Internship Report
July-December 2012

Alexandra Fiotaki

Practice with the XLE platform: development of toy grammars of Modern Greek

Athens, January 2013

Supervisor: Dr. Stella Markantonatou
INSTITUTE FOR LANGUAGE AND SPEECH PROCESSING/R.C. “ATHENA”
This is an internship report. I did my internship with the Institute of Language and Speech Processing of “Athena”-Research and Innovation Center in Information, Communication and Knowledge Technologies (ILSP/R.C. “Athena” ) in the framework of the Interdisciplinary Interuniversity Postgraduate Course in Language Technologies “TEXNOGLOSSIA” that is organised by the National & Kapodistrian University of Athens, Faculty of Filology - Dept. of Linguistics, the National Technical University of Athens, School of Electrical and Computer Engineering and the Institute for Language and Speech Processing.

ILSP/R.C. “Athena” plans to develop a Greek Grammar with XLE. XLE is an environment for the development of parallel LFG grammars (http://www2.parc.com/isl/groups/nltt/xle/). This was the first attempt of ILSP/R.C. “Athena” to work with XLE on Modern Greek data. Here, I report on work that mostly explores the tools and the expressivity of XLE with applications on Modern Greek grammatical phenomena that often are different from English ones.
CONTENTS

1. The formalisms of LFG and XLE ................................................................. 4
   1.1 Introduction ............................................................................................ 4
   1.2 Overview of LFG .................................................................................. 4
2. Parsing with XLE .......................................................................................... 13
3. Templates and Lexical Rules ..................................................................... 17
   3.1 Lexicon section ...................................................................................... 17
   3.2 Grammatical Functions ........................................................................ 25
1. The formalisms of LFG and XLE

1.1 Introduction

In this paper, we explore the platform XLE, a joint project between the NLTT group at PARC and the MLTT group in Grenoble that began in October 1993. XLE is a complete parsing implementation of the Lexical Functional Grammar (LFG) syntactic formalism and is considered as one of the best available parsing systems by taking into account particular criteria such as depth of analysis and linguistic motivation. XLE consists of a parser, a generator and a graphical user interface for writing and debugging such grammars. It has a very rich c-structure rule notation and various kinds of abbreviatory devices such as parameterized templates, macros, and complex categories.

In what follows below, first a very brief introduction to LFG is given. Then we embark on the detailed presentation of XLE. The features of XLE are exemplified with fragments of a toy grammar we developed for Modern Greek.

1.2 Overview of LFG

LFG is one of many formal methods of describing grammars of natural languages and is considered as a variety of generative grammar. It defines different levels of representation in order to encode syntactic, semantic and other information. The development of this theory was initiated by Joan Bresnan and Ronald Kaplan in the 1970s (Falk, 2001) LFG conceptualises language as being made up of multiple dimensions of structure. Each of these dimensions is represented as a structure with its own rules and forms. So, LFG assumes two synaptic levels of representation, each one exploiting a different formalism:

- c-structure (constituent structure):
  It is defined in terms of syntactic categories, terminal strings and their relationships or simply an overt linear and hierarchical organization of words into phrases. It is represented with phrase structure trees.

- f-structure (functional structure):
  It is an abstract functional organization of the sentence, explicitly representing syntactic predicate-argument structure and functional relations. It is represented with attribute-value matrices (feature structures). It is projected from the c-structure on the basis of lexical and syntactic information.
For instance, the XLE parser coupled with the LFG grammar assigns the structures (2) and (3) to the string (1):

(1) Η Μαρία αγοράζει μήλα.
   ‘Mary buys apples.’

The two structures are related to each other in a way that is explicitly stated on the phrase structure rules of the LFG grammar as will be explained in the rest of this presentation. In a nutshell, however, c-structure offers order and dominance relation information while f-structure offers information about deep features such as lexical properties and grammatical function information.

(2) C-structure

(3) F-structure
One pillar of LFG is its lexicalism, more particularly the assumption that syntactic processes do not mingle with word formation (the so-called Lexical Integrity Principle of LFG). Practically this means that syntactic rules are distinct from the morphological ones and that syntax does not affect the contents of the lexicon. A second pillar of LFG is that grammatical functions such as Subject and Object are primitives rather than being notions that are structurally defined. Among the far reaching consequences of this idea are that predication relations are treated independently of word order, that there is no need for c-structure transformations and that long distance dependency phenomena such as control and binding receive a treatment that does not use traces.

We have mentioned that the Grammatical functions can be seen in the grammatical rules. A grammatical function is a particular kind of dependency. A verb could have dependants such as subject and object, and each one of them has a different relation with the verb. LFG maintains that ‘subject’ and 'object' are distinct grammatical functions.

There are three basic groups of grammatical functions:
1. Subcategorisable functions: SUBJ, OBJ, OBJ-TH, COMP, XCOMP
2. Adjunct (non-subcategorisable) function: ADJ, XADJ

We will come back to the issue of grammatical functions in Section 3.2.

In LFG, phrase-structure rules are annotated with the so-called “Functional Equations” that encode the relationship between c-structure trees and f-structures. These equations make heavy use of f-structure metavariables. In each rule, the “^” metavariable on a non terminal node refers to the image of the mother c-structure node and the “!” metavariable refers to the non terminal node itself. The correspondence between the XLE notation and the LFG one: “^” = “↑” and “!” = “↓”.

For instance, in XLE notation we have:
(4) $S \rightarrow NP \quad VP$
\[ (^{SUBJ=(!)} \quad ^{=} !) \]

Metavariables are also used in lexical entries in order to encode information about the nodes that dominate them.
In Schema 1 the relationship is given between the c-structure tree of the sentence “John likes Mary” and the corresponding f-structure.

The f-structure representation of a sentence is constrained by
A. The f-structure constraints that are associated with the c-structure rules and lexicon entries, applied in the analysis of each sentence and expressed with functional equations
B. The three principles of functional completeness, coherence and uniqueness.

**Functional equations** can be divided into 2 types:

1. **Defining equations** (=): The functional equations that we have already used in example (4) above. They define an f-structure attribute as existing and having a particular value.
   For example: \(^= !\)

2. **Constraining equations** (=c): They do not introduce a new feature-value pair, but require a particular feature value to be present. They are presented by subscripting the letter c to the equal sign ` =c´.
   (5) τρωω (eat) \(\uparrow\text{PRED}) = \text{τρωω} (\uparrow\text{SUBJ}) (\uparrow\text{OBJ}) \)
   \((\uparrow\text{OBJ CASE})=\text{c} \quad \text{ACC}\)
   In (5) above, the constraint does not assign the case ACC to the OBJ, instead it presupposes it.
   Constraining equations do not provide direct information, only constraints.
   The given information could be considered as positive information.
Moreover, the f-structure has to satisfy 3 **principles**. These are:

- **Coherence:** All the given arguments in the structure must be required by a predicate.

  For instance, in the sentence “Παιδιά διαλέγουν βιβλία (Kids choose books)” the verb ‘διαλέγω’ has an OBJ, which is not required by the lexical entry:
  \((^\text{PRED}) = \text{διαλεγουν}(^\text{SUBJ})>\).

  So, the word ‘βιβλία’ is an argument that is not required by a predicate.

- **Completeness:** All the governable grammatical functions required by the PRED of the f-structure should have a value in the f-structure. For example,

  There is in our lexicon the lexical entry: \((^\text{PRED}) = \text{διαλεγω}(^\text{SUBJ}), (^\text{OBJ})>\)
  and we try to parse the string “εγω διαλέγω”. The constraint of completeness is violated and XLE will not return an output F-structure, because the predicate ‘διαλέγω’ asks for an OBJ which is not present in the string.

- **Uniqueness (consistency):** Every attribute must have a single value.

  For example, κοριτσι (girl) \(N\) \((^\text{NUM}) = \text{sg, pl}\)
  \((^\text{PRED}) = \text{κοριτσи’}\)
The predicate ‘κορίτσι’ has two values for the attribute ‘number’ and that will give an error message in parsing.

F-structures are total functions from attributes to values. Different attributes can have the same value but no attribute can have more than one value.

1.3 Grammar files in XLE

An XLE grammar consists of two files:

- The grammar file
- The lexicon and morphology file.

The typical preamble of an XLE grammar is (4) and includes:

- Configuration section (CONFIG) with references to other sections (even to other files) and specification of general concepts, such as the root category, the set of governable grammatical functions etc.
- Rule section (RULES) with specification of rules and macros.
- Lexicon section (LEXICON) with specification of lexicon entries
- Some more sections: templates, morphology and feature declaration.

(4) Preamble of an XLE grammar

```
TOY ENGLISH CONFIG (1.0)
ROOTCAT S.
FILES .
LEXENTRIES (TOY ENGLISH).
RULES (TOY ENGLISH).
TEMPLATES (TOY ENGLISH).
GOVERNABLERELATIONS SUBJ OBJ OBJS OBL OBL-? COMP XCOMP.
SEMANTICFUNCTIONS ADJUNCT TOPIC.
NONDISTRIBUTIVES NUM PERS.
EPSILON -
OPTIMALITYORDER NOGOOD.
----
TOY ENGLISH RULES (1.0)
----
TOY ENGLISH TEMPLATES (1.0)
----
TOY ENGLISH LEXICON (1.0)
----
```
Firstly, a representation of the c-structure is constructed drawing on (context free phrase structure) rules and on lexical entries. As already explained, LFG assumes that syntax does not affect information coming from the lexicon (the Lexical Integrity Principle). In grammar development the Lexical Integrity Principle is materialized as a grammar design requirement that XLE satisfies by defining independent sections of grammar and morphology rules.

Rules in LFG have the typical Context-Free Rule form \( A \rightarrow g \) (of course, they may extend over several lines). \( A \) is the mother category. This mother category is defined by the string \( g \), which determines the set of possible daughter strings. A daughter string must use the regular language that is prescribed in XLE for this purpose (see below).

Example: In the simplest form a top rule that would parse the sentence:

(5) Το κορίτσι ήρθε.

‘The girl came.’

would be \( S \rightarrow \text{NP VP} \).

As regards the regular expressions on right-hand side, the following expressive means are available:

- **Comments:** Comments enclosed in double quotes can be included in a rule to explain its function.
  Example: \( S \rightarrow \text{“simplest sentence”} \)
  \( \text{NP VP} \).

- **Optionality:** \( A \rightarrow (B) C \).
  Parentheses indicate optionality. The rule above compiles out to two rules
  \( A \rightarrow B \ C \) and \( A \rightarrow C \).
  Example: The rule \( \text{NP } \rightarrow (D) \text{ N} \) would parse both the NPs “το κορίτσι (the girl)” and “κορίτσι (girl)”

- **Disjunction:** \( A \rightarrow \{ B \mid C \} \ D \).
The notation \{…\} indicates disjunction. The mother category A is satisfied by any string meeting the conditions of at least one of the B or C predicates. So, the rule $A \rightarrow \{ B \mid C \}$ D above compiles out to two rules $A \rightarrow B D$ and $A \rightarrow C D$.

Example: If we have to parse the sentences

(6) Το κορίτσι έφαγε το μήλο.
   ‘The girl ate the apple’

(7) Το κορίτσι είχε φάει το μήλο.
   ‘The girl has eaten the apple.’

we can use the following single rule $S \rightarrow NP \{ VP \mid VPaux \}$ where VP and VPaux are defined with the following grammar rules:

- VP $\rightarrow V NP PP^*$
- VPaux $\rightarrow AYX VP$ (AUX: auxiliary verb, eg. the verb ‘to be’)

This constraint helps to simplify the statement of the grammatical possibilities.

- Kleene star (*): $A \rightarrow B^* C$. This symbol means zero or more occurrences of the annotated expression. Some of the rules to which $A \rightarrow B^* C$ compiles out are: $A \rightarrow B C$, $A \rightarrow C$, $A \rightarrow B B C$, $A \rightarrow B B B B C$ e.t.c.

Example: We can use the rule “NP $\rightarrow (D) AP^* N$” in order to parse the following NPs: “το εξυπνο όμορφο κορίτσι (the smart beautiful girl)” and “κορίτσι (girl)”

- Kleene plus (+): $A \rightarrow B C^+$. It is an abbreviation for “one or more occurrences of the annotated expression”. Kleene plus is different from Kleene star in that it requires that at least one element should exist.

Example: If we use the same grammar rule as before “NP $\rightarrow (D)AP^* N$”, but instead of the Kleene star, we use the Kleene plus then the NP “κορίτσι (girl)” will not be parsed.

- Grouping: $A \rightarrow [B C]$. The square brackets are used to explicitly mark that the components (B, C) are to be treated as a unit. This is really useful in cases like as VP $\rightarrow V NP^* PP^*$, where instead of that we can simply had VP $\rightarrow V [NP PP]^*$. Of course, the same could be parsed without the usage of the square brackets, but there are helpful in order to be used less symbols and be avoided mistakes. As result, all the following VPs would be parsed successfully:
(6) (Κάποιος) αγόραζε πορτοκάλια.
   ‘(Somebody) bought oranges.’

(7) (Κάποιος) τοποθέτησε το δοχείο πάνω στο τραπέζι.
   ‘(Somebody) put the vase in the table.’

(8) (Κάποιος) τοποθέτησε το δοχείο πάνω στο τραπέζι δίπλα από το κρεβάτι.
   ‘(Somebody) put the vase in the table next to the bed.’

More than one Regular expression may be used at the right-hand side of the same rule. Moreover, these notations are allowed in all places where a symbol can appear, such as categories, feature names or values, macros and templates.

There are also **f- structure designators** that express constraints:

1. **Negation/Complementation (\~):** It does not constrain a feature to have a particular value, instead it blocks a particular value. Therefore, the c subscript that we have already seen cannot be used in that occasion. For example, the expression “\~(↑TENSE) = PRES” would mean that this verb can be of whatever tense, but NOT present. So, clearly the given information is negative.

2. **Existential constraint (↑ p):** It requires a feature to be present, but it does not require a particular value for this feature. For instance, (↑TENSE) means that a Tense value is necessary to be given for the sentence to be parsed but there is no constraint as regards the value itself.

3. **Set Membership ($)**: The membership predicate declares that if we have “a₁ $ a₂”, the a₂ denotes a set and the a₁ denotes one of its elements. For instance, it is the designator for assigning adjuncts to the adjunct set: “! $ (↑ ADJUNCT)”. So, it will be used in order to parse the example:
   Η Μαρία τρώει μήλα στο τραπέζι.
   ‘Maria eats apples on the table.’
   The PP “στο τραπέζι” is given by the following rule: PP*:! $ (↑ ADJUNCT).
2. Parsing with XLE

"XLE is designed to automatically take advantage of context-freeness in the grammar of a natural language so that it typically parses in cubic time and generates in linear time" as noted in the XLE Documentation (2002-2011).

In particular, for a parsed sentence, XLE returns four charts.

Chart 1: Parse tree
Chart 2: F-structure chart
Chart 3: F-structure #1
Chart 4: Solutions

XLE processes all of the phrasal constraints first using a chart, and then uses the results to decide which functional constraints to process. After finishing the procedure (after examining the options of charts 2 and 4), it merges multiple feature structures together into a single, packed feature structure (chart 3) and presents the valid parse tree (chart 1). Below are given the four output charts for sentence (9):

(9) Ο Γιώργος παίζει τέννις.
    ‘George plays tennis.’

Chart 1: Parse Tree

```
"Ο Γιώργος παίζει τέννις"
 CS 1: S:31
    NP:13 VP:24
  D:2 N:4 V:6 NP:23
    o:1 Γιώργος:3 παίζει:5 N:8
    τέννις:7
```
The charts 2 and 3 do not show different F-structures because there are no ambiguities in the sentence parsed. For the same reason, there is no Chart 4 with different solutions, because there is only one solution, that is, only one way for parsing the particular the sentence. Of course, there was no ambiguity among others because the system knew that the form “τεννις” stood for the accusative case of the lemma “τεννις”.

For the following sentence, there is ambiguous information because the system knows that the form παιδί stands for both the nominative and the accusative case of the lemma “παιδί” (10):

(10) Ο Νίκος φεύγει με το παιδί.

‘Nikos leaves with the child.’
The ambiguity of παιδί between the nominative case (labeled a: 1) and the accusative case is presented in Chart 4 below where the valid solution is given in bold. In Chart 2 the F-structures of both the possible solutions are presented. However, the solution marked with red [solution (a: 2)] is preferred by the given Grammar. Finally, in Chart 3 the single valid solution is given.

Chart 1: Parse Tree

Chart 2: F-structure chart
XLE uses a form of Optimality Theory that allows the grammar writer to indicate that certain constructions or lexical items are preferred, not desirable or totally unacceptable (Miriam Butt et al., 1999), (Yehunda N.Falk, 2001).
3. Templates and Lexical Rules

3.1 Lexicon section

The XLE parser indicates the words of each string as lexical entries. The lexical entries come from a variety of different sources. In an overall grammar design, one might want to categorize words into the following types:

1. Core file lexicon.
2. A lexicon file with technical terms and words with special uses in specific context.
3. Lexicon files of semi-automatically generated verbs, nouns, adjectives etc.
4. A lexicon that includes the tags assigned by the morphological analyzer.

We will discuss only the first category, the core file lexicon that is hand-coded by the grammar writer. The writer should use the specific following format and the XLE annotation system:

Examples of the three main Categories of predicates:

- κοριτσι (girl) N (^NUM) = sg
  (^PRED) = ‘κοριτσι’

- ομορφο (beautiful) ADJ (^NUM) = sg
  (^PRED) = ‘ομορφο’

- διαλέγει (chooses) V (^TENSE) = present
  (^PRED) = ‘διαλέγω’

The examples above show that in XLE each lexical entry has the indicated specific structure and conveys the following information:

- Base form: the word e.g. ‘κοριτσι’.
- Category: Each one of the lexical items belongs to at least one syntactic category. In the examples above, ‘κοριτσι’ is a noun (N), ‘το’ an article, ‘ομορφο’ an adjective (Adj) and ‘διαλέγει’ a verb (V)\(^1\)

\(^1\) The categorical logic grammars have been investigated for usage in syntactic analysis by Moortgat (1987) and Morrill (1988) and for semantic analysis by Carpenter (1989).
• Attributes and Values: Each lexical entry has a list of them. For example, in
the formula \( (^{\text{NUM}}) = \text{sg} \), \text{NUM} is the feature and \text{sg}(\text{singular}) is its value.

This is the simplest way to define a lexical entry, but XLE grammars provide the
*template section* to help avoid redundancies in the lexicon. All the attributes and
values can be called via templates. Templates are macros that express linguistic
generalizations. A name is assigned to a complex formula and can be used in the place
of that formula in larger expressions. The symbol “@” followed by the name of a
template is used each time the particular template should be called. A simple
template is given in (11) and a complex one in (12). Consider the lexical entry below:

\[
\text{διαλέγει} \text{(chooses)} \quad V \quad (^{\text{TENSE}}) = \text{present} \quad
(^{\text{PRED}}) = '\text{διαλέγω} <(^{\text{SUBJ}}, (^{\text{OBJ}})>' \quad (11)
\]

If the template (11) is used, the lexical entry ‘διαλέγει’ above will be written as
following:

\[
\text{διαλέγει} \quad V \quad @ \text{(TENSE pres)} \quad
(^{\text{PRED}}) = '\text{διαλέγω} <(^{\text{SUBJ}}, (^{\text{OBJ}})>' \quad (12)
\]

Template (12) declares that each transitive verb, needs a subject and an object in
order to be correctly drafted. If template (12) is used, the lexical entry ‘διαλέγει’ will
be written as follows:

\[
\text{διαλέγει} \quad V \quad @ \text{(TENSE pres)} \quad 
@ \text{(TRANS διαλέγω)}
\]

Templates are really useful for describing verb sub categorization. The lexicon is
organized in such a way that each tuple representing subcategorisation properties
corresponds to a template. We have already seen the template for the transitive verbs
while the template for the intransitive verbs would be (13):

\[
\text{INTRANS}(P) = (^{\text{PRED}}) = 'P<(^{\text{SUBJ}})'><'. \quad (13)
\]
If our lexicon contained just these two templates, sentence (14) would be correctly parsed, but (15) would not be parsed, because in the lexicon the verb ‘διαλέγει’ presupposes an object.

(14) Η Μαρία διαλέγει βιβλία.
    ‘Maria chooses books.’

(15) Η Μαρία διαλέγει.
    ‘Maria chooses.’

Of course, it is actually the case that often transitive verbs are used as intransitives, for instance activity verbs in a habitual interpretation. The OPT-TRANS template, a combination of the TRANS and INTRANS template, is used for these verbs. It is based on disjunction. As explained before, the expressive means that are available for building context free grammar rules are also available for building templates.

3. OPT-TRANS(P) = "template for verbs that can be trans or intrans"
   \{ @(TRANS P)  
   \ | @(INTRANS P) \}.  

So, an opt-trans verb such as ‘τρωω’ (16), (17) will be correctly parsed.

(16) Το παιδί τρώει μήλα.
    ‘The child eats apples’

(17) Το παιδί τρώει.
    ‘The child eats.’

4. COUNT-NOUN(P N) = @(PRED P)
   (^ NUM) = N
   (^ PERS) = 3
   (^ NTYPE) = count.

It is well attested that countable nouns follow a general pattern that can be encoded with a template instead of an expand list of attributes and values. As result a lexical entry like ‘μπανάνα’ will be simply described as below: μπανάνα (banana) N * @(COUNT-NOUN μπανάνα sg).
Moreover, the example above demonstrates that templates can call other templates. For instance, the template that is called by the COUNT-NOUN template is:

\[ \text{PRED}(P) = (^{^\text{PRED}}) = 'P'. \]

One of the most complicated and important usage of the templates is that of encoding *lexical rules*. Generally speaking, a lexical rule is a form of syntactic rule used by several “lexicalistic” theories of natural language syntax such as HPSG, GPSG, LFG. These rules map lexical entries to lexical entries by either adding new schemata or by modifying the designators appearing in existing schemata. They offer valuable parsimony as they capture productive generalizations resulting to a reduction of the size of the lexical data-base.

Here, we will focus on Lexical rules in LFG where they serve the purpose of expressing lexical regularities. They create new lexical entries from existing ones. These rules can also be used to encode generalizations across grammatical function alternations.

We first present the schema for DATIVE-SHIFT that is used with ditransitive verbs. The idea is that the lexicon contains the verb

\[ \text{give} < (^{\text{SUBJ}}) (^{\text{OBJ}}) (^{\text{OBLTO}}) \]

If the DATIVE-SHIFT schema (18) is used the parser is able to parse (19) without any additional lexical entry for ‘give’:

(18) \[ \text{DAT-SHIFT (SCHEMA)} = \{ \text{SCHEMA} \]

\[ \quad \quad \quad \quad \quad (^{\text{OBL PCASE}}) = c \text{ TO} \]

\[ \quad \quad \quad \quad \quad \mid \text{SCHEMA} \]

\[ \quad \quad \quad \quad \quad \quad (^{\text{OBJ}}) \rightarrow (^{\text{OBJ2}}) \]

\[ \quad \quad \quad \quad \quad \quad (^{\text{OBL}}) \rightarrow (^{\text{OBJ}}) \}. \]

(19) \[ \text{Mary gave the dog a bone.} \]

The changes that are imposed by this schema are:

- The OBJ (a bone) is turned to OBJ2 in the sentence (19).
Lexical Rules can interact unproblematically. So, the DAT-SHIFT schema can interact with the PASS-SCHEMA. Passive in LFG is entirely lexical, which means that there is no syntactic movement. So in order to parse example (19) without introducing any new lexical entry for the passive voice of ‘give’, we will use the schema PASS (21):

(20) The dog was given a bone by Mary

(21) PASS(SCHEMATA) = { SCHEMATA |SCHEMATA
   (^ PASSIVE) = +
   (^ PARTICIPLE)=c past
   (^ OBJ)-->(^ SUBJ)
   (^ OBJ2) --> (^ OBJ) "for dative shift"
   { (^ SUBJ)-->(^ OBL-AG)
   |( ^ SUBJ)-->NULL} }.

So, in the passive form of a verb, the object is turned into the subject designator and the subject converts either to the oblique agent designator or it is given as the NULL symbol to indicate the absence of that grammatical function. The PASS rule applies to transitive verbs.

The template for ditransitive verbs:
DITRANS (P) = (^ PRED)=P<(^ SUBJ) (^ OBJ) (^ OBJ2)>
in combination with the PASS schema is given like that:
DITRANS (P) = @(PASS (^ PRED)=P<(^ SUBJ) (^ OBJ) (^ OBJ2)>).

And if the DATIVE-SHIFT template is taken into account:
DITRANS (P) = @(PASS @(DAT-SHIFT (^ PRED)=P<(^ SUBJ) (^ OBJ) (^ OBL-TO))>).

Although, the Dative-Shift template is useful in languages such as English and German, it cannot be used in Modern Greek. As a matter of fact, the goal argument of ditransitives in Modern Greek is marked either with a preposition (22), (23) or with
the genitive case (24) and never appears as the subject of the passive form of the verb (Fotopoulou, 2005).

(22) Ο Πέτρος έστειλε ένα δώρο στην Μαρία.
   ‘Petros send the gift to Maria.’
(23) Ένα δώρο στάλθηκε στην Μαρία από τον Πέτρο.
   ‘A present was sent to Maria by Petros.’
(24) Ο Πέτρος έστειλε της Μαρίας ένα δώρο.
   ‘Petros send Mary a gift.’
(25) *Η Μαρία στάλθηκε ένα δώρο από τον Πέτρο.
   ‘Maria was sent a gift by Petros’

Nevertheless, a certain alternation occurs despite the fact that the goal never appears as the subject of passivisation. We take it that the goal never becomes an object that is eventually visible by the passivisation lexical rule but an OBJ-GENITIVE or OBJ2 (as we decided to call it) that is not visible by the passivisation rule for Modern Greek.

After the previous verification, in the passivization of a sentence the OBJ2 will be NULL and in Dative shift the OBJ will be also NULL. So, the PASS Schema and the DATIVE-SHIFT Schema for Modern Greek will be:

(26) PASS(SCHEMATA) = { SCHEMATA
|SCHEMATA
(^ PASSIVE) = +
(^ PARTICIPLE) = c past
(^ OBJ) --> (^ SUBJ)
(^ OBJ2) --> NULL "for dative shift"
{ (^ SUBJ) --> (^ OBL-AG)
|(^ SUBJ) --> NULL }

And

(27) DAT-SHIFT (SCHEMATA) = { SCHEMATA
(^ OBL PCASE) = c TO
|SCHEMATA
(^ OBJ) --> (^ OBJ)
(^ OBL) --> (^ OBJ2) }. 
We present the XLE result of parsing sentence (28) in the following charts with the use of the templates (26) and (27). Only the entry

δίνω < (^SUBJ) (^OBJ) (^OBLTO)>

was available to the parser,

(28) Το πορτοκάλι δόθηκε στο παιδί από τον κηπουρό.

‘The orange was given to the child by the gardener.’

Chart 1: Parse Tree
Chart 2: F-structure chart

Chart 3: F-structure #1
As it can be seen in the Chart 4, XLE chose the a:1 and the b:2, as the correct options for the parsing of the given sentence.

3.2 Grammatical Functions

We first talked about the grammatical functions of LFG in Section 1.1. Here we will take up the issue in more detail. We remind the reader that a grammatical function is a particular kind of dependency. A verb could have dependants such as subject and object, and each one of them has a different relation with the verb. LFG maintains that ‘subject’ and 'object' are distinct grammatical functions.

There are three basic groups of grammatical functions:

1. Subcategorisable functions: SUBJ, OBJ, OBJ-TH, COMP, XCOMP

2. Adjunct (non-subcategorisable) function: ADJ, XADJ

3. Grammaticized discourse functions: FOCUS, TOPIC
Subcategorisable functions

These are the functions that are required by a verb predicate in order to ensure that a sentence with this predicate as a head is well-formed.

a) **SUBJ:** One of the two main constituents of a clause. It declares the origin of the action or the undergoer of the state.

   (29) Το παιδί έφυγε.
   ‘The child left.’

   SUBJ V

b) **OBJ:** Direct objects are the part of the clause that shows what or whom the verb subject is acting upon.

   (30) Ο Νίκος αγόρασε μήλα.
   ‘Nick bought apples.’

   SUBJ V OBJ

c) **OBJ2:** As discussed in Section 3.1, Modern Greek ditransitive verbs, such as “δίνω (give), διαβάζω (read)” are allowed to take two objects. The first one is the direct object and the second one is the OBJ2, the thematically restricted object. This is always marked with the genitive case and never appears in a passive form of the verb.

   (31) Ο Νίκος έδωσε παιχνίδια του παιδιού.
   ‘Nick gave toys to the child.’

   SUBJ V OBJ OBJ2

Therefore, there is an open issue whether Modern Greek has an OBJ2 as English does (Markantonatou, 1994; Anagnostopoulou, 1994). In our grammars we take care that the Greek OBJ2 is marked with the genitive case and that it blocks passivisation.

At this point, a short excursion to Prepositional Phrases (PPs) is due. “Prepositions, as well as prepositional phrases (PPs) and markers of various sorts have a mixed history in computational linguistics (CL), as well as related fields such as artificial intelligence, information retrieval (IR), and computational psycholinguistics” as noted by Timothy Baldwin et al. (2009) These small but really
significant words introduce phrases that can indicate the actor and agency, manner and measure, instrument and intention, goal and source, range and reference. PPs generally act as complements and adjuncts of noun phrases and verb phrases. For instance:

(32) Ο Γιώργος από την Αθήνα παίζει τένις.

‘George from Athens plays tennis.’ (Adjunct in a noun phrase)

(33) Τρέχει κάτω από τη γέφυρα.

‘Runs under the bridge.’ (Adjunct in a verb phrase)

(34) Η Κάτια δίνει δώρα στη Μαρία.

‘Katia gives gifts to Mary.’ (Oblique complement of a verb)

(35) Οι γονείς μιλάνουν με τα παιδιά.

‘Parents are arguing with the children.’ (Complement of a verb)

LFG distinguishes two types of prepositions (Markantonatou, 2011):

A. Preposition is a predicate with its own predicate domain.

The preposition contributes semantics that is not specified by the verbal predicate and works like a modifier (MOD) to it.

The grammar rules needed to be parse example (36) are (37):

(36) Η Μαρία έφυγε με το μετρό.

‘Maria went by the metro.’

(37) S \rightarrow NP \quad VP

↑SUBJ=↓

VP \rightarrow V \quad (NP) \quad (PP) *

↑OBJ=↓ \quad ↑€(↑MOD)

B. Preposition is not a predicate with its own predicate domain. These are a subset of the so called OBL arguments of LFG.

OBL: Oblique arguments are typically marked with prepositions or Cases expressing θ-roles. Oblique arguments are arguments with an explicit indication
of the thematic role. This indication is most often encoded by means of prepositions. The set of oblique arguments includes such grammatical functions as OBL-GOAL, which in English is typically encoded with a Prepositional Phrase (PP) introduced with the preposition to and in Greek with a PP introduced with the preposition “to-σε”, BENEFACTIVE (introduced with the prepositions for-για, for English and Greek respectively), SOURCE (from-από), INSTRUMENT (with-με), AGENT (by-από), LOCATION (with various prepositions) etc. This notation emphasizes the fact that these grammatical functions belong to the oblique family of functions. As a group, they are referred to as OBLθ.

The fact is that there are several such thematic roles and markers and if each one of them was encoded as a separate OBLθ, a proliferation of minimally different phrase structure rules would occur. LFG uses the PCASE feature as a variable ranging over such roles. Thus, in order to parse

(38) Ο Νίκος δόθηκε ένα λουλούδι στη Μαρία.

‘Nikos gave a flower to Mary.’

the following grammar and lexicon fragment is defined:

$$\text{VP} \rightarrow V \ (\ NP) \quad \text{PP}$$

$$↑\text{OBJ}=↓ \quad ↑(↓\text{PCASE})=↓$$

$$\text{PP} \rightarrow P \quad \text{NP}$$

$$↑=↓ \quad ↑\text{OBJ}=↓$$

$$Δίνω \ V \ [\text{PRED}=‘ΔΙΝΩ’< (↑\text{OBJ}) (↑\text{OBLgoal})>]$$

$$στη \ \text{Prep} \ [\text{PRED}=‘ΣΕ’, ↑\text{PCASE}=\text{OBLgoal}]$$

The XLE allows the grammar author to express the different thematic roles

a) By declaring it separately like the OBL-AG as it is used only in Passive voice. Example (14) that is discussed in the Pass-schemata, uses OBL-AG.

b) By using only one OBL Grammatical Function and each time instantiate the PSEM variable with a value drawn for a set of names of thematic roles.
In XLE platform, PPs are marked by the feature [PTYPE ], which could be “sem” or “nosem”: “sem” /semantic if the PP belongs to the second category above and nosem/ non semantic if the PP belongs to first one.

- **COMP:** Closed complement is the clausal argument that has its own subject.

  (39) Εγώ θέλω η Μαρία να έρθει σπίτι.
  ‘I want Mary to come’

  In the above example the main and the subordinate verbs have different subjects.
  a. εγώ ‘I’ = SUBJ of the main verb “θέλω”
  b. η Μαρία ‘Maria’ = SUBJ of the subordinate verb “έρθει”

  So, the lexical entry of the verb ‘θέλω’ would be:

  θέλω  V * (^PRED) = ‘ΘΕΛΩ < (^SUBJ) (^OBJ) (^COMP) ’

  (^COMP SUBJ) = ‘PRO’

  “(^PRED) = ‘WANT < (^SUBJ) (^COMP) ’

  (^COMP SUBJ) = ‘PRO’.”

  (40) Ήθελε να κοιμάται.
  ‘He/she/it wanted to sleep’

Example (40) would be considered ungrammatical by an English speaker. English language is a non pro- drop language. Nonetheless, subject pronouns are almost always dropped in imperative sentences (e.g., Study now). On the other hand, subject pronouns can be often omitted in Greek. That could happened, when they are in some sense pragmatically inferable (Joseph, 1994). The lexical entry in (40) is appropriate for parsing a pro-drop sentence. In charts 2 and 3 it is obvious the unification of the subject ‘Pro’ with the use of the number (1) in front of the subject.
Chart 1: Parse Tree

```
"ηθελε να κοιμηται"

CS 1: S:28
   VP:26
      V:2 VPcomp:25
         ηθελε:1 COMPL:4 VP:19
            να:3 V:6
               κοιμηται:5
```

Chart 2: F-structure chart

```
F-structure chart
"ηθελε να κοιμηται"

[ PRED "WANT<1-SUBJ:PRO>, [3:KOIMANAI]>'
  SUBJ [ PRED 'PRO'
     CASE NOM, NUM Sg, PERS 3 ]
  XCOMP [ PRED 'KOIMANAI<1-SUBJ:PRO>'
     SUBJ [1-SUBJ:PRO]
     COMPL FORM να, MOOD indicative, TENSE pres]
  MOOD indicative, TENSE past
```
**XCOMP:** The open predicate complement is a predicate argument with no overt subject predication. Extra information is needed in order to specify the subject. This information in LFG is encoded with the so called functional equations like the one below that specifies that the missing subject of the XCOMP is the same with the subject of the mother f-structure of the XCOMP:

\[ \uparrow \text{XCOMP \, SUBJ} = \uparrow \text{SUBJ} \]

This piece of information may be part of the description of a lexical entry (a lexical property) or part of a Phrase Structure rule (structural property).

Example of lexical property:

(41) Ἡ Μαρία ξέρει να πλέκει κάλτσες

‘The Maria knows to knit socks.’

“Maria knows how to knit socks”.

The predicate “Μαρία” is the subject of the main verb “ξέρει” and at the same time is the subject of the verb “πλέκει”.

The lexical entry of the verb “ξέρει” will be:

\[ ξέρει \, V \, (\uparrow \text{PRED}) = 'ξερω \, <(\uparrow \text{XCOMP}) \, >(\uparrow \text{SUBJ})' \]

\[ (\uparrow \text{SUBJ}) = (\uparrow \text{XCOMP \, SUBJ}) \]
The F-structure of the sentence (41) will be:

\[
\begin{align*}
\text{PRED } \xiέρει & (\uparrow \text{XCOMP}) (\uparrow \text{SUBJ}) > \\
\text{SUBJ } [\text{PRED } '\text{Μαρία'}] & [1] \\
\text{TENSE } \text{PRESENT} \\
\text{XCOMP} [\text{PRED } '\piλέκει'] & (\uparrow \text{SUBJ}) > \\
\text{SUBJ } [1] \\
\text{OBJ } [\text{PRED } 'κάλτσες']
\end{align*}
\]

In the example above, the SUBJ of the verb “ξέρει” (knows) functionally controls the SUBJ of the subordinate XCOMP. This means that the f-structure that is the value of the SUBJ of “ξέρει” (knows) must be the same as the SUBJ of the subordinate XCOMP. I am inclined to adopt the thesis that aspectual and volitional verbs control the subject of their sentential complement. Apart from the aspectual verbs, such as “ξεκινώ-begin”, “τελειώνω-finish”, “συνεχίζω-continue”, “σταματάω-stop” that control the subject of their comp, the category of the verbs that express knowledge, such as “know- ξέρω”, “learn-μαθαίνω” (see the example above).

Example: (42) Η Μαρία αρχίζει να μαθαίνει Αγγλικά.

“The Mary starts to learn English.”

The lexical entry of the verb “αρχίζει” will be:

\[
\begin{align*}
\text{αρχίζει } V (\uparrow \text{PRED}) = 'αρχίζω<(\uparrow \text{SUBJ})(\uparrow \text{XCOMP})>' \\
(\uparrow \text{SUBJ}) = (\uparrow \text{XCOMP SUBJ})
\end{align*}
\]

The output of the XLE for a clause (43) with “άρχισε” as a main verbal predicate is.

(43)Το μωρό άρχισε να τρώει φρούτα.

“The baby started eating fruits.”
Chart 1: Parse Tree

```
"το μπορεί αρχίσε να τσώσει φρουτά"

CS 1: S:45
    NP:17 VP:44
    D:2 N:4 V:5 VPcomp:43
    to:1 ωρα:3 αρχίσε:5 COMPL:8 VP:36
    vo:7 V:10 NP:34
    τσώσε:9 N:12 φρουτά:11
```

Chart 2: F-structure chart

```
F-structure chart

"το μπορεί αρχίσε να τσώσει φρουτά"

PRED 'ΑΡΧΙΖΩ<1:ΜΟΡΟ>, [7:ΔΙΑΛΕΞΩ]>
PRED 'ΜΟΡΟ'
SUBJ [CASE NOM, DEF +, NTYPE count, NUM SG, NUMBER SG, EERS 3]
    PRED 'ΔΙΑΛΕΞΩ<1:ΜΟΡΟ>, [11:ΦΡΟΥΤΑ]>
    SUBJ [1:ΜΟΡΟ]
XCOMP OBJ 11 [CASE ACC, NTYPE count, NUM PL, EERS 3]
    COMPL_FORM vo, MOOD indicative, TENSE pres
    COMPL FORM vo
    MOOD indicative, TENSE past
```
Given that lexically specified functional control is related with the existence of infinitives in a language, the question is whether Modern Greek has lexically specified functional control given that infinitives have ceased to exist in this language (Philippaki-Warburton and Veloudis 1984, Philippaki-Warburton 1994). Infinitive is a grammatical term used to refer to certain verb forms but the fact is that there is no single definition applicable to all languages. In traditional descriptions of English, the infinitive is the basic dictionary form of a verb when it is used with or without the particle to. In many other languages the infinitive is a single word, often with a characteristic inflective ending, such as “lieben” ((to) love) in German or “ζειν” ((to) live) in Ancient Greek. Infinitives were attested in Ancient Greek, Latin and other languages, but they are marginalized in fossilized grammatical constructions in Modern Greek. On the other hand, a modal construction of a complementizer and an independent main clause can be identified in use in this language. This modal construction has been analyzed as a potential Balkanism, as noted by Andreas Ammann and Johan van der Auwera (2004).

However, problems concerning the design of a grammar that includes functional control appear only when an object controls the subject of the subordinate clause (as is the case with English examples such as “I persuaded John to leave me alone.”) and not when a main clause subject controls the subordinated clause one.
Adjunct function

- **ADJ**: Adjunct is the default grammatical function for non-subcategorized arguments. Adjuncts can be:
  
  ✓ Adverbials, such as “γρήγορα (quickly)
    
    For instance, Ο Νίκος τρέχει γρήγορα.
    
    “Nick runs quickly.”
  
  ✓ PPs, such as “με το αυτοκίνητο (by car)”
    
    For instance: Ο Νίκος έφυγε με το αυτοκίνητο.
    
    “Nick left by car.”
  
  ✓ NPs, such as “την Πέμπτη”
    
    For instance, Ο Νίκος έφυγε την Πέμπτη.
    
    “Nick left on Thursday.”

Below I will present an XLE output for the sentence:

(44) Ο Νίκος έφυγε με το αυτοκίνητο
    ‘Nick left by car.’

XLE first notices the ambiguity of the word form “αυτοκίνητο”:

```
True
a:1 (f7:AUTOKINHTO CASE)=NOM
a:2 (f7:AUTOKINHTO CASE)=ACC
```

The charts are:

**Chart 1: Parse Tree**

```plaintext
"ο Νίκος έφυγε με το αυτοκίνητο"

CS 1: 5:39

νικος:17
νεφερ:35

νικος:17
νεφερ:35

το:9 autokeinhto:11
```
One more example of adjunct parsing will be presented illustrating the agreement between an adjective and a modified noun.

NP --> (D)
   AP*: ! $ ( ^ ADJUNCT)
       (^GENDER)=(!GENDER)
       (^CASE)=(!CASE)
       (^NUMBER)=(!NUMBER); N.

** AP --> ADV*: ! $ (^ADJUNCT); Adj*. 
So, with this restriction example (45) won’t be parsed in contrast to example (46).

(45) Το όμορφη κορίτσι ήρθε.
‘The beautiful girl came’

(46) Το αγόρι έφαγε το πράσινο μήλο.
‘The boy ate the green apple.’

Chart 1: Parse Tree

Chart 2: F-structure chart
**XADJ**: It is an open predicate adjunct. This adjunct is controlled by the subject of the main clause or by the object.

For instance,

(47) Ο Γιώργος απάντησε στο τηλέφωνο τρώγοντας.

‘George answered the phone while eating’

In the example (47) George is the same person that both answered the phone and was eating. This is a case of structural control, that is, a control equation is applied on the phrase structure rule and ensures subject identity.

We remind the reader that, unlike governable grammatical functions, more than one adjunct function can appear in a sentence:

(48) Ο Νίκος έτρωγε μακαρόνια σπίτι μου χθες.

‘Nikos was eating spaghetti at my place yesterday.’
Since the ADJ function can be multiply filled, its value is a set of f-structures:

\[
\begin{align*}
\text{PRED} & \colon \text{ΤΡΩΩ < SUBJ, OBJ>}
\text{SUBJ} & \colon \text{PRED } \text{Νίκος}
\text{OBJ} & \colon \text{PRED } \text{μακαρόνια}
\{ \text{PRED} \colon \text{χθες} \}
\{ \text{PRED} \colon \text{σπίτι} \}
\end{align*}
\]

More than one XADJ phrase can appear in a single sentence, but the value of the XADJ feature is also a set of f-structures.
References

Θεοφανοπούλου-Κοντο Δ. 1989, Μετασχηματιστική σύνταξη: Από την θεωρία στην πράξη, Εκδόσεις Καρδαμίτσα, Αθήνα.


Μαρκαντωνάτου Σ. 2011. Μαθήματα 1 έως 10 με τίτλο «Γραμματικοί Φορμαλισμοί, Π.Μ.Σ. "Τεχνογλωσσία"». http://www.ilsp.gr/el/profile/staff?view=member&task=show&id=38

Τζάρτζανος Α. 1946. Νεοελληνική σύνταξη (της κοινής δημοτικής), β’ έκδοση. Θεσσαλονίκη: ΑΦΟΙ Κυριακίδη, ανατύπωση 2002.


Φωτοπούλου Α. 2001b. «Προσδιορίζοντας μία κατηγορία δίπτωτων ρημάτων συναλλαγής (datifs): "λεξικοποιημένη" και "εκτενής" δοτική». Πρακτικά της 21ης Συνάντησης του Τομέα Γλωσσολογίας της Φιλοσοφικής σχολής του ΑΠΘ.


Andreas Ammann and Johan van der Auwera, 2004, Complementizer-headed main clauses for volitional moods in the languages of South-Eastern Europe: a
Balkanism?, Center for Grammar, Cognition and Typology University of Antwerp (UIA), [http://webhost.ua.ac.be/cgct/publications/L-AmmAuw.pdf](http://webhost.ua.ac.be/cgct/publications/L-AmmAuw.pdf)


**Mary Dalrymple, 2002**, Lexical Functional Grammar, Centre for Linguistics and Philology, Oxford, United Kingdom, [http://users.ox.ac.uk/~cpgl0015/lfg.pdf](http://users.ox.ac.uk/~cpgl0015/lfg.pdf)


