university of groningen / faculty of science and engineering

Prolog vs XLE: Parsing a Fragment of the Romanian Language

Bachelor's Project Thesis

Andreea-Ioana Tudor, s4020960, a.tudor.3@student.rug.nl,
Supervisor: Dr. Stephen Jones

Abstract: Several elementary structures in the Romanian language exhibit complex linguistic phenomena. The clitic doubling phenomenon, among others, has received much interest in the literature not only for its complexity and historical inheritance, but also due to cross-linguistic differences. Even though the theoretical background has been extensively studied, practical solutions that allow an adequate implementation of Romanian direct object constructions are lacking. This study aims to compare a parser for a fragment of the Romanian language built in Prolog using Definite Clause Grammars (DCGs) and a second parser written in Xerox Linguistic Environment (XLE) under the Lexical-Functional Grammar (LFG) formalism. Designing the models requires the development of a complex set of rules that account for clitic doubling and other challenging linguistic features in Romanian. The analysis of some substantial linguistic phenomena is presented along with their implementation in the aforementioned frameworks. Both parsers can identify whether the input is grammatical or not. When it comes to encoding complex linguistic phenomena, Prolog offers a more accessible and adaptive solution for parsing. At the same time, XLE provides an environment that displays more valuable insights for linguistic analysis.

1 Introduction

"Human knowledge is expressed in language. So computational linguistics is very important."
– Mark Steedman, ACL Presidential Address (Steedman, 2008)

Computational linguistics emerged as a field of Artificial Intelligence (AI) aiming to allow computer systems to generate and interpret natural language. Three of the main application classes are represented by machine translation, information retrieval and human-machine interface (Grishman, 1986). Besides the aforementioned applications, a natural function of this field is the testing of grammars proposed by theoretical linguists.

Machine learning techniques used in applications of the field have shown major accomplishments. For instance, Google Translate represents the state-of-the-art in machine translation. Other widely used application examples include chatbots, virtual assistants (e.g., Siri, Amazon Alexa), and even grammar checkers (e.g., Grammarly - only available in English). However, these systems display limitations, as understanding and processing natural language is difficult due to challenging linguistic phenomena (Coheur, 2020). Traditional parsing techniques might offer an additional layer of understanding complex language features. Therefore, we propose a traditional parsing approach for a fragment of Romanian and present the difficulties raised by the language.

The aim of this study is to design and compare two parsers in Prolog and XLE for a fragment of the Romanian language. Romanian is an understudied language that displays several complex linguistic phenomena such as clitic doubling, which might lead to challenges in the language modelling and the implementation of the systems. We begin by presenting a brief overview of the frameworks (Sections 1.1 and 1.2) and describing some important phenomena present in the Romanian language (Section 1.3) that are relevant for this study.

1.1 Prolog

Context-Free Grammar (CFG) represents a declarative formal system for modelling constituent structure in natural languages. The CFGs defined by Noah Chomsky provide a simple and efficient parsing formalism and are the most common declarative representations of grammatical structure (Schubert, 2020). They consist of sets of rules that define what strings are legal and their grammatical structure (Blackburn et al., 2006). An example of a simplified CFG for a fragment of English is shown in Figure B.1, and a parse tree of a sentence it can generate is displayed in Figure B.2.

Prolog is one of the first logic programming languages. Its original usage field is natural language processing. Given a CFG, Prolog allows defining a set of predicates that encode the logic of the grammar. Definite clause grammars (DCGs) represent a notation provided in Prolog for writing grammar rules, hiding the underlying difference list variables (Blackburn et al., 2006).

1.2 XLE

Lexical-Functional Grammar (LFG) represents a linguistic theory aiming to model the complex linguistic information that native speakers possess. The fundamental assumption of LFG is that this multifaceted linguistic information, connected by functional constraints, can model and describe best the language (Börjars et al., 2019; Dalrymple, 2001).

Phrase structure rules define the possible configurations of phrase structures in a language. The lexicon describes the words' features and requires a different entry for each word accompanied by the lexical specifications. An example of rules and lexicon for a short fragment of English is illustrated in Figure B.3. Two main levels of representation in LFG are the c-structure and the f-structure. The c-structure is a depiction of the constituent structure, licensed by the phrase structure rules. The f-structure (functional structure), represented as a table of attributes and values, embeds the grammatical functions (Dalrymple, 2001). Figure B.4 shows the c-structure and the f-structure for a sentence generated using the rules and lexicon in Figure B.3.

Xerox Linguistic Environment (XLE) is a computational environment meant to assist in efficiently parsing and generating language grammars (Crouch et al., 2011). It provides several grammatical notations that follow the expressive LFG formalism, facilitating the writing of grammars. The language is encoded using phrase structure rules and lexical specifications for the words, and XLE can display the c-structure and the f-structure for every possible parse. Butt et al. (1999) provide a "cookbook" for writing grammars in XLE, presenting analyses of language and their implementation alongside engineering-related issues.

1.3 Relevant linguistic phenomena in Romanian

The models discussed in this project cover a few elementary phrase structures of the Romanian language. These phrase structures include basic noun phrases and verb phrases with subject and (direct) object arguments. Even though they represent fundamental structures in the language, they exhibit some complex linguistic phenomena. The current subsection presents a comprehensive description and analysis[1] of the aforementioned phrase types.

1.3.1 Nouns

English nouns encode number features (singular/plural). In addition to that, in languages such as French, German (Butt et al., 1999) and Romanian, nouns also encode gender and case. Romanian nouns are divided into three genders: masculine, feminine and neuter. Neuter gender morphologically identifies with the masculine in the singular and the feminine in the plural (Dindelegan, 2013). Number and gender

[1]The examples provided in this paper follow the Leipzig Glossing Rules, which represent the standard convention for presenting linguistic data.

can be marked by an inflectional ending, but also by the definite enclitic article in the definite declension, based on their case, as shown in Table A.1 (Dindelegan, 2013).

The noun case-marking system is inherited from Latin and involves four cases: nominative, accusative, genitive and dative. Unlike pronouns (singular, first and second person), Romanian nouns have the same form in nominative-accusative and genitive-dative. Besides the aforementioned features, nouns are also encoded with a specific NTYPE (Butt et al., 1999), which makes a distinction between countable nouns (e.g. *dog, apple*), mass nouns (e.g. *water*) and proper names.

1.3.2 Determiner Phrases

Romanian possesses a definite article and an indefinite article. The definite article is enclitic and varies in gender, number and case, as presented in Table A.1. The indefinite article also expresses number, gender and case (Table A.2) (Dindelegan, 2013).

The demonstratives in Romanian are distinguished based on proximity, namely proximal demonstratives (*acesta* 'this') and distal demonstratives (*acela* 'that'). They inflect for number, gender and case, and function as prenominal and postnominal (1) (Dindelegan, 2013). The prenominal demonstratives display a short form and require an articleless noun (1a), while the postnominal demonstratives have a long form, and the corresponding noun must be definite (1b). The forms of the proximal demonstrative *acest/a* ('this') are presented in Table A.3.

(1) a. ***acest*** *băiat*
this.SHORT-FORM boy
'this boy'

b. *băiatul* ***acesta***
boy.DEF this.LONG-FORM
'this boy'

1.3.3 Subject

According to Dindelegan (2013), the subject and the verb impose restrictions on one another: the subject is assigned the nominative case by the verbal inflection, and the finite verb must agree with the subject in number and person. A specific characteristic of the Romanian language is represented by the fact that the subject is not mandatory, as it is a pro-drop language. (Dindelegan, 2013).

1.3.4 Direct object clitic doubling

The clitic doubling is a construction including the co-occurrence of a clitic with a nominal phrase, and it also exists in other Romance languages. However, Romanian behaves differently, as several rules in the language determine whether the clitic doubling is required, optional or not allowed in specific constructions. In Romanian, the clitic and the nominal phrase are selected by the same transitive verb when the phenomenon occurs with direct objects (Dindelegan, 2013). Table A.4 shows the forms of the Romanian direct object clitics. Some relevant semantic and pragmatic properties that trigger clitic doubling are animacy/humanness, referentiality, specificity and definiteness (Alexandru and Silvina, 2020).

When common nouns act as direct objects, there are two possible constructions for the doubling clitic patterns. The direct object can be preceded by the PE marker and be doubled by an accusative clitic (2a), or it can appear without the PE marker, in which case the clitic doubling is not allowed (2b) (Babyonyshev and Marin, 2005). Hence, the clitic doubling only occurs when the direct object follows the accusative marker PE.

(2) a. **(Îl) văd *(pe) băiat.*
*(CL.3SG.M) see.1SG *(PE) boy.SG.M
'I see the boy.'

b. *(*Îl) văd (*pe) băiatul.*
(*CL.3SG.M) see.1SG (*PE) boy.SG.M.DEF
'I see the boy.'

The sentences in example 2 show how the same noun, but in different morphological forms, can be used either with the direct object marker or without it. However, these patterns follow specific rules.

According to Dindelegan (2013), a nominal phrase in the direct object position must have the PE marker when it has the features [+personal/+animate] and [+specific] (3a). The same applies for proper names and pronouns (3b).

(3) a. *Fata* *(îl) *iubește* *(pe) *acest băiat.*
girl.SG.F.DEF *(CL.3SG.M) loves *(PE) this boy.SG.M
'The girl loves this boy.'

b. *Fata* *(îl) *iubește* *(pe) *Andrei.*
girl.SG.F.DEF *(CL.3SG.M) loves *(PE) Andrew
'The girl loves Andrew.'

Since clitics do not double objects that are not PE-marked (2), they never occur in sentences with non-human noun phrases (4a) or indefinites (4b) as direct objects. Moreover, an exception to the previous rule is given by definite direct objects without a determiner (with the definite inflectional article), which are not PE-marked regardless of the [+ human] and [+ specific] features (4c) (Barbu and Toivonen, 2018).

(4) a. *(*Îl) *mănânc (*pe) mărul.* [– animate] [+ definite]
(*CL) eat.1SG (*PE) apple.DEF
'I eat the apple.'

b. *(*O) *văd* *(*pe) o fată.* [+ animate] [– definite]
(*CL) see.1SG (*PE) a girl
'I see a girl.'

c. *(*O) *văd* *(*pe) fata.* [+ animate] [+ definite (inflectional)]
(*CL) see.1SG (*PE) girl.DEF
'I see the girl.'

The non-doubling clitics refer more freely, as they can refer to non-human NPs (5) (Barbu and Toivonen, 2018).

(5) *Am* *comandat un taxi.* *Îl* *aștept.*
have.1SG ordered a taxi.SG.N CL.3SG.M wait.1SG
'I have ordered a taxi. I am waiting for it.'

1.4 Previous work and current study

The clitic doubling phenomenon has received much interest in the literature not only for its complexity and historical inheritance but also due to compelling cross-linguistic differences. For instance, Romanian and Spanish acquire the object clitic doubling construction, while it is not present in French or Italian (Anagnostopoulou, 2006). The sentence in (6) shows the usage of clitic doubling of the indirect object, which differs from French (7), in which it is ungrammatical (examples retrieved from Anagnostopoulou (2006)).

(6) *Miguelito (le)* *regaló un caramelo a Mafalda.*
Miguelito (CL.DAT) gave a candy A Mafalda
'Miguelito gave Mafalda a candy.'

(7) *Jean (*lui) a donné des bonbons à Marie.*
Jean (*CL.DAT) gave candies À Marie
'Jean gave Marie candies.'

The theoretical background has been extensively studied. Some LFG approaches include Sadler (1997) on Welsh clitics, Mayer (2003) on clitic doubling in some Spanish dialects and Jaeger and Gerassimova (2002) on Bulgarian word order and clitics. However, practical solutions that allow an adequate implementation of Romanian direct object constructions are lacking.

The current paper mainly bases the grammatical analysis on Dindelegan (2013), which provides "the first comprehensive grammar in English of the present-day standard Romanian". For the clitic doubling, we follow the approach given by Barbu and Toivonen (2018), who propose a dual analysis for the Romanian clitics: as agreement markers and pronouns.

The research question this study aims to answer is whether a parser for a fragment of the Romanian language built in Prolog using DCGs can achieve the same performance in terms of time, accuracy and practicality as a parser obtained using XLE. In order to perform a fair comparison, the same set of sentences is used to test both parsers.

2 Methods

The fragment of Romanian language accounted for in this study is restricted to the phrase structures described in Section 1.3. The selection consists of fundamental structures commonly used in written and spoken language. Despite the frequency of usage, the complex underlying features establish a compelling foundation for this study. The current section provides an extended description of the language fragment boundaries (2.1) and grammar model (2.2), followed by an overview of how the models will be tested (2.3).

2.1 Model restrictions on the language fragment

The nouns used in the lexicon of the parsers bear the nominative-accusative case. They vary in number and gender, so the agreement between the noun and determiner can be checked. The definite article is enclitic; hence the definite nouns require a new lexical entry. On the other hand, the indefinite article and the demonstratives accompanying nouns form a new phrase type, determiner phrases (DPs).

The verbs are restricted to present tense, indicative mood. Using this form exclusively facilitates a less complicated lexicon while allowing several linguistic aspects to be checked. These aspects include subject-verb agreement and the correct use of direct objects and clitics. In order to test them, the lexicon contains transitive and intransitive verbs, ranging in number and person.

The Romanian language has a quite free word order variation. As in English, the standard word order in a sentence is subject-verb-object. However, other constructions, with the subject or the object in preverbal or postverbal positions, are also possible (Dindelegan, 2013). Since nouns have the same form when bearing nominative and accusative cases and the position of the subject and object can be interchanged, ambiguities may occur, as shown in (8).

(8) a. *Fata mănâncă un măr.*
the girl eats an apple
'The girl eats an apple.'

b. *Un măr mânăncă fata.*
an apple eats the girl
* 'An apple eats the girl.'

However, the language provides special rules to account for specific ambiguities. For example, when the object is in the preverbal position and has the [+ specific] feature, the doubling clitic is required, as shown in (9) (Dindelegan, 2013). Jaeger and Gerassimova (2002) account for direct object clitic doubling occurrences in Bulgarian, which resembles this exception in Romanian, using a discourse function approach. The discourse functions fall outside the scope of this project, however, their relevance is discussed

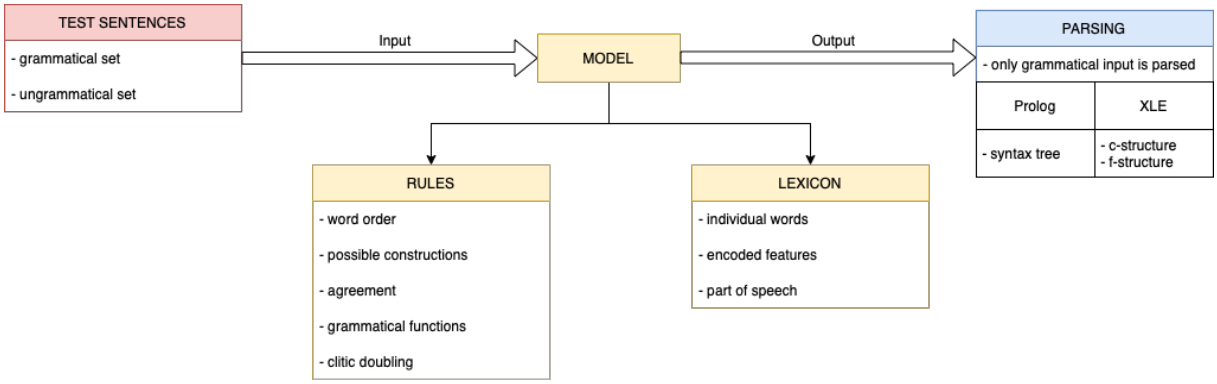

Figure 2.1: Conceptual model of the Prolog and XLE parsers

in Section 4.3. Hence, the object-verb constructions are not implemented in the parsers and excluded from the testing suite.

(9) a. * *Cartea citesc.*
book.SG.F.DEF read.1SG

b. *Cartea o citesc.*
book.SG.F.DEF CL.3SG.F read.1SG
'I am reading the book'.

2.2 Model representation

Prolog and XLE are used in this project for parsing grammatical constructions for the fragment of Romanian. They share the conceptual model in Figure 2.1. However, they are different systems that require different implementations. This subsection briefly presents the language model representations used in these frameworks and the main differences between their implementations[2][3].

2.2.1 Prolog

The rules written in Prolog are based on CFG rules. Listing 1 shows the rules required for an IP construction containing a verb and an optional subject. For each rule in the CFG, every grammatical construction and word order must be explicitly specified in a new Prolog rule.

```
ip --> subj, verb.
ip --> verb, subj.
ip --> verb.
```

Listing 1: Prolog rules required for an IP with subject and verb

In order to add features to the DCG, extra arguments need to be used (Blackburn et al., 2006). They help with encoding features for the words in the lexicon, as well as for the grammar rules. An instance where extra arguments are needed is exemplified in Listing 2 for the subject-verb agreement. The rules on the first two lines manage the agreement between subject and verb in person and number. Then, a DP which agrees in the third person can be used as a subject. The last two rules are part of the lexicon. They encode the number feature of *fata* ('the girl') - singular, and the person and number of *dansează* ('dances') - third person, singular. Therefore, by the given rules, only the grammatical structures are allowed.

[2]The models can be accessed on GitHub at `https://github.com/andreea-ait/Bachelor-Project`.

[3]The Prolog parser file handling the output of the trees and the test features is retrieved from the Computational Grammar course, Assignment 2 (Kreutz, personal communication, 2022).

```
ip --> dp_subj(PERS, NUM), vp(PERS, NUM).
ip --> vp(PERS, NUM), dp_subj(PERS, NUM).
ip --> vp(PERS, NUM).

dp_subj(PERS, NUM) --> dp(PERS, NUM, _, _, _).
vp(PERS,NUM) --> verb(PERS,NUM).

dp(pers3, sg, fem, definite, animate) --> [fata].
verb(pers3, sg) --> [dansează].
```

Listing 2: Using extra arguments for subject-verb agreement

The solution provided for the clitic doubling constructions requires more rules and arguments. Firstly, the DPs used as objects are separated into two categories: `dp_obj` and `dp_obj_cl`. The first one is used without the clitic, and it can match with inanimate, non-specific (indefinite) DPs or definite (with inflectional article) nouns. As shown in (1.3.4), these are the cases in which the DP in object position does not require the accusative marker. The `dp_obj_cl`, on the other hand, refers to DPs that are human and specific, and they are added the accusative marker PE. Furthermore, the person, number and gender features need to be encoded in their arguments, such that the agreement with the clitic can be done. The rules are illustrated in Listing 3.

```
dp_obj --> dp(_, _, _, _, inanimate).
dp_obj --> dp(_, _, _, indefinite, animate).
dp_obj --> dp(_, _, _, definite-marker, animate).

dp_obj_cl(PERS, NUM, GEN) -->
            prt(pe),
            dp(PERS, NUM, GEN, definite, animate).
```

Listing 3: Object DPs

Then, the verb phrase (VP) rules are separated as well into `vp` and `vp_cl`. The `vp` matches with the constructions not requiring clitic doubling. The `vp_cl` covers the cases that allow/require clitic doubling and consists of the clitic (which agrees with the object DP in person, number and gender), the verb and the `dp_obj_cl`.

We follow the approach suggested by Bleam (2000) for the syntax trees that contain clitics, namely, moving the clitic from the object DP to the specifier position of the VP. This provides a correct representation of the word order. However, we offer a simplified notation (i.e. "clit" for the clitic, simplified nodes).

Furthermore, when the subject and object have the same number and person features, if the structure does not allow clitic doubling, ambiguities such as (8) may occur. Hence, even though syntax trees do not include information about the grammatical functions, we provide additional "subj" and "dobj" tree nodes to distinguish between the subject and object DPs. An example of syntax trees in Prolog are shown in Figure 2.2.

(10) a. *Fata îl iubește pe Andrei.*
the girl CL loves PE Andrew
'The girl loves Andrew.'

b. *Fata îl iubește.*
the girl CL loves
'The girl loves him.'

2.2.2 XLE

XLE is a framework that provides an environment adapted for LFG. While in Prolog every rule needs to be manually encoded, XLE supports several linguistic notations. For instance, brackets can be used to

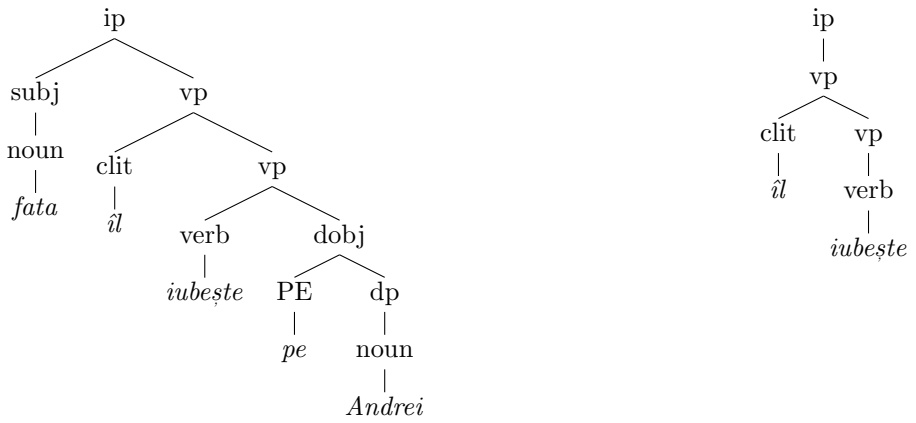

Figure 2.2: Syntax trees for the sentences in (10)

declare optionality and a comma for free arguments order. Several other grammatical notations, such as f-structure metavariables, equality definition, equality constraint, and disjunction are supported as well (Crouch et al., 2011).

Figure 2.3 shows two examples of c-structures that are desired for the XLE parser. We follow the approach given by Barbu and Toivonen (2018) to represent the clitics in the c-structures. In order to achieve these structures, the model requires a set of rules and a lexicon. Each node is defined by a rule that specifies the children nodes, order, optionality of arguments and other constraints. The lexicon is represented by an entry for every word, accompanied by the word type (part of speech) and lexical specifications. An example of a lexical entry for *fata* ('the girl') is shown in Listing 4.

```
fata   N * (^ PRED) = 'GIRL'
           (^ NTYPE) = COUNT
           (^ NUM) = SG
           (^ GEN) = F
           (^ PERS) = 3
           (^ DEF) = +
           { (^ CASE) = NOM
           | (^ CASE) = ACC }
           (^ ANIM) = +.
```

Listing 4: Lexical entry example

An example of an IP rule in XLE is given in Listing 5. The DP is the subject and is marked as optional using parenthesis. Nouns have the same form in nominative and accusative. Hence, the case is constrained by the grammatical function: the subject must bear nominative and the direct object accusative.

```
IP -->  (DP: (^ SUBJ) = !
             (! CASE) =c NOM),
        I': ^=!.
```

Listing 5: XLE rule for IP

Even though several rules are quite straightforward, like the one in the example above, some require multiple disjunctions. For instance, the D′ consists of four rules, each corresponding to a different phrase type: definite nouns, nouns that occur with indefinite articles or demonstratives (short and long form). Another complex rule is the DP (Listing 6). The nouns that require the accusative marker PE were firstly separated. Then, in the DP rule, the accusative case is assigned to them only when the marker is used. The nouns that do not require the marker can bear both cases without any additional constraints.

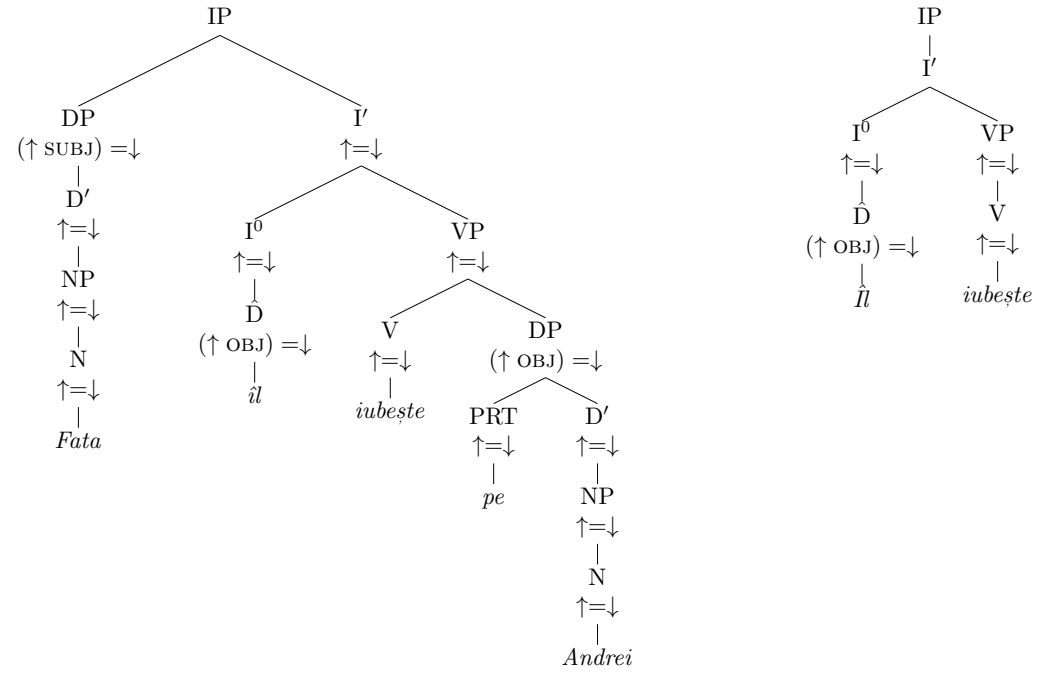

Figure 2.3: C-structures for the sentences in (10)

Moreover, the nouns requiring the accusative marker occur in structures that involve clitic doubling. In order to ensure the correct use of doubling clitics (i.e., only appear in the sentences when required), a new feature was added. Under the I^0 rule, the $\hat{D}$, which holds the clitic, assigns the direct object clitic (DOC) feature, as shown in Listing 7. This feature is further used in the DP rule. The DOC feature is constrained when the PE marker is used. In other words, the doubling clitic is required whenever the direct object in a sentence has the accusative marker PE. The DP rules mentioned above, containing the DOC feature, are illustrated in Listing 6.

```
DP -->  { "nouns that do not require PE:"
          D′: ^ = !;
          |
          "nouns that require PE:"
          D′: (^ CASE) = NOM;
          |
          PRT: (^DOC) =c +;
          D′: (^ CASE) = ACC;
          }.
```

Listing 6: Simplified version of DP rule

```
I0 --> Dhat: (^ OBJ) = !
             (! DOC) = +.
```

Listing 7: XLE rule for Iº

Lastly, it is essential to mention the different functions the clitics can have. When the direct object DP is not present in the sentence, the clitic functions as a pronoun. However, when the object DP occurs, the clitic functions as an agreement marker (Barbu and Toivonen, 2018). This dual analysis ensures a

well-formed f-structure, as shown in Figure 2.4. The lexical entry for the masculine clitic 'îl' is shown in (8). The specifications provided do not only establish the pronouns and the agreement marker functions of the clitic, but they also ensure that the agreement marker occurs only in the allowed structures (i.e. when the object DP is definite/specific, but not for inflected definite nouns).

```
îl      Dhat * (^ PERS) = 3
               (^ NUM) = SG
               (^ GEN) = M
               (^ DEF) = +
             { "pronoun"
               (^ PRED) = 'pro'
               (^ CASE) = ACC
             | "agreement marker"
               (^ CASE) =c ACC
               {(^ ART) ~= +
               |(^ DEM) = +
                (^ FORM) = LONG
               }
             }.
```

Listing 8: Lexical entry for the clitic 'îl'

$$\begin{bmatrix} \text{PRED} & \text{'love}\left\langle \text{SUBJ, OBJ}\right\rangle\text{'} \\ \text{TENSE} & \text{PRES} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'girl'} \\ \text{CASE} & \text{NOM} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'Andrew'} \\ \text{PERS} & 3 \\ \text{NUM} & \text{SG} \\ \text{GEN} & \text{M} \\ \text{CASE} & \text{ACC} \end{bmatrix} \end{bmatrix} \qquad \begin{bmatrix} \text{PRED} & \text{'love}\left\langle \text{SUBJ, OBJ}\right\rangle\text{'} \\ \text{TENSE} & \text{PRES} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'girl'} \\ \text{CASE} & \text{NOM} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'PRO'} \\ \text{PERS} & 3 \\ \text{NUM} & \text{SG} \\ \text{GEN} & \text{M} \\ \text{CASE} & \text{ACC} \end{bmatrix} \end{bmatrix}$$

Figure 2.4: F-structures for the sentences in (10)

2.3 Experiment design

Both models integrate the same amount of syntactic information. Furthermore, the testing suite consists of a set of grammatical and ungrammatical sentences that can be seen in Appendix C. They exhaust all the possible constructions containing direct objects, as well as other linguistic aspects accounted for in this paper. The parsers are expected to identify whether the input is grammatical or not and parse the grammatical sentences.

In terms of time efficiency, the Prolog parser might manage to parse the input sentences and generate the syntax trees faster, as the DCG rules act as patterns and the model works solely based on inferences. On the other hand, XLE is a more complex framework. Even though it might not be as fast, the system can execute more tasks, such as generating the c-structure and the f-structure and displaying the features of each lexical entry of the given input. Hence, the challenging part of comparing the two language models is based on criteria such as usefulness, applicability or implementation complexity.

```
parsed [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26] good sentences,          missed [] sentences
parsed [] ungrammatical sentences, missed [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,
31,32,33,34,35,36,37,38,39,40,41,42,43,44] ungrammaticals
true.
```

Figure 3.1: Prolog parser results

```
26 sentences, 0 errors, 0 mismatches
26 good sentences
22 sentences had exactly one grammatical parse
```

```
44 sentences, 0 errors, 0 mismatches
44 sentences had 0 parses
```

Figure 3.2: XLE parser results: grammatical input (left) and ungrammatical (right)

3 Results

The models can receive as input a testing suite, as well as individual sentences. The output of the models for the sentences in (11) can be found in Appendix C.2. In the Prolog parser, when the input is grammatical, the syntax trees of all possible parses are shown (Figure C.1). If the input is ungrammatical, Prolog returns `false` (Figure C.2). On the other hand, XLE provides the c-structures and the f-structures for grammatical sentences (Figure C.3), as well as invalid trees with highlights on the nodes that generated errors for ungrammatical input (Figure C.4).

(11) a. *Fata îl iubește pe Andrei.*
the girl CL.3SG.M loves PE Andrew
'The girl loves Andrew.'

b. * *Fata iubește pe Andrei.*
the girl loves PE Andrew

Sentences with multiple readings lead to multiple parse trees. For instance, given the sentence in (12), both models generate two trees corresponding to the different readings, as "the boy" can either have the subject or the object grammatical function in the sentence. Sometimes, sentences are correlated to specific readings based on focus-prosody rules in Romanian (Gobbel, 2003). However, the information structure is not part of this project. Hence, every grammatical structure within the fragment boundaries is allowed by the parsers, regardless of its meaning or prosody.

(12) *Iubește băiatul.*
loves the boy
'He/she loves the boy.'
'The boy loves.'

The parsers were tested on the sets of sentences presented in Appendix C.1. Both models managed to parse the grammatical sentences and identify the ungrammatical ones. Figure 3.1 shows the output of the Prolog parser after testing. Only the grammatical sentences were parsed, while the others were missed. The XLE parser displays the same result, as seen in Figure 3.2: the grammatical sentences are labelled as "good input", and the ungrammatical ones have no parses. Hence, the parsers have a score of 100% accuracy on the presented sentences.

For individual sentences, Prolog shows the output immediately (0.000 execution time), while the longest execution time of XLE is 0.006 seconds. For the testing suite, Prolog manages to parse the grammatical sentences and identify the ungrammatical ones in 0.004 seconds. On the other hand, XLE executes the same task in ≈0.1 seconds.

4 Discussion

4.1 Comparison

The results provided in Section 3 show that both Prolog and XLE parsers scored 100% accuracy on the input presented in Appendix C.1. The main difference occurs in the output of the models. Even though the same amount of syntactic information and lexical features are encoded in the models, Prolog only displays the syntax tree. At the same time, XLE captures much richer information in the f-structure, offering more insights from a linguistic perspective. Furthermore, XLE shows the invalid parse trees of the ungrammatical sentences and highlights the invalid feature structures that caused the error. Additionally, if a word in the input is not recognised, meaning it either contains a spelling error or is not in the lexicon, the word is flagged in the shell. These features can be valuable, especially in the testing/debugging phase.

In terms of time performance, on individual parses, the difference in execution time is insignificant. However, Prolog is able to parse both sets of sentences in C.1 in 0.004 seconds, time that XLE requires for only one parse. The difference is unnoticeable on a testing suite that consists of less than 100 sentences, as the output is shown in less than 1 second. However, for a much larger set, or an increase in the number of rules, the execution time might raise issues in the XLE framework.

Another important aspect to consider in the comparison between Prolog and XLE is the programmer's perspective. First of all, Prolog is easily accessible. Several Prolog environments and interpreters can be downloaded online (e.g., SWI-Prolog - used in the current project, Tau Prolog, GNU Prolog). XLE Project, on the other hand, can be accessed after a request is approved by one of the project's moderators.

Secondly, grammars can be written in Prolog using DCGs and additional arguments. There are no other constraints on the implementation, hence the programmer has more freedom when it comes to encoding grammatical rules. While XLE provides several notations following the LFG formalism, the programmer must adhere to the expressive syntax rules. For example, the noun forms "girl" and "girls" can be introduced and accessed in Prolog in multiple ways, as shown in Listings 9 and 10. However, XLE requires two different entries, as shown in Listing 11, even though the only difference is the number feature.

```
np(NOUN, NUM) --> noun(NOUN, NUM).

noun(girl, sg).
noun(girls, pl).
```

Listing 9: Introducing and accessing "girl" and "girls" in Prolog, example 1

```
np(NOUN, sg) --> noun(NOUN, _).
np(NOUN, pl) --> noun(_, NOUN).

noun(girl, girls).
```

Listing 10: Introducing and accessing "girl" and "girls" in Prolog, example 2

```
girl  N * (^ PRED) = 'GIRL'
          (^ NUM) = SG.

girls N * (^ PRED) = 'GIRL'
          (^ NUM) = PL.
```

Listing 11: Lexical entries for "girl" and "girls" in XLE

Romanian nouns, however, display more features compared to the English forms of "girl" presented above. Regardless of the representation method, both parsers encode the same features, namely number, gender, case, noun type, definiteness and animacy.

Besides the engineering and programming views, the underlying linguistic theories of the parsers constitute a substantial point in the current discussion. Even though the Prolog parser relies on a CFG, the representation of the grammar does not require following any linguistic theory. For instance, the parse trees generated do not follow Chomsky's theory of movement. The clitic's movement to the specifier

position of the VP offers a depiction of the surface word order of the sentences, not an ideal representation of the phenomenon under the X-bar theory. Moreover, the presence of the dp_subj and dp_obj categories in the output trees serves as a disambiguation solution for structures as shown in example (8).

Indeed, the lack of generality and the inability to capture cross-linguistic principles of the very specific transformational approaches led to the emergence of linguistic theories such as LFG (Dalrymple, 2001). By these means, the XLE framework provides the means to represent complex linguistic aspects under the non-transformational LFG theory. The c-structures displayed in XLE adhere to the basic principles of X-bar theory, which are a theoretical-based representation of the language, contrasting the parse trees generated in Prolog.

4.2 Conclusion

Prolog and XLE are frameworks that facilitate language encoding. We have shown that complex linguistic phenomena such as clitic doubling can be difficult to implement and require rigorous rules. Previous studies have shown several ways to analyse these phenomena under different linguistic theories. However, practical solutions that allow the computation of these linguistic aspects are needed. Nevertheless, the provided models display impressive results on the small fragment of the Romanian language. We conclude that there is no significant difference between the two models in terms of time and accuracy. However, choosing one of the two frameworks requires a trade-off between the accessibility and implementation flexibility of Prolog and the LFG built-in design and rich structure information offered by XLE. Furthermore, the relevance of the theoretical background for the development of a linguistic project must be considered, as the programming freedom offered by Prolog might compromise the accuracy of linguistic theories, while XLE ensures an LFG conforming environment.

This project serves as an in-depth presentation of the process of traditionally parsing complex linguistic phenomena under two different frameworks. However, an underlying intent is to emphasize the difficulties encountered in a task involving natural language. While several state-of-the-art computational linguistics applications employ advanced AI techniques (i.e., the Neural Machine Translation approach used in Google Translate (Wu et al., 2016)), the systems present limitations due to linguistic complexities (Coheur, 2020). Computers do not understand grammar, hence language processing is undoubtedly difficult. Hybrid systems consisting of machine learning and traditional parsing techniques could represent a solution that accounts for large amounts of available data while ensuring more accurate processing of natural language.

4.3 Future research

The implementation of the models in this project is limited to the grammatical structure of specific constructions in Romanian. However, several linguistic studies focus not only on the syntactic structure, but also on its relation to information structure (Dalrymple, 2001). Word order and information structure can be represented in LFG, for instance, using the TOPIC and FOCUS discourse functions, as proposed by Butt and Holloway (1996).

Given the various possible word orders allowed in Romanian sentences, the discourse functions could provide a solution to account for the differences between the possible structures. Furthermore, they could help manage the special cases of clitic doubling, such as the exception shown in (9). This is an example of a sentence involving the topicalization of the object, phenomenon that could be implemented in XLE using discourse functions. Following the approach given by Dalrymple (2001) and Jaeger and Gerassimova (2002) for the Bulgarian clitics, where the specifier of the IP is the FOCUS position, the sentence in (9b) could be generated by the parser, and its c-structure would correspond to the one shown in Figure 4.1.

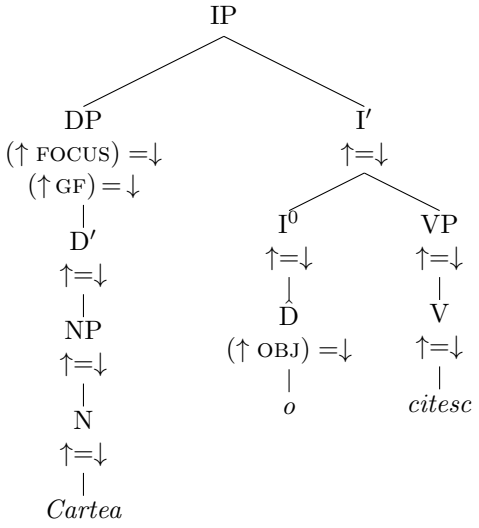

Figure 4.1: C-structure for the sentence in (9b) following the Bulgarian clitic approach (Dalrymple, 2001; Jaeger and Gerassimova, 2002)

Acknowledgements

I would like to thank my Bachelor Project supervisor, Dr. Stephen Jones, for the guidance and assistance offered throughout the entire process of completing the project. Accomplishing this paper was possible due to his involvement, patience and constant encouragement. Thank you for supporting me throughout this process!

References

Alexandru, M. and Silvina, M. (2020). *The Acquisition of Differential Object Marking*. Number volume 26 in Trends in Language Acquisition Research. John Benjamins Publishing Company. Retrived from `http://search.ebscohost.com.proxy-ub.rug.nl/login.aspx?direct=true&db=nlebk&AN=2494123&site=ehost-live&scope=site`.

Anagnostopoulou, E. (2006). Clitic doubling. *The Blackwell companion to syntax*, 1:519–581.

Babyonyshev, M. and Marin, S. (2005). The acquisition of object clitic constructions in Romanian. In Rubin, E. J. and Gess, R. S., editors, *Theoretical and Experimental Approaches to Romance Linguistics: Selected Papers from the 34th Linguistic Symposium on Romance Languages (LSRL), Salt Lake City, March 2004*, number v. 272 in Amsterdam Studies in the Theory and History of Linguistic Science, page 21–40. John Benjamins Publishing Co.

Barbu, M.-R. and Toivonen, I. (2018). Romanian Object Clitics: Grammaticalization, agreement and lexical splits. In Butt, M. and King, T. H., editors, *Proceedings of the LFG'18 Conference, University of Vienna*, pages 67–87, Stanford, CA. CSLI Publications.

Blackburn, P., Bos, J., and Striegnitz, K. (2006). *Learn Prolog Now!* College Publications.

Bleam, T. M. (2000). Leísta Spanish and the syntax of clitic doubling. In *IRCS Technical Reports Series*. Retrived from `https://repository.upenn.edu/ircs_reports/33`.

Butt, M. and Holloway, T. (1996). Structural Topic and Focus without Movement. In Butt, M. and Holloway, T., editors, *Online Proceedings of the First LFG Conference, Rank Xerox, Grenoble, August 26-28, 1996*. Retrieved from `http://web.stanford.edu/group/cslipublications/cslipublications/LFG/1/lfg96butt.pdf`.

Butt, M., King, T. H., Niño, M.-E., and Segond, F. (1999). *A Grammar Writer's Cookbook*. CSLI Publications, Leland Stanford Junior University.

Börjars, K., Nordlinger, R., and Sadler, L. (2019). *Lexical-Functional Grammar: An introduction*. Cambridge University Press.

Coheur, L. (2020). From Eliza to Siri and Beyond. *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, page 29–41. Retrived from `https://link.springer.com/chapter/10.1007/978-3-030-50146-4_3`.

Crouch, D., Dalrymple, M., Kaplan, R., King, T., Maxwell, J., and Newman, P. (2011). XLE Documentation. Retrived from `https://ling.sprachwiss.uni-konstanz.de/pages/xle/doc/xle.html`.

Dalrymple, M. (2001). *Lexical Functional Grammar*. BRILL.

Dindelegan, G. P. (2013). *The Grammar of Romanian*. Oxford University Press.

Gobbel, E. (2003). On the relation between focus, prosody and word order in Romanian. *Romance languages and linguistic theory 2001: Selected papers from 'Going Romance', Amsterdam, 6-8 December 2001*, pages 75–92.

Grishman, R. (1986). *Computational Linguistics: An Introduction*. Studies in Natural Language Processing. Cambridge University Press.

Jaeger, T. F. and Gerassimova, V. (2002). Bulgarian Word Order and the Role ff the Direct Object Clitic in LFG. In Butt, M. and King, T. H., editors, *Proceedings of the LFG02 Conference, National Technical University of Athens, Athens*. CSLI Publications.

Mayer, E. (2003). *Clitic Doubling in Limeño. A Case Study in LFG*. PhD thesis.

Sadler, L. (1997). Clitics and the Structure-Function Mapping. In Butt, M. and King, T. H., editors, *Proceedings of the LFG'97 Conference, University of California, San Diego*. CSLI Publications.

Schubert, L. (2020). Computational Linguistics. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Spring 2020 edition.

Steedman, M. (2008). Last Words: On Becoming a Discipline. *Computational Linguistics*, 34(1):137–144. Retrived from `https://aclanthology.org/J08-1008`.

Wu, Y., Schuster, M., Chen, Z., Le, Q., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, , Gouws, S., Kato, Y., Kudo, T., Kazawa, H., and Stevens, K. (2016). Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. Retrieved from `https://arxiv.org/pdf/1609.08144.pdf`.

A Appendix

A.1 Nouns and determiners

		MASCULINE		FEMININE		NEUTER	
		SG	PL	SG	PL	SG	PL
Articleless nouns	NOM≡ACC	elev-Ø	elev-**i**	fat-**ă**	fet-**e**	măr-Ø	mer-**e**
	GEN≡DAT	elev-Ø	elev-**i**	fet-**e**	fet-**e**	măr-Ø	mer-**e**
		pupil-SG		girl-SG		apple-SG	
Nouns bearing the definite article	NOM≡ACC	elev-**u-l**	elev-**i-i**	fat-**a**	fet-**e-le**	măr-**u-l**	mer-**e-le**
	GEN≡DAT	elev-**u-lui**	elev-**i-lor**	fet-**e-i**	fet-**e-lor**	măr-**u-lui**	mer-**e-lor**
		pupil-SG-DEF		girl-SG-DEF		apple-SG-DEF	

Table A.1: Gender marking on Romanian nouns. Retrieved from (Dindelegan, 2013)

		MASCULINE		FEMININE
SG	NOM≡ACC	un		o
	GEN≡DAT	unui		unei
PL	NOM≡ACC		niște	
	GEN≡DAT		unor	

Table A.2: The forms of the indefinite article. Retrieved from (Dindelegan, 2013)

	MASCULINE				FEMININE			
	SINGULAR		PLURAL		SINGULAR		PLURAL	
CASE	SHORT	LONG	SHORT	LONG	SHORT	LONG	SHORT	LONG
NOM≡ACC	acest	acest**a**	acești	acești**a**	această	aceast**a**	aceste	aceste**a**
GEN≡DAT	acestui	acestui**a**	acestor	acestor**a**	acestei	acestei**a**	acestor	acestor**a**

Table A.3: The forms of *acest/a* (this). Retrieved from (Dindelegan, 2013)

A.2 Clitic doubling

PERS	NUMBER	
	SG	PL
1	mă/m	ne
2	te	vă/v
3M	îl/l	îi/i
3F	o	le

Table A.4: The direct object clitic forms. Retrieved from (Barbu and Toivonen, 2018)

B Appendices

B.1 Prolog

```
s -> np vp
np -> det n
vp -> v np
vp -> v
det -> a
det -> the
n -> woman
n -> man
v -> shoots
```

Figure B.1: Simple context-free grammar

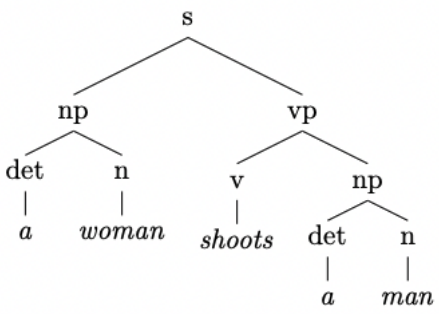

Figure B.2: Parse tree example

B.2 XLE

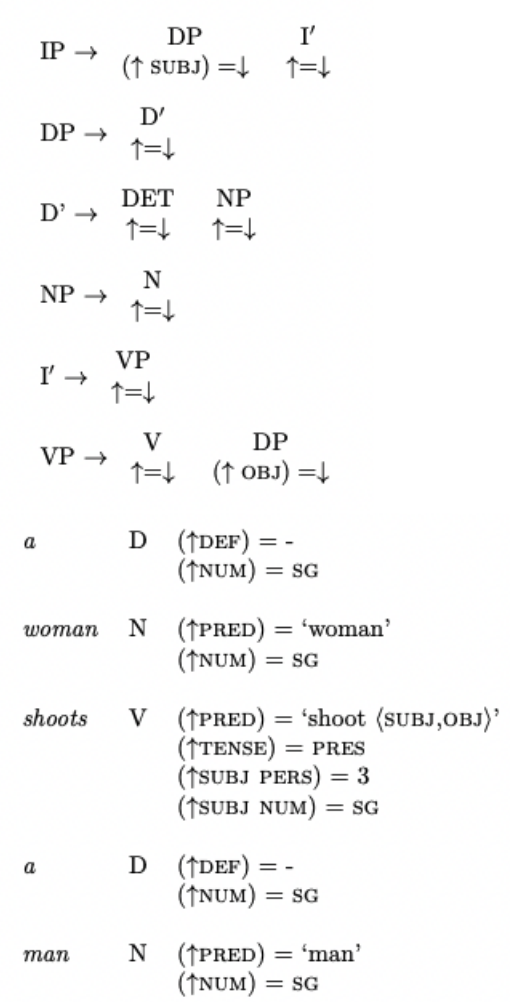

Figure B.3: Phrase structure rules (above) and lexical specifications (below)

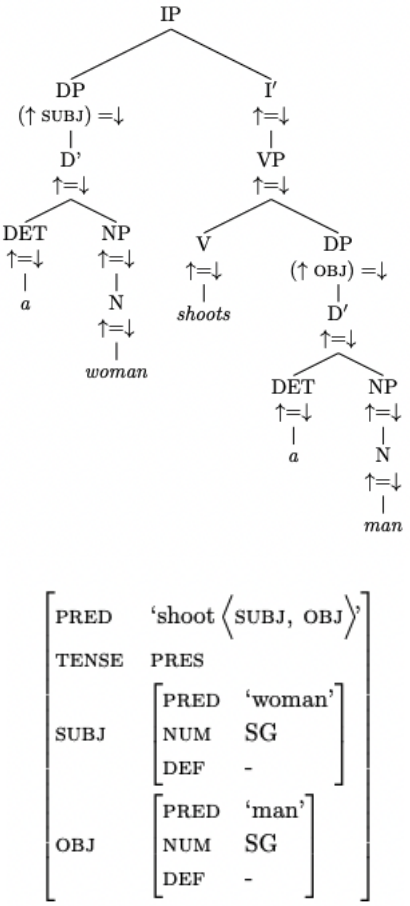

Figure B.4: C-structure (above) and f-structure (below)

C Appendices

C.1 Input

Grammatical test sentences:

o fată mănâncă
fata mănâncă un măr
fata mănâncă acest măr
fata mănâncă mărul acesta
fata mănâncă mărul
fata mănâncă niște mere
Andrei dansează
această fată dansează
dansează fata
Andrei mănâncă un măr
Andrei îl mănâncă
Andrei o iubește pe Ana
fata îl iubește pe Andrei
fata iubește băiatul
fata iubește un băiat
fata îl iubește pe acest băiat
fata îl iubește pe băiatul acesta
Ana o iubește pe fata aceasta
fata iubește fata
iubește băiatul

Ungrammatical test sentences:

Andrei acesta dansează
un fată dansează
un băiatul dansează
acest băiatul dansează
acest un băiat dansează
această băiat dansează
această băiatul dansează
un băiat acesta dansează
băiatul aceasta dansează
fata mănâncă această măr
fata mănâncă acești mere
fata mănâncă pe mărul acesta
fata iubește pe băiatul
fata iubește pe un băiat
fetele mănânc merele
fată mănâncă
fata dansez
Andrei îl mănâncă un măr
Andrei îl mănâncă pe măr
fata îl iubește Andrei
fata îl iubește pe un băiat
fata îl iubește un băiat
fata îl iubește pe Ana
fata îl dansează
fata îl iubește băiatul
fata îl iubește pe băiatul
fata iubește pe un băiat
fata iubește pe băiatul
Andrei mănâncă pe măr
fata iubește pe un băiat
fata îl iubește băiatul
pe această fată dansează
fata iubește pe Andrei
fata iubește pe acest băiat
mănâncă pe un măr
mănâncă pe acest măr
îl mănâncă pe acest măr
fata îl mănâncă acest măr
iubește pe o fată
o iubește pe o fată
iubește pe această fată
iubește pe Andrei
iubește pe băiatul
îl iubește pe băiatul

C.2 Output examples

Prolog output:

```
?- parse([fata,îl,iubește,pe,acest,băiat]).
  ip ---- subj -- dp ---- noun -- fata
     ---- vp ---- clit -- îl
             ---- vp ---- verb -- iubește
                     ---- dobj -- PE ---- pe
                                 -- dp ---- det --- acest
                                        ---- noun -- băiat

true .
```

Figure C.1: Prolog output on grammatical input

```
?- parse([fata,iubește,pe,acest,băiat]).
false.
```

Figure C.2: Prolog output on ungrammatical input

XLE output:

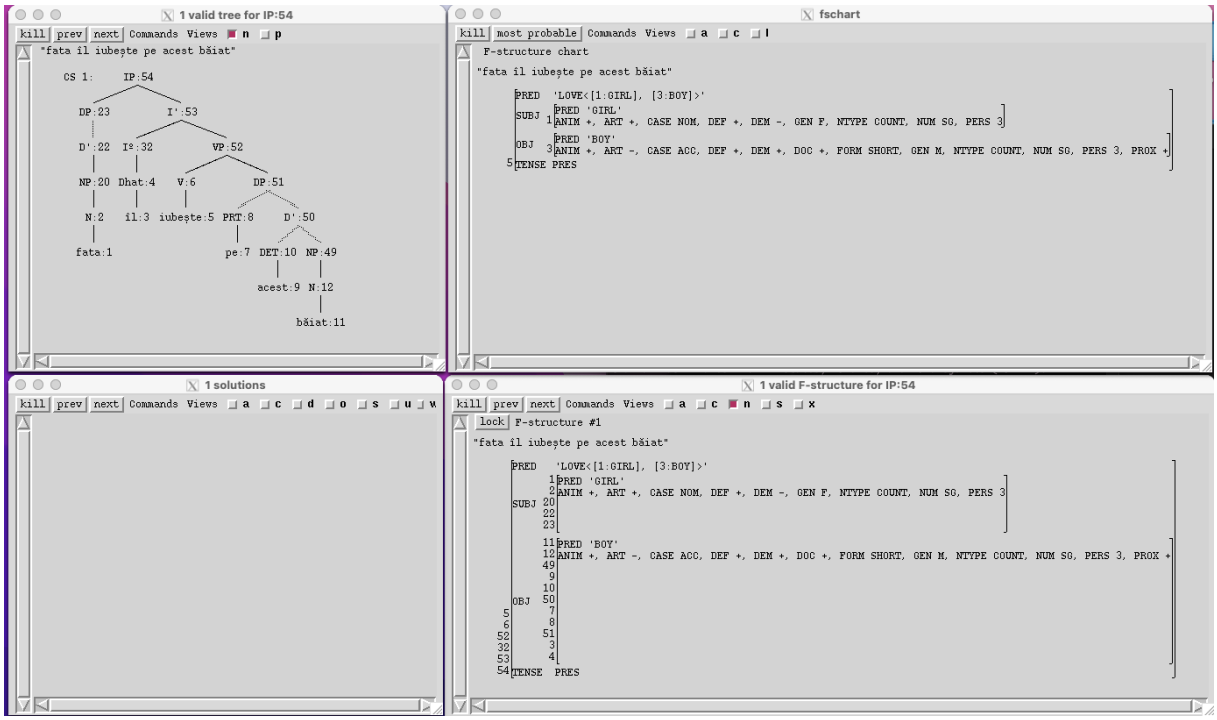

Figure C.3: XLE output on grammatical input

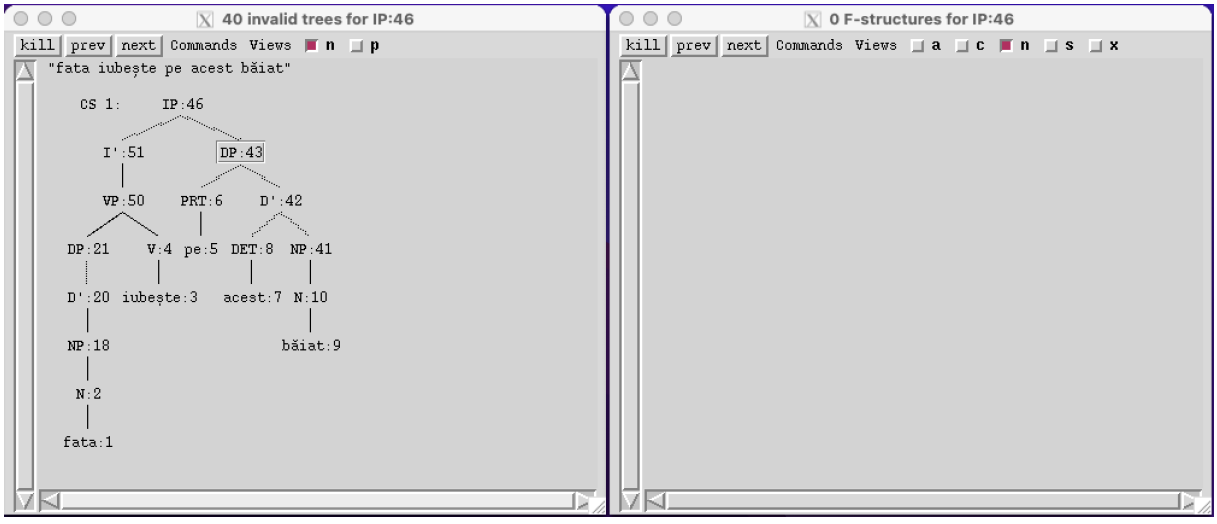

Figure C.4: XLE output on ungrammatical input