achieve an almost complete deterministic policy. Parameterized rules are scattered throughout the grammar so that they can be made operative as soon as a given rule is entered by the parser.

In particular, a rule may belong either to a set of languages, e.g. Romance or Germanic, or to a subset thereof, like English or Italian, thus becoming a peripheral rule. Rules are activated at startup and whenever a switch is being operated by the user, by means of logical flags appropriately inserted in the right hand side of the rule. No flags are required for rules belonging to the common core grammar.

Some such rules include the following ones: for languages like Italian and Spanish, a Subject NP may be an empty category, either a referential little pro or an expletive pronoun; Subject NPs may be freely inverted in postverbal position, i.e. preverbal NP is an empty category in these cases. For languages like Italian and French, PP or adverbial adjuncts may intervene between Verb and Object NP; adjectival modifiers may be taken to the right of their head Noun. For languages like English and German, tense and mood may be computed in CP internal position, when taking the auxiliary or the modal verb. English allows an empty Complementizer for finite complement and relative clauses, and negation requires do-support. Italian only allows for a highly genre marked (literary style) untensed auxiliary in Comp position.

Syntactic and semantic information is accessed and used as soon as possible: in particular, both categorial and subcategorization information attached to predicates in the lexicon is extracted as soon as the main predicate is processed, be it adjective, noun or verb, and is used to subsequently restrict the number of possible structures to be built. Adjuncts are computed by semantic compatibility tests on the basis of selectional restrictions of main predicates and adjuncts heads.

Syntactic rules are built using CP-IP functional maximal projections. Thus, we build and process syntactic phenomena like wh- movement before building f-structure representations, where quantifier raising and anaphoric binding for pronominals takes place. In particular, all levels of Control mechanisms which allow coindexing at different levels of parsing give us a powerful insight into the way in which the parser should be organized.

Yet the grammar formalism implemented in our system is not fully compliant with the one suggested by LFG 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. On the other hand, DCGs being based on Prolog language, give 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. In addition, we find that topdown parsing policies are better suited to implement parsing strategies that are essential in order to cope with attachment ambiguities (but see below).

We use 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 chosen languages. 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.

Being a DCG, the parser is strictly a top-down, depth-first, one-stage parser with backtracking: differently from most principle-based parsers presented in Berwick et al.(1991), which are two-stage parsers, our parser computes its representations in one pass. This makes it psychologically more realistic. The final output of the parsing process is an f-structure which serves as input to the binding module and logical form: in other words, it constitutes the input to the semantic component to compute logical relations. In turn the binding module may add information as to pronominal elements present in the structure by assigning a controller/binder in case it is available, or else the pronominal expression will be available for discourse level anaphora resolution. As to the most important features of DCGs, we shall quote from Pereira and Warren(1980) conclusions, in a comparison with ATNs:

"Considered as practical tools for implementing language analysers, DCGs are in a real sense more powerful than ATNs, since, in a DCG, the structure returned from the analysis of a phrase may depend on items which have not yet been encountered in the course of parsing a sentence. ... Also on the practical side, the greater clarity and modularity of DCGs is a vital aid in the actual development of systems of the size and complexity necessary for real natural language analysis. Because the DCG consists of small independent rules with a declarative reading, it is much easier to extend the system with new linguistic constructions, or to modify the kind of structures which are built. ... Finally, on the philosophical side, DCGs are significant because they potentially provide a common formalism for theoretical work and for writing efficient natural language systems."(ibid,278).

1.1 Implementing LFG theory

As said above, there are a number of marked differences in the treatment of specific issues, concerning Romance languages, which are not sufficiently documented in the linguistic literature (however see Bresnan, 2001). In particular,

- we introduced an empty subject pronominal - little pro - for tensed propositions, which has different referential properties from big PRO; this has 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 (MUS). In turn MUS or little pros could be computed as variables in case the subject was realized in postverbal position. At the beginning, LFG posited the existence of a rule for sentence structure which could be rewritten as VP both in case there was no subject, and in case the subject was expressed in postverbal position, an approach that we did not implement;
- as to functional constituents: 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(but see below);
- 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 for the assignment of 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, which are used to compute tense/aspect structural representations;
- 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, we generate a fully annotated c-structure representation which is then checked for Grammatical Principles Consistency at the level of number/type of arguments and of Adequacy for adjuncts, on the basis of lexical form of each predicate and semantic consistency crossed checks for adjuncts.

1.2 Disambiguating constituency with functional mapping

As shown in Fig.2 below, 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, i.e. constituent information is dropped and DAGs are built in order to produce f-structure configuration.

Mapping into f-structure is a one-to-many operation: 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.

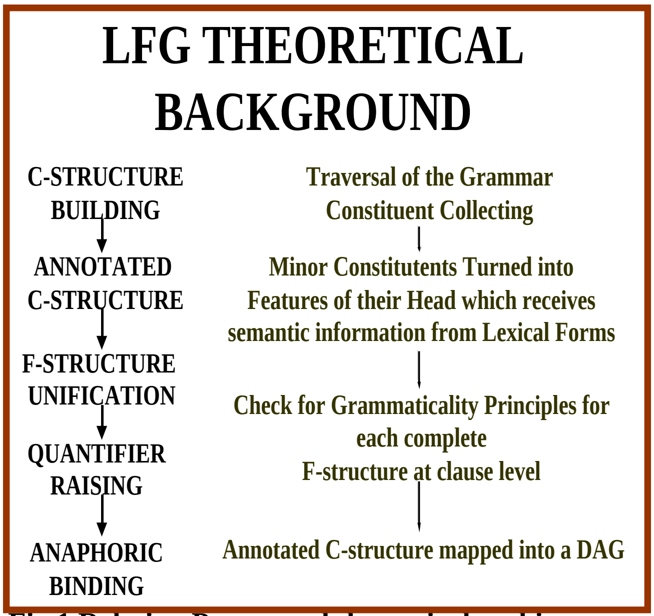
1.3 Quantifier Raising

Since we know that quantifiers and quantified NPs usually take scope at propositional and NP level, we assume f-structure to be an adequate level of representation in which quantifer scope can be computed. In this we partially follow Halvorsen’s proposal (see Halvorsen; Halvorsen & Kaplan), which however requires a further mapping from f-structures to s-structures in order to do that. We proceed as follows: after assigning Q-Markers to quantifiers and quantified NPs and adding this information as attribute-value pair at f-structure, we perform Quantifier Raising by traversing f-structure until we reach a propositional node. At that level we deposit a Quantifier-Operator(Q-Op), in an attribute that has a list as its value. Once Q-Ops have been produced, we are in a position to assign quantifier scope. In case more than one Q-Op is present in the list, the algorithm simply reorders the operators according to their quantifying force, and/or to grammatical function. Otherwise, a search downward is performed in the f-structure for other q-ops. When some q-marker is found another attribute-value pair is added at pred level indicating a quantified interpretation. We could duplicate the procedure at NP level by taking into account all NP modifiers in case they are quantified (but see Delmonte & Bianchi, 1992; Delmonte 1997; Dibattista et al. 1999).

1.4 The Binding Module

The output of grammatical modules is fed then onto the Binding Module(BM) which activates an algorithm for anaphoric binding in LFG terms using f-structures as domains and grammatical functions as entry points into the structure. Pronominals are internally decomposed into a feature matrix which is made visible to the Binding Algorithm(BA) and allows for the activation of different search strategies into f-structure domains. Antecedents for pronouns are ranked according to grammatical function, semantic role, inherent features and their position at f-structure. Special devices are required for empty pronouns contained in a subordinate clause which have an ambiguous context, i.e. there are two possible antecedents available in the main clause. Also split antecedents trigger special search strategies in order to evaluate the possible set of antecedents in the appropriate f-structure domain. Special care is paid to pronominals bound by quantifiers or quantified NPs in order to detect crossover violations. The output of the BA is then passed on to an Interpretation Module which operates locally in order to spot the presence of conditions for Specific or Arbitrary Reading for pronominal expressions (see Delmonte & Bianchi, 1991). Eventually, this information is added into the original f-structure graph and then passed on to the Discourse Module(DM) (see Delmonte & Bianchi, 1999; Delmonte 2002).
In Fig.1 We show the interrelations existing between the parser architecture as it has been described so far and the LFG theoretical background. The different

levels of representations required by LFG are coupled with modules and processes of the parser.



LFG THEORETICAL BACKGROUND

C-STRUCTURE BUILDING

ANNOTATED C-STRUCTURE

F-STRUCTURE UNIFICATION

QUANTIFIER RAISING

ANAPHORIC BINDING

Traversal of the Grammar Constituent Collecting

Minor Constitutents Turned into Features of their Head which receives semantic information from Lexical Forms

Check for Grammaticality Principles for each complete F-structure at clause level

Annotated C-structure mapped into a DAG

Fig.1 Relating Parser and theoretical architectures

2. Tracing c-structure rules

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. In addition to that there are all lexical form provided by a fully revised version of COMLEX, and in order to take into account phrasal and adverbial verbal compound forms, we also use lexical entries made available by UPenn and TAG encoding. Their grammatical verbal syntactic codes have then been adapted to our formalism and is used to generate an approximate subcategorization scheme with an approximate aspectual and semantic class associated to it. Semantic inherent features for Out of Vocabulary Words , be they nouns, verbs, adjectives or adverbs, are provided by a fully revised version of WordNet in which we used labels similar to those provided by CoreLex.

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). 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.

Eventually, structural information is 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.

As said above, we think it highly important to organize c-structure rules for sentence level representation by means of the introduction of functional major constituents at the following basic levels:

```
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.

However, rule ordering may clash with the need to produce the most adedquate structure as soon as possible without incurring in an inefficient andc psychologically unmotivated backtracking.

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 construction: 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 erased 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 – i.e. preposed Object NP followed by postverbal Subject NP.

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 function 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: in case of reverse constructions, these 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. In some cases, however, interpretation will folow from the actual structural representation rather than from lexical information alone: this is the case of all non subcategorized cases of secondary predication, like resultative structures and other cases of attributive open complements. 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:

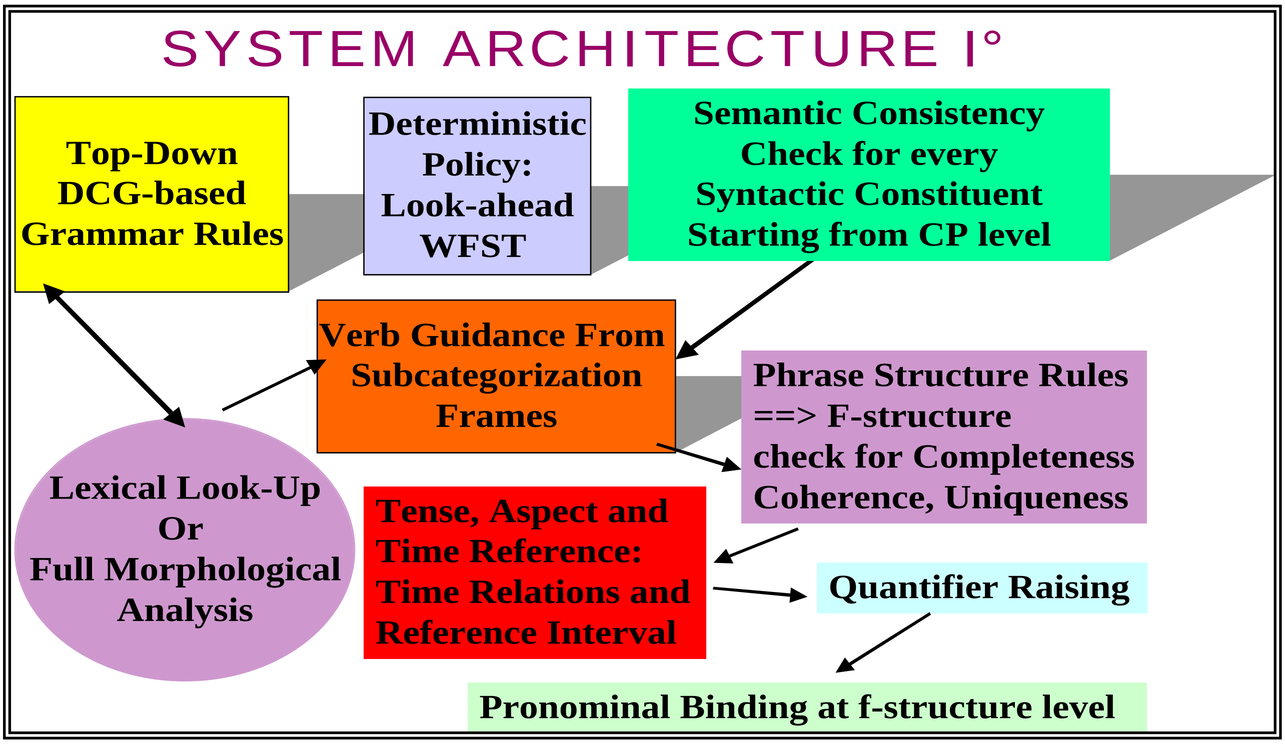
```
utterance          --> assertion_direct
utterance          --> standard_utterance
standard_utterance-->  wh_question
standard_utterance-->  yes_no_question
standard_utterance-->  assert_cp

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.

Fig.2 GETARUN Parser Architecture



SYSTEM ARCHITECTURE I°

Top-Down DCG-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

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.

3. Mapping Functional Structures to an XML output

Lately, the output of the parser, which is computed as a DAG, i.e. a list of interconnected arcs, nodes and leaves, has been ported to an XML format in order to be viewed in any browser. We decided to treat f-structures and subsidiary f-structures as arcs which contain nodes consisting of features which are attribute-value pairs. A distinctive node has been preserved for predicates, which are called Semantic Words <sw>. We also assigned a special structure to tense/aspect structural nodes which are then treated at the same level of higher f-structures and assigned to an arc. We show here below a simple example based on the sentence "Did Mary like John?"

```
f([did, mary, like, john, ?], es34).
rete(es34).
leaf(n2, f1, es34).
leaf(n3, yes/no_question, es34).
leaf(n4, like, es34).
leaf(n5, [sn/sogg/esperiente/[umano, animato], sn/ogg/tema_emot/
[stato, umano, animato, oggetto]], es34).
leaf(n6, active, es34).
leaf(n7, ind, es34).
leaf(n8, past, es34).
leaf(n9, emotivo, es34).
leaf(n11, sn1, es34).
leaf(n12, [umano], es34).
leaf(n13, mary, es34).
leaf(n15, fem, es34).
leaf(n16, sing, es34).
leaf(n17, 3, es34).
leaf(n19, 0, es34).
leaf(n21, sn2, es34).
leaf(n22, [umano], es34).
leaf(n23, john, es34).
leaf(n25, mas, es34).
leaf(n26, sing, es34).
leaf(n27, 3, es34).
leaf(n29, 0, es34).
leaf(n14, [+ref, -pro, -ana, -class], es34).
leaf(n24, [+ref, -pro, -ana, -class], es34).
leaf(n30, stato, es34).
leaf(n31, [tr(f1_es34)<td(f1_es34)], es34).
leaf(n32, [tr(f1_es34)=tes(f1_es34)], es34).
leaf(n33, -, es34).
leaf(n34, [tr(f1_es34)], es34).
arc(n1, n2, indice, es34).
arc(n1, n3, perf, es34).
arc(n1, n4, pred, es34).
arc(n1, n5, lex_form, es34).
arc(n1, n6, voice, es34).
arc(n1, n7, modo, es34).
arc(n1, n8, tempo, es34).
arc(n1, n9, cat, es34).
arc(n1, n10, sogg/esperiente, es34).
arc(n10, n11, indice, es34).
arc(n10, n12, cat, es34).
arc(n10, n13, pred, es34).
arc(n10, n15, gen, es34).
arc(n10, n16, num, es34).
arc(n10, n17, pers, es34).
arc(n10, n18, spec, es34).
arc(n18, n19, def, es34).
arc(n1, n20, ogg/tema_emot, es34).
arc(n20, n21, indice, es34).
arc(n20, n22, cat, es34).
arc(n20, n23, pred, es34).
arc(n20, n25, gen, es34).
arc(n20, n26, num, es34).
arc(n20, n27, pers, es34).
arc(n20, n28, spec, es34).
arc(n28, n29, def, es34).
arc(n10, n14, tab_ref, es34).
arc(n20, n24, tab_ref, es34).
arc(n1, n30, aspetto, es34).
arc(n1, n31, rel1, es34).
arc(n1, n32, rel2, es34).
arc(n1, n33, definitezza, es34).
arc(n1, n34, ref_int, es34).
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<text words="did mary like john ?">
<sent init="es34">
<node type= "index" ><leaf  val="f1"> </leaf></node>
<node type= "perf" ><leaf val="yes/no_question">
</leaf></node>
<sw id="sw_167" type="pred" ><leaf  val="like"> </leaf></sw>
<node type= "lex_form" ><leaf  val="[sn/sogg/esperiente/[umano,
animato], sn/ogg/tema_emot/[stato, umano, animato, oggetto]]">
</leaf></node>
<node type= "voice" ><leaf  val="active"> </leaf></node>
<node type= "mood" ><leaf  val="ind"> </leaf></node>
<node type= "tense" ><leaf  val="past"> </leaf></node>
<node type= "cat" ><leaf  val="emotivo"> </leaf></node>
<arc type= "subj/experiencer" ><node type= "index" ><leaf
val="sn6"> </leaf></node>
          <node type= "cat" ><leaf  val="[umano]"> </leaf></node>
          <sw id="sw_168" type="pred" ><leaf  val="mary">
</leaf></sw>
            <node type= "gen" ><leaf  val="fem"> </leaf></node>
            <node type= "num" ><leaf  val="sing"> </leaf></node>
            <node type= "pers" ><leaf  val="3"> </leaf></node>
            <node type= "spec" ><node type= "def" ><leaf
val="'0'"> </leaf></node>
</node>
            <node type= "tab_ref" ><leaf  val="[+ref, -pro, -ana, -
class]"> </leaf></node>
</arc>
<arc type= "obj/emot_theme" ><node type= "index" ><leaf
val="sn16"> </leaf></node>
          <node type= "cat" ><leaf  val="[umano]"> </leaf></node>
          <sw id="sw_169" type="pred" ><leaf  val="john">
</leaf></sw>
             <node type= "gen" ><leaf  val="mas"> </leaf></node>
             <node type= "num" ><leaf  val="sing"> </leaf></node>
             <node type= "pers" ><leaf  val="3"> </leaf></node>
            <node type= "spec" ><node type= "def" ><leaf  val="'0'">
</leaf></node>
</node>
      <node type= "tab_ref" ><leaf  val="[+ref, -pro, -ana, -class]">
</leaf></node>
</arc>
<arc type= "'tense/aspect'" ><leaf val="state"> </leaf>
<node type= "rel1" ><leaf val="[after(tr(f1_es34),
td(f1_es34))]"> </leaf></node>
```

```
    <node type= "rel2" ><leaf  val="[tr(f1_es34)=tes(f1_es34)]">
</leaf></node>
      <node type= "definiteness" ><leaf  val="-"> </leaf></node>
      <node type= "ref_int" ><leaf  val="[tr(f1_es34)]">
</leaf></node>
</arc>
</sent>
</text>
```

4. Parsing Strategies and Preferences

The parser has been built to simulate the cognitive processes underlying the grammar of a language in use by a speaker, taking into account the psychological nuances related to the wellknown problem of ambiguity, which is a pervading problem in real text/communicative situation, and it is regarded an inseparable benchmark of any serious parser of any language to cope with.

In order for a parser to achieve psychological reality it should satisfy three different types of requirements: psycholinguistic plausibility, computational efficiency in implementation, coverage of grammatical principles and constraints. Principles underlying the parser architecture should not conform exclusively to one or the other area, disregarding issues which might explain the behaviour of the human processor. In accordance with this criterion, we assume that the implementation should closely mimick phenomena such as Garden Path effects, or an increase in computational time in presence of semantically vs. syntactically biased ambiguous structures. We also assume that a failure should ensue from strong Garden Path effects and that this should be justified at a psycholinguistic interpretation level (see Pitchett.

Differently from what is asserted by global or full paths approaches (see Schubert, 1984; Hobbs et al., 1992), we believe that decisions on structural ambiguity should be reached as soon as possible rather than deferred to a later level of representation. In particular, Schubert assumes "...a full paths approach in which not only complete phrases but also all incomplete phrases are fully integrated into (overlaid) parse trees dominating all of the text seen so far. Thus features and partial logical translations can be propagated and checked for consistency as early as possible, and alternatives chosen or discarded on the basis of all of the available information(ibid., 249)." And further on in the same paper, he proposes a system of numerical 'potentials' as a way of implementing preference trade-offs. "These potentials (or levels of activation) are assigned to nodes as a function of their syntactic/semantic/pragmatic structure and the preferred structures are those which lead to a globally high potential. Other important approaches are represented by Hindle et al., 1993, who attempt to solve the problem of attachment ambiguity in statistical terms (see Delmonte 2000b,2000c,2000d).

Among contemporary syntactic parsing theories, the garden-path theory of sentence comprehension proposed by Frazier(1987a, b), Clifton & Ferreira (1989) among others, is the one that most closely represents our point of view. It works on the basis of a serial syntactic analyser, which is top-down, depth-first - i.e. it works on a single analysis hypothesis, as opposed to other theories which take all possible syntactic analysis in parallel and feed them to the semantic processor. From our perspective, it would seem that parsing strategies should be differentiated according to whether there are argument requirements or simply semantic compatibily evaluation for adjuncts. As soon as the main predicate or head is parsed, it makes available all lexical information in order to predict if possible the complement structure, or to guide the following analysis accordingly. As an additional remark, note that not all possible syntactic structure can lead to ambiguous interpretations: in other words, we need to consider only cases which are factually relevant also from the point of view of language dependent ambiguities.

Parsing theories are of two kinds: the first garden-path theory (hence GPT) is syntactically biased, and the second incremental-interactive theory (hence IIT) is semantically biased. There is a crucial difference between the two theories: whereas GPT claims that a single analysis of a syntactic structure ambiguity is initially constructed and passed to the semantic module, IIT claims that the syntactic module offers all grammatical alternatives to the semantic one, in parallel, to be evaluated. There is evidence that is favourable to both sides on this issue (see G.Altman, 1989).

The basic claims of GPT are that the sentence processing mechanism (the parser) uses a portion of its grammatical knowledge, isolated from world knowledge and other information, in initially identifying the relationships among the phrases of a sentence. IIT permits a far more intimate and elaborate interaction between the syntax and the semantic/referential modules. Altmann(1989) offers a functional argument against a system in which choices are initially made by a syntactic processor, and later corrected by appeal to meaning and context. He says that if referential or discourse information is available, only a strange processor would make decisions without appealing to it. It is also our opinion that all lower level constraints should work concurrently with higher level ones: in our parser all strategies are nested one inside another, where Minimal Attachment (hence MA) occupies the most deeply nested level.

The list of examples here below includes sentences used in the literature to support one or the other of the two parsing theories, GPT and IIT (see Altman (ed), 1989).

11. Mary put the book on the table
12. Mary put the book on the table in her bag
13. Mary saw the cop with the binoculars
14. Mary saw the cop with the revolver
15. The thieves stole the painting in the museum
16. The thieves stole the painting in the night
17. John saw Mary in the kitchen
18. John saw Mary from the bathroom

Steedman & Altmann(1989) note that several of the ambiguities present in these sentences and resolved by Minimal Attachment involve contrasts between NP modification and other structures. However, examples 12, 13 and 15 require some more knowledge, linguistic one.

In example 11 we assume that the PP should be computed as an argument of the main predicate "put", thus following a MA strategy when parsing the NP "the book". However, the same strategy would lead to a complete failure in example 12, where the PP should be taken as np modifier. If we look at subcategorization requirements of the verb, example 1 constitutes a clear case of Verb Guidance and of semantic role satisfaction requirements: the main predicate requires an argument which is a locative, so at every decision point in which a PP might be taken, argument requirements should be accessed and a MA strategy imposed locally by FP.

Example 13 contains an instrumental adjunct: when the head preposition "with" is met, the parser will not close the NP "the cop" and will continue building an internal PP modifier since preposition "with" heads a compatible NP modifier, a comitative. In our dictionary the verb "see" has one single lexical entry but a list containing two different lexical forms. The first form in the list has a higher number of arguments,

I. see <SUB/perceiv, OBJ/theme, PCOMP/locat>

where the Pcomp predicates a location of the Object; and a second form, where the Pcomp is absent. In order for a Location PP to be accepted, the head preposition should be adequate, and "with" does not count as such. In examples 13 and 14, the first decision must be taken when computing NP structure. In fact, a PP headed by preposition "with" is a semantically compatible np modifier - a comitative - and the analysis should be allowed to continue until the PP is fully analysed. In other words the parser should verify PP attachment consistency inside the NP constituent.

In the following examples (15, 16), argument structure plays no role whatsoever: instrumentals, comitatives, locatives with predicate "steal" are all cases of sentential adjuncts, and only semantic criteria can apply. As a matter of fact, "in the museum" might be freely attached lower at NP level as well as higher at Sentence level. This is due to the fact that head preposition "in" constitutes a viable local NP modifier and there are no argument requirements from the main verb predicate. However, "in the night" is not a possible NP modifier and example 16 is a clear case of minimal attachment sentence. On the contrary, in example 15 PP attachment is ambiguous between a np internal modifier and VP level attachment.

In example 17 we understand that the location at which Mary was when the seeing event took place is the kitchen: we also understand that John might have been in the same location or in a different one, already provided by the previous context, and this can be achieved by FP which activates MA and makes the locative PP available at VP level.

In example 18, on the contrary, we understand that the location from the which the seeing event took place is the bathroom and that John was certainly there; however we are given no information whatsoever about Mary's location. This case is treated as the previous one, except that the PP is computed as sentence adjunct rather than as VP complement.

4.1 Two mechanisms at work

We implemented two simple enough mechanisms in order to cope with the problem of nondeterminism and backtracking. At bootstrapping we have a preparsing phase where we do lexical lookup and we look for morphological information: at this level of analysis of all input tokenized words, we create the lookahead stack, which is a stack of pairs input wordform - set of preterminal categories, where preterminal categories are a proper subset of all lexical categories which are actually contained in our lexicon. The idea is simply to prevent attempting the construction of a major constituent unless the first entry symbol is well qualified. The following list of preterminal 14 symbols is used:

19. PRETERMINAL SYMBOLS
1. v=verb-auxiliary-modal-clitic-cliticized verb
2. n=noun – common, proper;
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

As has been reported in the literature (see Tapanainen, 1994; Brants, 1995), English is a language with a high level of homography: readings per word are around 2 (i.e. each word can be assigned in average two different tags). Lookahead in our system copes with most cases of ambiguity: however, we also had to introduce some disambiguating tool before the input string could be safely passed to the parser. Disambiguation is applied to the lookahead stack and is operated by means of Finite State Automata. The reason why we use FSA is simply due to the fact that for some important categories, English has unambiguous tags which can be used as anchoring in the input string, to reduce ambiguity. I'am now referring to the class of determiners which is used to tell apart words belonging to the ambiguity class [verb,noun], the most frequent in occurrence in English.

In order to cope with the problem of recoverability of already built parses we built a more subtle mechanism that relies on Kay's basic ideas when conceiving his Chart(see Kay, 1980; Stock, 1989). Differently from Kay, however, we are only interested in a highly restricted topdown depthfirst parser which is optimized so as to incorporate all linguistically motivated predictable moves. Any already parsed NP/PP is deposited in a table lookup accessible from higher levels of analysis and consumed if needed. To implement this mechanism in our DCG parser, we assert the contents of the structure in a table lookup storage which is then accessed whenever there is an attempt on the part of the parser to build up a similar constituent. In order to match the input string with the content of the stored phrase, we implemented a WellFormed Substring Table(WFST) as suggested by Woods(1973).

Now consider the way in which a WFST copes with the problem of parsing ambiguous structure. It builds up a table of well-formed substrings or terms which are partial constituents indexed by a locus, a number corresponding to their starting position in the sentence and a length, which corresponds to the number of terminal symbols represented in the term. For our purposes, two terms are equivalent in case they have the same locus and the same length.

In this way, the parser would consume each word in the input string against the stored term, rather than against a newly built constituent. In fact, this would fit and suit completely the requirement of the parsing process which rather than looking for lexical information associated to each word in the input string, only needs to consume the input words against a preparsed well-formed syntactic constituent.

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:

a. to prevent expanding a certain rule

b. to prevent backtracking from taking place by delaying retracting symbols from input 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. Since in some cases we extend the lookahead mechanism to include two symbols and in one case even three symbols, possibilities become 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 almost all our rules satisfy the efficiency requirement to have a preterminal in first position in their right-hand side. This is usually related to the property belonging to the class of Regular Languages. 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 example 16 that we repeat here below:

16. 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) 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.

To give a simple example, suppose we have taken the PP "in the night" within the NP headed by the noun "painting". At this point, the lookahead stack would be set to the position in the input string that follows the last word "night". As a side-effect of failure in semantic compatibility evaluation within the NP, the PP "in the night" would be deposited in the backtrack WFST storage. The input string would be restored to the word "in", and analysis would be restarted at the VP level. In case no PP rule is met, the parser would continue with the input string trying to terminate its process successfully. However, as soon as a PP constituent is tried, the storage is accessed first, and in case of non emptiness its content recovered. No structure building would take place, and semantic compatibility would take place later on at sentence level. The parser would only execute the following actions:

- match the first input word with the (preposition) head of the stored term;
- accept new input words as long as the length of the stored term allows it by matching its length with the one computed on the basis of the input words.

As said above, 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.

Reanalysis of a clause results in a Garden Path(GP) in our parser because nothing is available to recover a failure that encompasses clause level reconstruction: we assume that GP obliges the human processor to dummify all naturally available parsing mechanisms, like for instance lookahead, and to proceed by a process of trial-and-error to reconstruct the previously built structure in order not to fall into the same mistake. The same applies to our case which involves interaction between two separate modules of the grammar.

As an example, consider processing time 5.6 secs with strategies and all mechanisms described above activated, as compared to the same parse when the same are disactivated – 17.3 secs, in relation to the following highly ambiguous example taken from Legal Texts in the Appendix:

Producer means the manufacturer of a finished product, the producer of any raw material or the manufacturer of a component part and any person who by putting his name, trade mark or other distinguishing feature on the product presents himself as its producer.

Computation time is calculated on a Macintosh G4.

In more detail, suppose we have to use the information that "put" is a verb which requires an oblique PP be present lexically in the structure, as results from a check in its lexical form. We take the verb in I position and then open the VP complement structure, which at first builds a NP in coincidence with "the book". However, while still in the NP structure rules, after the head has been taken, a PP is an option freely available as adjunct.

We have implemented two lookahead based mechanisms which are used in the PP building rule and are always triggered, be it from a position where we have a noun as head and we already built part of the corresponding constituent structure; be it from a position where we have a verb as head and we want to decide whether our PP will be adequate as argument rather than as adjunct - in the latter case it will become part of the Adjunct Set.

The first one is called,

- Cross Compatibility Check (CCC)

This mechanism requires the head semantic features or inherent features to be checked against the preposition, which in turn activates a number of possible semantic roles for which it constitutes an adequate semantic marker. For instance, the preposition "on" is an adequate semantic marker for "locative" semantic role, this will cause the compatibility check to require the presence in the governing heading of inherent or semantic features that allow for location. A predicate like "dress" is computed as an object which can be assigned a spatial location, on the contrary a predicate like "want" is computed as a subjective intensional predicate which does not require a spatial location. However, in order to take the right decision, the CCC must be equipped with the second mechanism we implemented;

The second one is called,

- *Argument Precedenc (AP)*

The second mechanism we implemented allows the parser to satisfy the subcategorization requirements in any NP constituent it finds itself at a given moment if the parsing process. Suppose that after taking "put" as the main verb, this mechanism is activated, by simply copying the requirements on PP oblique locative present in the lexical form associated with the predicate "put" in the lexicon, in the AP. As soon as the NP "the book" is opened, after taking "book" as N at the head position, the parser will meet the word "on", which allows for a PP adjunct. While in the P head position, the parser will fire the CCC mechanism first to see whether the preposition is semantically compatible, and in case it is, the second AP mechanism will be fired. This will cause the system to do the following steps:

i. check whether the requirements are empty or not;

ii. and in case it is instantiated, to control the semantic role associated with it;

iii. to verify whether the P head is a possible semantic marker for that semantic role: in our case, "on" is a possible semantic marker for "locative" semantic role;

iv. finally to cause the parser to fail on P as head of a PP adjunct of the head noun;

v. produce a closure of NP which obeys Minimal Attachment principle.

4.2 Some examples

In the texts reported in the Appendix there is a great number of examples which can be used as empirical evidence for the need to use lexical information in order to reduce parsing loads resulting from backtracking procedures. We use examples taken from the Legal text included in the Appendix at the end of the paper: we mark decision points with a bar,

20. Council directive | **of** july 1985 | **on** the approximation | **of** the laws, | regulations and | administrative provisions | **of** the Member States | concerning liability | **for** defective products.

At the first boundary we have "of" which is non semantically marked and no prediction is available, so that the default decision is to apply Late Closure, which turns out to be the correct one. When the second preposition is found we are in the NP of the PP headed by "of", and we have taken the date "1985": this will cause the CCC to prevent the acceptance of the preposition "on" as a semantically compatible marker thus preventing the construction of the NP headed by "approximation".

Notice, that in case that would be allowed, the NP would encompass all the following PPs thus building a very heavy NP: "the approximation of the laws, regulations and administrative provisions of the Member States concerning liability for defective products". In case the parser had a structure monitoring strategy all this work would have to be undone and backtracking would have to be performed. Remember that the system does not possibly know where and how to end backtracking unless by trying all possible available combination along the path. In our case, the presence of a coordinate structure would render the overall process of structure recoverability absolutely untenable.

Another important decision has be taken at the boundary constituted by the participial head "concerning": in this case the CCC will take the inherent features of the head "States" and check them with the selection restrictions associated in the lexical form for the verb "concern". Failure in this match will cause the NP "the Member States" to be closed and will allow the adjunct to be attached higher up with the coordinated head "laws, regulations and administrative provisions". In this case, all the inherent features are collected in a set that subsumes them all and can be used to fire CCC.

Notice that the preposition "for" is lexically restricted in our representation for the noun "liability", and the corresponding PP that "for" heads interpreted as a complement rather than as an adjunct. We include here below the relevant portion of each utterance in which the two mechanisms we proposed can be usefully seen at work. We marked with a slash the place in the input text in which, usually when the current constituent is a NP a decision must be taken as to whether causing the parser to close (MA) or to accept more text (LC) is actually dependent upon the presence of some local trigger. This trigger is mostly a preposition; however, there are cases in which, see e., f., h., i., the trigger is a conjunction or a participle introducing a reduced relative clause. Coordinate NPs are a big source of indecision and are very hard to be detected if based solely on syntactic, lexical and semantic information. For instance, e. can be thus disambiguated, but h. requires a matching of prepositions; In the case represented by i. we put a boundary just before a comma: in case the following NP "the Member State" is computed as a coordination - which is both semantically, syntactically and lexically possible, the following sentence will be deprived of its lexical SUBJect NP - in this case, the grammar activates a monitoring procedure independently so that backtracking will ensue, the coordinate NP destroyed and the comma computed as part of the embedded parenthetical (which is in turn an hypothetical within a subordinate clause!!). Notice also that a decision must be taken in relation to the absolutives headed by a past participle which can be intended as an active or a passive past participle: in the second case the head noun would have to be computed as an OBJect and not as a SUBJect

b. a differing degree of protection of the consumer | **against** damage caused by a defective product | **to** his health or property

c. in all member states | **by** adequate special rules, it has been possible to exclude damage of this type | **from** the scope of this directive

d. to claim full compensation for the damage | **from** any one of them

e. the manufacturer of a finished product, the producer of any raw material or the manufacturer of a component part | **and** any person

f. The liability of the producer | **arising** from this directive

g. any person who imports into the community a product | **for** sale, hire or any form of distribution | **in** the course of his business

h.both by a defect in the product | **and** by the fault of the injured person

i.However, if... the commission does not advise the Member State | **concerned** that it intends submitting such a proposal | **to** the council | , the Member State

4.3 Principles of Sound Parsing

o Principle One: Do not perform any unnecessary action that may overload the parsing process: follow the Strategy of Minimal Attachment;

o Principle Two: Consume input string in accordance with look-ahead suggestions and analyse incoming material obeying the Strategy Argument Preference;

o Principle Three: Before constructing a new constituent, check the storage of WellFormed Substring Table(WFST). Store constituents as soon as they are parsed on a stack organized as a WFST;

o Principle Four: Interpret each main constituent satisfying closer ties first - predicate-argument relations - and looser ties next - open/closed adjuncts as soon as possible, according to the Strategy of Functional Preference;

o Principle Five: Erase short-memory stack as soon as possible, i.e. whenever clausal constituents receive Full Interpretation.

oStrategy Functional Preference: whenever possible try to satisfy requirements posed by predicate-argument structure of the main governing predicate as embodied in the above Principles; then perform semantic compatibility checks for adjunct acceptability.

oStrategy Minimal Attachment: whenever Functional Preference allows it apply a Minimal Attachment Strategy.

The results derived from the application of Principle Four are obviously strictly linked to the grammatical theory we adopt, but they are also the most natural ones: it appears very reasonable to assume that arguments must be interpreted before adjuncts can be and that in order to interpret major constituents as arguments of some predicate we need to have completed clause level structure. In turn adjuncts need to be interpreted in relation both to clause level properties like negation, tense, aspect, mood, possible

subordinators, and to arguments of the governing predicate in case they are to be interpreted as open adjuncts.
As a straightforward consequence, owing to Principle Five we have that reanalysis of a clause results in a Garden Path(GP) simply because nothing is available to recover a failure that encompasses clause level reconstruction: we take that GP obliges the human processor to dummify all naturally available parsing mechanisms, like for instance look-ahead, and to proceed by a process of trial-and-error to reconstruct the previously built structure in order not to fall into the same mistake.

As a peculiar case of parser architecture interaction with theoretically driven strategies, consider the following couple of examples of the Extraposed Relative Clause containing a Short Anaphor taken from the Appendix in the Made-Up sentences reported here below,
21a. The doctor called in the son of the pretty nurse who hurt herself.
21b. The doctor called in the son of the pretty nurse who hurt himself.

In the second example we have the extraposition of the relative clause (hence RC), a phenomenon very common in English but also in Italian and other languages. The related structures theoretically produced, could be the following ones:

```
21a.
s[np[The doctor],
  ibar[called in],
        vp[np[the son, pp[of, np[the pretty nurse,
        cp[who, s[pro, ibar[hurt],
                vp[sn[herself]]]]]]]]

21b.
s[np[The doctor],
    ibar[called in],
        vp[np[the son, pp[of, np[the pretty nurse]],
                        cp[who, s[pro, ibar[hurt],
                vp[sn[himself]]]]]]]
```

If this is the correct input to the Binding Module, it is not the case that 10a. will be generated by a parser of English without special provisions. The structure produced in both cases will be 1a. seen that it is perfectly grammatical, at least before the binding module is applied to the structure and agreement takes place locally, as required by the nature of the short anaphor. It is only at that moment that a failure in the Binding Module warns the parser that something wrong has happened in the previous structure building process. However, as the respective f-structures show, the only output available is the one represented by 10b, which wrongly attaches the RC to the closest NP adjacent linearly to the relative pronoun:

```
10b.
s[np[The doctor],
        ibar[called in],
        vp[np[the son,
                pp[of, np[the pretty nurse,
        cp[who, s[pro, ibar[hurt],
                vp[sn[himself]]]]]]]]]
```

The reason why the structure is passed to the Binding Module with the wrong attachment is now clear: there is no grammatical constraint that prevents the attachment to take place. The arguments of the governing predicate HURT are correctly expressed and are both coherent and consistent with the information carried out by the lexical form. At the same time the Syntactic Binding has taken place again correctly by allowing the empty "pro" in SUBJect position of the relative adjunct to be "syntactically controlled" by the relative pronoun, which is the TOPic binder, in turn syntactically controlled by the governing head noun, the NURSE. There is no violation of agreement, nor of lexical information, nor any other constraint that can be made to apply at this level of analysis in order to tell the parser that a new structure has to be produced.
The question would be to prevent Failure since we do not want Constituent Structure Building to be dependent upon the Binding of the Short Anaphor. The only way out of this predicament is that of anticipating in Sentence Grammar some of the Agreement Checking Operations as proposed above. So the Parser would be able to backtrack while in the Grammar and to produce the attachment of the Relative Clause at the right place, in the higher NP headed by the masculine N, “the son”. The important result would be that of maintaining the integrity of Syntax as a separate Module which is responsible in “toto” of the processing of constituent structures. The remaining Modules of the Grammar would be fully consistent and would use the information made available in a feeding relation, so that interpretation will follow swiftly.
The reason why the structure is passed to the Binding Module with the wrong attachment is now clear: there is no grammatical constraint that prevents the attachment to take place. The arguments of the

governing predicate HURT are correctly expressed and are both coherent and consistent with the information carried out by the lexical form. At the same time the Syntactic Binding has taken place again correctly by allowing the empty "pro" in SUBJect position of the relative adjunct to be "syntactically controlled" by the relative pronoun, which is the TOPic binder, in turn syntactically controlled by the governing head noun, the NURSE. There is no violation of agreement, nor of lexical information, nor any other constraint that can be made to apply at this level of analysis in order to tell the parser that a new structure has to be produced (see .

To integrate this suggestion coming from Implementation problems, into the theoretical Framework of LFG or other similar theories we need to integrate GRAMMATICALITY PRINCIPLES as they have been stipulated so far, to be consisting of:

- UNIQUENESS
- COHERENCE
- COMPLETENESS

with the additional restriction:

- BOUND ANAPHORA AGREEMENT

i.e. short anaphors should be checked before leaving sentence grammar, for agreement with their antecedents iff available in their Minimal Nucleus.

Bibliography

Delmonte R. 1990. Semantic Parsing with an LFG-based Lexicon and Conceptual Representations, Computers & the Humanities, 5-6, 461-488.

Delmonte R. 2000a. Generating and Parsing Clitics with GETARUN, Proc. CLIN'99, Utrech, pp.13-27.

Delmonte R., D. Bianchi. 2002. From Deep to Partial Understanding with GETARUNS, Proc.ROMAND2002, Università Roma2, Roma, pp.57-71.

Delmonte R., D.Bianchi, E.Pianta. 1992. GETA_RUN - A General Text Analyzer with Reference Understanding, in Proc. 3rd Conference on Applied NLP, Systems Demonstrations, Trento, ACL, 9-10

Bresnan, Joan. 2001. Lexical-Functional Syntax. Blackwells.

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

Delmonte R. R.Dolci. 1997. Sound Parsing and Linguistic Strategies, Atti Apprendimento Automatico e Linguaggio Naturale, Torino, pp.1-4.

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.

Berwick, R.C., S. Abney, and C. Tenny. 1991. Principle-Based Parsing: Computation and Psycholinguistics. New York: Kluwer Academic Publishers.

Pereira F. and Warren D. 1980. Definite Clause Grammars for Language Analysis. A Survey of the Formalism and a Comparison with Augmented Transition Networks. Artificial Intelligence, 13, 231-278.

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

Delmonte R. 1997. Lexical Representations, Event Structure and Quantification, Quaderni Patavini di Linguistica, 15, 39-93.

Dibattista D., E.Pianta, R.Delmonte(1999), Parsing and Interpreting Quantifiers with GETARUN, Proc. VEXTAL, Unipress, 215-225.

Delmonte R., D.Bianchi. 1991. Binding Pronominals with an LFG Parser, Proceeding of the Second International Workshop on Parsing Technologies, Cancun(Messico), ACL 1991, pp. 59-72.

Delmonte R., D.Bianch. 1999. Determining Essential Properties of Linguistic Objects for Unrestricted Text Anaphora Resolution, Proc. Workshop on Procedures in Discourse, Pisa, pp.10-24.

Delmonte R. 2002a. From Deep to Shallow Anaphora Resolution: What Do We Lose, What Do We Gain, in Proc. International Symposium RRNLP, Alicante, pp.25-34.

Delmonte R. 1987. Grammatica e Ambiguità in Italiano, Annali di Ca' Foscari, XXVI, 1-2, 257-333.

Halvorsen P.K. and R.Kaplan. 1988. Projections and semantic description, Proceedings of the International Conference on Fifth Generation Computer System, Tokyo, 1116-1122.

Halvorsen P.K. 1983. Semantics for Lexical Functional Grammar,Linguistic Inquiry 4,567-616.

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

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

Pritchett B.L. 1992. Grammatical Competence and Parsing Performance, The University of Chicago Press, Chicago.

Schubert L.K. 1984. On Parsing Preferences, Proc. of COLING, 247-250.

Hobbs J.R., D.E.Appelt, J.Bear, M.Tyson. 1992. Robust Processing of Real-World Natural-Language Texts, Proc.2nd Conference on NLP, 186-192.

D.Hindle & M.Roth. 1993. Structural Ambiguity and Lexical Relations, Computational Linguistics 19, 1, 103-120.

Frazier L. 1987a. Sentence processing, in M.Coltheart(ed), Attention and Performance XII, Hillsdale, N.J., Lawrence Elbaum.

Delmonte R. 2000b. Shallow Parsing And Functional Structure In Italian Corpora, LREC, Atene, pp.113-119.

Delmonte R. 2000c. Parsing Preferences and Linguistic Strategies, in LDV-Forum - Zeitschrift fuer Computerlinguistik und Sprachtechnologie - "Communicating Agents", Band 17, 1,2, (ISSN 0175-1336), pp. 56-73.

Delmonte R. 2000d. Parsing with GETARUN, Proc.TALN2000, 7° confèrence annuel sur le TALN,Lausanne, pp.133-146.

Frazier L. 1987b. Theories of sentence processing, in J.Garfield(ed), Modularity in knowledge representation and natural language understanding, Cambridge Mass., MIT Press, 291-308.

Clifton C., & F. Ferreira. 1989. Ambiguity in Context, in G.Altman(ed), Language and Cognitive Processes, op.cit., 77-104.

Altman G.T.M.(ed.). 1989. Language and Cognitive Processes 4, 3/4, Special Issue - Parsing and Interpretation.

Steedman M.J. & Altmann G.T.M. 1989. Ambiguity in Context: a Reply, in Altman(ed), op.cit., 105-122.

Tapanainen P. and Voutilainen A. 1994. Tagging accurately - don't guess if you know, Proc. of ANLP '94, pp.47-52, Stuttgart, Germany.

Brants T. & C.Samuelsson. 1995. Tagging the Teleman Corpus, in Proc.10th Nordic Conference of Computational Linguistics, Helsinki, 1-12.

Kay Martin. 1980. Algorithm Schemata and Data Structures in Syntactic Processing, CSL-80-12, Xerox Corporation, Palo Alto Research Center.

Stock O. 1989. Parsing with flexibility, dynamic strategies and idioms in mind, in Computational Linguistics, 15, 1, 1-18.

Woods W.A. 1973. An Experimental Parsing System for Transition Network Grammars, in Rustin(ed.) 1973. Natural Language Processing, Algorithmic Press, New York, pp.111-154.

Delmonte R. 2002b. Relative Clause Attachment And Anaphora: A Case For Short Binding, Proc.TAG+6, Venice, 84-89.

Delmonte R.(1992), Linguistic and Inferential Processing in Text Analysis by Computer, Padova, Unipress.

APPENDIX

We report in the appendix the texts that have been used to test the functioning of the parser. We only include English and Italian texts. There is no space here to show the output of the parser, which is however freely available from the author (but see also Delmonte, 1992).

SECTION 1. ENGLISH TEXTS

LEGAL TEXT – European Council Directive

Council directive of july 1985 on the approximation of the laws, regulations and administrative provisions of the Member States concerns liability for defective products.

The Council of the European Communities has adopted this directive.

Having regard to the proposal from the commission.

Whereas approximation of the laws of the Member States concerning the liability of the producer for damage caused by the defectiveness of his products is necessary because the existing divergences may entail a differing degree of protection of the consumer against damage caused by a defective product to his health or property.

Whereas liability without fault should apply only to movables which have been industrially produced.

Whereas protection of the consumer requires that all the producers involved in the production process should be made liable in so far as their finished product, component part or any raw material supplied by them was defective.

Whereas, to the extent that liability for nuclear injury or damage is already covered in all Member States by adequate special rules, it has been possible to exclude damage of this type from the scope of this directive.

Whereas to protect the physical well-being and property of the consumer the defectiveness of the product should be determined by reference not to its fitness for use but to the lack of the safety which the people at large is entitled to expect.

Producer means the manufacturer of a finished product, the producer of any raw material or the manufacturer of a component part and any person who by putting his name, trade mark or other distinguishing feature on the product presents himself as its producer.

The producer shall be liable for damage caused by a defect in his product.

The injured person shall be required to prove the damage, the defect and the causal relationship between defect and damage.

Where, as a result of the provisions of this directive, two or more persons are liable for the same damage.

They shall be liable jointly and severally, without prejudice to the provisions of national law concerning the rights of contribution or recourse.

A product is defective when it does not provide the safety which a person is entitled to expect, taking all circumstances into account.

The liability of the producer arising from this directive may not, in relation to the injured person be limited or excluded by a provision limiting his liability or exempting him from liability.

This directive shall not apply to injury or damage arising from nuclear accidents and covered by international conventions ratified by the Member States.

Without prejudice to the liability of the producer any person who imports into the community a product for sale, hire or any form of distribution in the course of his business shall be deemed to be a producer within the meaning of this directive.

However, if within three months of receiving the said information the commission does not advise the Member State concerned that it intends submitting such a proposal to the council, the Member State may take the proposed measure immediately.

The liability of the producer may be reduced or disallowed when, having regard to all the circumstances, the damage is caused both by a defect in the product and by the fault of the injured person or any person for whom the injured person is responsible.

Any member state may provide that a producer's total liability for damage resulting from a death or personal injury and caused by identical items with the same defect shall be limited to an amount which may not be less than 70 million ecu.

LITERARY TEXTS – excerpts taken from Virginia Woolf's novels

John gave Mary a rose.
She took it and put it in her hair.
She knew that she had been given a present, something precious.
When Steve faced them saying : "are you enjoying yourselves ?".
"It was horrible ! It was shocking !".
Not for herself.
She felt only hostility and his determination to ruin that wonderful moment.
John smiled and went away embarassed.
The three friends went all outdoors.
As they were walking in the garden, John said to himself "Sara will marry that man", without any resentment.
Richard would marry Sara.
He felt strongly about that.
She was the right person for a man like Richard.
For himself he was absurd.
His demands upon Sara were absurd.
She would have accepted him still if he had been less absurd.
Richard began to sing.
Mary picked up the phone and called Jason.
Her husband, she thought, would have considered such a move as untruthful and utterly base.
Perhaps there was something bad in herself that could not help but do the wrong thing at the wrong time.
Jason answered immediately.
John went into a restaurant.
There was a table in the corner.
The waiter took the order.
The air was nice and clean.
The atmosphere was warm and friendly.
He began to read his book.

LINGUISTICALLY BIASED MADE-UP TEXTS

John beats the donkey that he saw.
John beats every donkey that he saw on his house.
John loves every boy that saw his sister.
John spoke to his wife.
John spoke to his wife because she loves his sister.
Every man that has a donkey beats it.
John talked to Mary about herself on the bus.
John talked to Frank about himself.
John talked to Mary about himself.
Mary talked to Frank about herself.
Which boy saw his sister?
His wife likes John.
John gave him a book.
John wants Mary to read a book.
The farmer beats his donkey because it does not obey him.
Which boy said that he loves his sister?
The farmer does not beat his donkey because he prefers to feed it.
Which farmer beats every donkey that he owns?
Who beats his donkey?
John, who spoke about Mary, said that she loves him.
Who said that John loves Mary ?
Talking about himself pleases Mary.
John beats his donkey with the telescope.
John beats his donkey from the bus.
John does not beat his wife.
His brother.
Who does beat John?
Who does not beat Mary?
Is Mary wise?
Did Mary love John?
While he spoke about his brother_in_law, Frank said to Mary , who insulted him, to let him go.
John will not marry Mary because she is rich.
John told her that Mary is nice.
What did John give to Mary?
What did John say?
What is it about?
Who do his friends like?
Who likes his friends?
His friends like John.
About his friends.
Mary put the book on the table.
Mary saw the cop with the binoculars.
Mary saw the cop with the revolver.
The cop killed the man with the revolver.
Mary promised his brother that she would come.
Mary knew the answer very well.
The doctor called in the son of the pretty nurse who hurt herself.
The doctor called in the son of the pretty nurse who hurt himself.
Mary will say that it rained yesterday.
Mary said that it will rain yesterday.
Mary will say that it rained tomorrow.
Mary said that it will rain tomorrow.
The authorities refused permission to the demonstrators because they feared violence.
The authorities refused permission to the demonstrators because they supported the revolution.
The thieves stole the painting in the night.
The thieves stole the painting in the museum.

SECTION II: ITALIAN TEXTS

o La storia dei tre porcellini / The story of the three little pigs

C'erano una volta tre fratelli porcellini che vivevano felici nella campagna. Nello stesso luogo però viveva anche un terribile lupo che si nutriva proprio di porcellini grassi e teneri. Questi allora, per proteggersi dal lupo, decisero di costruirsi ciascuno una casetta. Il maggiore, Jimmy che era saggio, lavorava di buona lena e costruì la sua casetta con solidi mattoni e cemento. Gli altri, Timmy e Tommy, pigri se la sbrigarono in fretta costruendo le loro casette con la paglia e con pezzetti di legno. I due porcellini pigri passavano le loro giornate suonando e cantando una canzone che diceva: chi ha paura del lupo cattivo. Ma ecco che improvvisamente il lupo apparve alle loro spalle. Aiuto, aiuto, gridarono i due porcellini e cominciarono a correre più veloci che potevano verso la loro casetta per sfuggire al terribile lupo. Questo intanto si leccava già i baffi pensando al suo prossimo pasto così invitante e saporito. Finalmente i porcellini riuscirono a raggiungere la loro casetta e vi si chiusero dentro sbarrando la porta. Dalla finestra cominciarono a deridere il lupo cantando la solita canzoncina: chi ha paura del lupo cattivo. Il lupo stava intanto pensando al modo di penetrare nella casa. Esso si mise ad osservare attentamente la casetta e notò che non era davvero molto solida. Soffiò con forza un paio di volte e la casetta si sfasciò completamente. Spaventatissimi i due porcellini corsero a perdifiato verso la casetta del fratello. "Presto, fratellino, aprici! Abbiamo il lupo alle calcagna". Fecero appena in tempo ad entrare e tirare il chiavistello. Il lupo stava già arrivando deciso a non rinunciare al suo pranzetto.Sicuro di abbattere anche la casetta di mattoni il lupo si riempì i polmoni di aria e cominciò a soffiare con forza alcune volte. Non c'era niente da fare. La casa non si mosse di un solo palmo. Alla fine esausto il lupo si accasciò a terra. I tre porcellini si sentivano al sicuro nella solida casetta di mattoni. Riconoscenti i due porcellini oziosi promisero al fratello che da quel giorno anche essi avrebbero lavorato sodo.

o La stroria di Avveduti /The story of Avveduti

Fino a tre anni fa Franco Avveduti non si era mai immischiato col mondo della pubblica amministrazione. Come burocrate, era un immigrato che veniva dal di fuori. Figlio di buona famiglia, a venti anni decise di iscriversi alla accademia militare di cavalleria. Era un buon allievo con ottime qualifiche. Più tardi fu un ufficiale di successo. Poi nel 1945 Avveduti si dimise dallo esercito. I militari lo avevano deluso. Avveduti, deposta la divisa, si iscrisse alla università. Nel 1947 era laureato e nel 1948 era procuratore legale. Intanto a Verona aveva conosciuto Paola, figlia di Antonio Alberti, potente senatore democristiano, e la aveva sposata. Il senatore poteva considerarsi il più influente uomo politico veronese. Gli elettori lo mandavano al parlamento coprendolo di voti preferenziali. Al seguito di Alberti, che era diventato vicepresidente del senato, Franco Avveduti nello immediato dopoguerra si trasferì a Roma. Tuttavia, da principio si tenne lontano dalla sfera di interessi del suocero.

Gli piaceva parlare del suocero come di una facile occasione mancata che chiunque altro avrebbe sfruttato ma che lui, Avveduti, preferiva lasciare perdere. Solo verso il 1950 decise di accettare un posto nella organizzazione della fiera di Verona. Lo nominarono delegato cioè una specie di funzionario viaggiante con incarichi diplomatici di tenere i rapporti con le delegazioni commerciali, curare i produttori stranieri, le grandi ditte, la stampa. Questo era un compito che corrispondeva bene alla sua vocazione e nel quale Avveduti sapeva giostrare con notevole agilità. Quando il suocero morì, egli non perse il posto. A Verona il collegio di Alberti lo aveva ereditato Trabucchi e col collegio aveva ereditato la presidenza della fiera. Trabucchi continuò a valersi della collaborazione di Avveduti. L'ex-ufficiale del Novara-Cavalleria gli era simpatico. La sua distinzione lo impressionava. Lo confermò nell'incarico alla fiera. Avveduti funzionava benissimo come segretario particolare. Sapeva mobilitare prefetti e questori. Tutti gli invidiavano il suo segretario particolare.

o La storia dei tre porcellini rivisitata / The story of the little pigs revisited

Questa è la storia di tre porcellini che andarono per il mondo a cercare fortuna. I loro nomi erano Timmy, suonatore di flauto, Tommy, violinista e Jimmy, grande lavoratore. Giunti in un bel bosco, decisero di costruire ognuno una comoda casetta. A Timmy non piaceva per niente lavorare così pensò di costruirsi rapidamente una capanna di paglia. In breve la casetta fu pronta e Timmy decise allora di andare a vedere che cosa stavano facendo i suoi fratellini. Incontrò dapprima Tommy il violinista. Anche lui non aveva molta voglia di faticare così costruiva con dei pezzi di legno una semplice casetta. Ben presto anche la casa di legno fu pronta. Come quella di paglia, non era certo molto resistente. Ma i due porcellini scansafatiche se la erano sbrigata in poco tempo ed ora potevano tranquillamente divertirsi. Mentre Timmy suonava il flauto, Tommy lo accompagnava con il suo violino e insieme se la spassavano allegramente. Poi stanchi di fare baldoria, decisero di andare a vedere che cosa stava facendo il loro fratellino. Si misero in cammino e ben presto raggiunsero Jimmy. Il bravo porcellino stava costruendo anche lui la sua casetta. Ma poiché Jimmy era previdente e non aveva paura di lavorare sodo, la costruiva con mattoni e cemento. Jimmy voleva una casa robusta perché sapeva che il lupo cattivo viveva nel bosco vicino. Quando i due pigri porcellini videro Jimmy impegnato nel suo duro lavoro, si misero a ridere a crepapelle. Ma quei due sciocchi porcellini non pensavano al pericolo. Così continuarono a prendere in giro il saggio Jimmy, canticchiando e suonando con il flauto e il violino. I due porcellini, sempre suonando e ballando, tornarono ciascuno alla propria fragile casetta. Ma appena Timmy aprì la porta, sbucò fuori dal bosco il lupo cattivo. Il porcellino lo vide e tremante di paura si chiuse immediatamente in casa. Il lupo cattivo cominciò a chiamarlo. "Apri la porta e fammi entrare nella tua casetta di paglia."

LINGUISTICALLY BIASED MADE-UP TEXTS

Jimmi è in casa.
C è jimmi in casa.
C è jimmi.
Jimmi ha un libro in casa.
Jimmi ha paura del lupo.
Jimmi c'ha un libro in casa.
Jimmi c'ha paura del lupo.
Jimmi è un porcellino.
Jimmi è saggio.
Si costruirono molte case.
Sembra essersi lavorato bene.
Sembra essersi sbrigato in fretta.
Avendo jimmy lavorato bene timmy era felice.
Costruì tutto.
Ne costruì tutto.
Costruì tutti.
Li costruì tutti.
Ne costruì tutti.

Ne costruì molti.
Costruì molto.
Costruì molti.
Avvisò molti.
Apparvero molti.
Apparvero tutti.
Ne apparvero tutti.
Ne apparvero molti.
Avvisò tutti.
Jimmi costruì molte delle sue case con il cemento.
Jimmi ne costruì molte con il cemento.
I porcellini si sono costruiti una casa.
I porcellini gli hanno costruito una casa.
Parlare del suocero era importante.
Parlare del suocero era importante per avveduti.
Il lupo costruì loro una casa.
Il lupo costruì loro una casa ciascuno.
Il lupo gli leccava i baffi.
Il lupo gli riempì i polmoni di aria.
Jimmi gli ha rotto un mattone sulla spalla.
Gli si è rotto un mattone sulla spalla.
Le case sono state ereditate.
Sono state ereditate due case.
Le case furono costruite da jimmi.
Jimmi si è rotto un mattone sulla spalla.
Mario non sapeva a chi parlava.
I suoi lavori alla propria casa.
Il suo attaccamento alla propria famiglia.
La liberazione dei prigionieri da parte del ministro.
La uscita della statale 592.
La uscita dalla statale 592.
Il libro di jimmi di maria.
La interruzione della statale 592 di canelli per lavori che dureranno fino a due settimane.
Questa è la storia di tre porcellini.
Jimmi costruì molte delle sue case coi mattoni.
Però ne costruì alcune con la paglia.
Furono costruite molte case.
Non è arrivato nessuno.
Nessuno è arrivato.
Marco non è arrivato.
Non è arrivato marco.
Il padrone insultò gli studenti.
Il padrone insultò studenti.
Mario corre.
Mario è corso.
Mario correrà da maria.
Mario corre da maria.
Mario è corso per ore.
Mario è corso per tre ore.
Mario domani corre.
Mario domani corre da maria.
Mario ieri corre.
Mario ieri è corso da maria.
Mario giovedì corre da maria.
Mario ogni giorno corre da maria.
Mario ogni tre giorni corre da maria.
Mario giovedì è corso da maria.
Mario è corso tre volte da maria.
Mario sta correndo alle tre.
Mario domani sarà corso da maria.
Mario giovedì era malato.
Mario fa mangiare la mela a maria.
Mario fa avere mangiato la mela a maria.
Mario dice a maria che luigi mangia le mele.
Mario dice a maria che luigi mangia mele.
Mario era malato quando luigi è arrivato.
Mario era malato quando luigi giovedì è arrivato.
Mario partiva quando luigi è arrivato.
Mario partiva quando luigi arrivava.
Maria è stata malata mentre era a verona.
Maria è con luigi.
Ogni uomo che ha un asino lo picchia.
L'asino è arrabbiato.
È un animale disobbediente.
Sono animali disobbedienti.
Uno studente ha letto tutti i libri.
È mio fratello.
Uno studente sta leggendo un libro.
È molto bello.
Uno studente legge un libro.
È necessario per la sua formazione.
Tre ragazzi hanno mangiato un pesce.
Erano salmoni.
Tre ragazzi hanno mangiato un pesce ciascuno.
Gino vuole leggere un libro.
È un libro di storia.
Gino vuole conoscere uno studente che studi marx.
È un amico di maria.
Gino ama molto marx.
Gino vuole conoscere uno studente che studia marx.
Ognuno ama un libro che ha letto.
Era un libro di trabucchi.
Ognuno ama un libro che ha letto.
Erano libri di trabucchi.
Tutte le donne costruirono una casa.
La casa era solida.
Era solida.
Erano solide.
Una donna dice che ogni uomo la ammira.
Vuole anche essere felice.
Una donna vuole che ogni uomo la ammiri.
Gino la conosce.
Ogni donna ha mangiato un pesce.
Era un salmone.
Una donna ha detto che ogni uomo la ammira.
Sono vanitose.
Ogni uomo che ha un asino lo batte.
Parlare di se stesso piace.
La propria salute è necessaria.
La propria salute era necessaria.
Parlare di se stesso piaceva.
La propria salute preoccupa ognuno.
La sua salute preoccupa ognuno.
Ognuno ama i film che ha visto.
I film che ha visto piacciono a ognuno.
La madre di ogni ragazzo pensa che sia un genio.
Mario vede ogni trasmissione che parli della propria storia.
La salute della propria famiglia è importante.
Una donna desidera che ogni uomo la ammiri.
Una donna desidera che gino la ammiri.
Gli uomini che hanno un libro di trabucchi li leggono.
I porcellini hanno visto un lupo da ogni casetta.
I porcellini in ogni casa gridarono aiuto.
Il porcellino in ogni casa gridò aiuto.
Mario telefonò a luigi perché voleva delle informazioni.

Mario criticava luigi perché ha rovinato il file.
Mario critica luigi perché è ipercritico.
Mario criticava luigi perché voleva il file.
La guardia sparò al ladro perché stava scappando.
Le autorità rifiutarono il permesso ai dimostranti perché temevano la violenza.
Le autorità rifiutarono il permesso ai dimostranti perché sostenevano la rivoluzione.
I ladri rubarono i quadri nella notte.
I ladri rubarono i quadri nel museo.
Mario promise a suo fratello che sarebbe tornato.
Il dottore chiamò il figlio della infermiera che si era fatta male.
Il dottore chiamò il figlio della infermiera che si era fatto male.
Parlando di suo suocero, trabucchi ha ordinato a avveduti che lo aspettava di lasciare perdere.
Avveduti ritiene che maria ami la propria famiglia.
Avveduti ritiene che lui ami la propria famiglia.
Avveduti ritiene che la propria sorella ami gino.
Lui ritiene che la propria sorella ami gino.
L'insegnante promosse lo studente poiché era preparato.
La ragazza che la propria salute preoccupa è tua sorella.
La ragazza che sua madre ama è sua sorella.
Mario ha visto lo asino sopra la propria casa.
Mario ha visto lo asino dalla propria casa.
Mario batte un asino sulla propria casa.
Mario batte un asino dalla propria finestra.
Mario ha visto lo asino sopra la sua casa.
Mario ha visto lo asino sopra se stesso.
Mario ha visto lo asino sopra di sé.
Mario ha visto lo asino sopra di lui.
Quale ragazzo hai detto che mario ha visto ?
Chi dici che mario ha visto ?
Chi dici che ha visto se stesso ?
Mario ha visto un asino col cannocchiale.
A se stesso gino ritiene che mario non ci pensi mai.
Di se stesso gino ritiene che mario non parli mai.
Gino ritiene che mario non parli mai di se stesso.
Gino ritiene che mario ci pensi mai a se stesso.
Ognuno ama il suo asino.
Ogni uomo ha detto che batte il suo asino.
Trabucchi ha visto ogni uomo che batte il suo asino.
Mario ritiene che la propria libertà sia importante.
Ognuno ama i film che ha visto.
I film che ha visto piacciono a ognuno.
La madre di ogni ragazzo pensa che sia un genio.
Mario vede ogni trasmissione che parli della propria storia.
Mario ha visto un asino col cannocchiale sopra la propria casa.
Quale libro mario ha perso ?
Di quale libro mario parlava ?
Con quale uomo mario parlava ?
Chi batte lo asino ?
Di chi parlava mario ?
Con chi parlava mario ?
Quale oceano che confina con l'Africa è inquinato ?
Quale è l'oceano che confina con i paesi della Africa ?
Quale padrone che ha un asino lo batte ?
Chi batte l'asino che lui ha visto ?
Quale è l'asino che il padrone batte ?
Di quale ragazzo sua madre parlava ?
Di quale ragazzo la propria madre parlava ?
Quale ragazzo la propria salute preoccupa ?
Chi sai che mario ha visto ?
Avveduti ha incontrato trabucchi che lo accusava di avere perso il suo libro.
Avveduti ha incontrato trabucchi che lo accusava di avere rubato il libro a lui.
Avveduti ha incontrato trabucchi che si accusava di avere preso il libro a lui.
Partire in quel modo provocò a trabucchi dispiacere.
Partire in quel modo gli provocò male di testa.
Io ho incontrato trabucchi che mi ha detto di dire a avveduti che lo aspetta.
Non abbiamo votato per nessun candidato dal momento che il presidente lo aveva raccomandato.
I suoi sostenitori non hanno votato per nessun candidato.
L'uomo che tu dovresti avvertire ogni qualvolta tu lo incontri.
Noi non siamo stati in grado di leggere nessuna confessione prima che lo autore decidesse di renderla pubblica.
Lo uomo che tu dovresti avvertire ogni qualvolta tu incontri.
Noi non siamo stati in grado di leggere nessuna confessione prima che il suo autore decidesse di renderla pubblica.
A loro interessano se stessi.
Avveduti sperava che i giornali parlassero di sé.
Avveduti sperava che i giornali parlassero di lui.
Maria riteneva ognuno innamorato di sé.
Maria riteneva ognuno innamorato di lei.
Mario vide il toro sopra di lui.
Se stessi interessano loro.
Avveduti ritiene che quella casa appartiene alla propria famiglia.
Lui ritiene che gino sia amato dalla propria sorella.
Ad ognuno sta a cuore la propria salute.
Avveduti mi insultò pesantemente quando lo interrogai.