

A LINK GRAMMAR FOR TURKISH

A THESIS
SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULLFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Özlem İstek
August, 2006

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. İlyas Çiçekli (Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. H. Altay Güvenir

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Ferda Nur Alpaslan

Approved for the Institute of Engineering and Sciences:

Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

A LINK GRAMMAR FOR TURKISH

Özlem İstek
M.S. in Computer Engineering
Supervisor: Asst. Prof. Dr. İlyas Çiçekli

August, 2006

Syntactic parsing, or syntactic analysis, is the process of analyzing an input sequence in order to determine its grammatical structure, i.e. the formal relationships between the words of a sentence, with respect to a given grammar. In this thesis, we developed the grammar of Turkish language in the link grammar formalism. In the grammar, we used the output of a fully described morphological analyzer, which is very important for agglutinative languages like Turkish. The grammar that we developed is lexical such that we used the lexemes of only some function words and for the rest of the word classes we used the morphological feature structures. In addition, we preserved the some of the syntactic roles of the intermediate derived forms of words in our system.

Keywords: Natural Language Processing, Turkish grammar, Turkish syntax, Parsing, Link Grammar.

ÖZET

TÜRKÇE İÇİN BİR BAĞ GRAMERİ

Özlem İstek
Bilgisayar Mühendisliği Bölümü, Yüksek Lisans
Tez Yöneticisi: Yar. Doç. Prof. Dr. İlyas Çiçekli
Ağustos, 2006

Sözdizimsel çözümleme veya ayrıştırma, bir tümcenin dilbilgisel yapısını yani kelimeleri arasındaki ilişkiyi ortaya çıkarmak amacıyla verilen bir gramere göre inceleme işlemidir. Bu çalışmada, Türkçe için bir bağ grameri geliştirilmiştir. Sistemimizde Türkçe gibi çekimli ve bitişken biçimbirimlere sahip diller için çok önemli olan, tam kapsamlı, iki aşamalı bir biçimbirimsel tanımlayıcının sonuçları kullanılmıştır. Geliştirdiğimiz gramer sözcükseldir ancak, bazı işlevsel kelimeler oldukları gibi kullanılırken, diğer kelime türleri için kelimelerin kendilerinin yerine biçimbirimsel özellikleri kullanılmıştır. Ayrıca sistemimizde kelimelerin ara türeme formlarının sözdizimsel rollerinin bazıları muhafaza edilmiştir.

Anahtar Kelimeler: Doğal Dil İşleme, Türkçe Dilbilgisi, Türkçe sözdizimi, Sözdizimsel Çözümleme, Bağ Grameri.

Acknowledgement

I would like to express my deep gratitude to my supervisor Asst. Prof. Dr. İlyas Çiçekli for his invaluable guidance, encouragement, and suggestions throughout the development of this thesis.

I would also like to thank Prof. Dr. H. Altay Güvenir and Assoc. Prof. Ferda Nur Alpaslan for reading and commenting on this thesis.

I would like to thank my friends Abdullah Fişne and Serdar Severcan for their help. I am also grateful to my friend Arif Yılmaz for his invaluable help, moral support, encouragement and suggestions.

I am grateful to my family for their infinite moral support and help throughout my life.

To my mother, Fatma İSTEK

Contents

1 Introduction.....	1
1.1 Linguistic Background.....	3
1.2 Thesis Outline.....	7
2 Link Grammar	8
2.1 Introduction.....	8
2.2 Main Rules of the Grammar.....	9
2.3 Language and Notion of Link Grammars	10
2.3.1 Rules for Writing Connector Blocks or Linking Requirements.....	10
2.3.2 The Concept of Disjuncts.....	12
2.4 General Features of the Link Parser	13
2.5 Special Features of the Dictionary.....	14
2.6 Coordinating Conjunctions	17
2.6.1 Handling Conjunctions	18
2.6.2 Some Problematic Conjunctional Structures.....	20
2.7 Post-Processing	21
2.7.1 Introduction	21
2.7.2 Structures of Domains.....	21
2.7.3 Rules in Post Processing	22
3 Turkish Morphology and Syntax.....	24
3.1 Distinctive Features of Turkish	24
3.2 Turkish Morphotactics.....	28
3.2.1 Inflectional Morphotactics	29
3.2.2 Derivational Morphotactics.....	33
3.2.3 Question Morpheme.....	37
3.3 Constituent Order in Turkish.....	38

3.4 Classification of Turkish Sentences	40
3.4.1 Classification by Structure	41
3.4.2 Classification by Predicate Type	42
3.4.3 Classification by Predicate Place.....	44
3.4.4 Classification by Meaning.....	44
3.5 Substantival Sentences.....	45
4 Design.....	47
4.1 Morphological Analyzer	47
4.1.1 Turkish Morphological Analyzer	47
4.1.2 Improvements and Modifications to Turkish Morphological Analyzer	49
4.2 System Architecture.....	52
5 Turkish Link Grammar	61
5.1 Scope of Turkish Link Grammar.....	63
5.2 Linking Requirements Related to All Words.....	63
5.3 Compound Sentences, Nominal Sentences, and the Wall	67
5.4 Linking Requirements of Word Classes	74
5.4.1 Adverbs	74
5.4.2 Postpositions.....	76
5.4.3 Adjectives and Numbers	78
5.4.4 Pronouns.....	81
5.4.5 Nouns	85
5.4.6 Verbs	90
5.4.7 Conjunctions.....	93
6 Performance Evaluation	95
7 Conclusion	101
BIBLIOGRAPHY	103
A Turkish Morphological Features	106

B Summary of Link Types	108
C Input Document and Statistical Results.....	112
D Example Output from Our Test Run.....	113

List of Figures

Figure 1 METU-Sabancı Turkish Treebank	3
Figure 2 Typical Order of Constituents in Turkish.....	39
Figure 3 Architecture of a Two Level Morphological Analyzer	48
Figure 4 System Architecture	53
Figure 5 Special Preprocessing for Derived Words.....	58
Figure 6 Example to Preprocessing for Derived Words.....	58
Figure 7 Linking Requirements of Intermediate Forms of a Word, W_x	64
Figure 8 Change of Linking Requirements of an IDF According to Its Place ...	65
Figure 9 Macro for the Derivation Boundary and Question Morpheme	67
Figure 10 Linking Requirements of the LEFT-WALL	69
Figure 11 Rules for Adjectives	71
Figure 12 Suffixless Adjective to Verb Derivation, an Example Illustrative Sentence Structure	72
Figure 13 Linking Requirements of Adverbs	75
Figure 14 Linking Requirements of Postpositions.....	77
Figure 15 Linking Requirements of Adjectives.....	78
Figure 16 Linking Requirements of Numbers	80
Figure 17 Linking Requirements of Nominative Pronouns.....	81
Figure 18 Linking Requirements of Genitive and Accusative Pronouns	83
Figure 19 Linking Requirements of Locative/Ablative/Dative/Instrumental Pronouns	85
Figure 20 Left Linking Requirements Common to All Nouns.....	88
Figure 21 Right Linking Requirements of Nouns.....	89

List of Tables

Table 1 Effects of Causation to Verbs.....	36
Table 2 Verb Subcategorization Information	55
Table 3 Subscript Set for S (Subject) Connector	82
Table 4 Statistical Results of the Test Run.....	97

List of Abbreviations

SOV	Subject object verb
POS	Part of speech tag
LG	Link Grammar
IDF	Intermediate Derived Form
LG	Link Grammar
TLG	Turkish Link Grammar
LR	Linking Requirements
DLR	Derivational Linking Requirements
LLR	Left Linking Requirements
RLR	Right Linking Requirements
NDLR	Non-Derivational Linking Requirements
NDLLR	Non-Derivational Left Linking Requirements
NDRLR	Non-Derivational Right Linking Requirements
DC	Dependent Clause
IC	Independent Clause
NLP	Natural Language Processing

Chapter 1

Introduction

Syntax is the formal relationships between words of a sentence. It deals with word order, and how the words depend on other words in a sentence. Hence, one can write rules for the permissible word order combinations for any natural language and this set of rules is named as grammar. Syntactic parsing, or syntactic analysis, is the process of analyzing an input sequence in order to determine its grammatical structure with respect to a given grammar. There are different classes of theories for the natural language syntactic parsing problem and for creating the related grammars. One of these classes of formalisms is categorical grammar motivated by the principle of compositionality¹. According to this formalism, syntactic constituents combine as functions or in a function-argument relationship. In addition to categorical grammars, there are two other classes of grammars, and these are phrase structure grammars, and dependency grammars. Phrase structure grammars are the well-known Type-2, i.e. context free, grammars of Chomsky hierarchy. Phrase grammar constructs constituents in a three-like hierarchy, head-driven phrase structure grammars (HPSG), and lexical functional grammars are some popular types of phrase structure grammars. On the other hand, dependency grammars build simple relations between pairs of words. Since dependency grammars are not defined by a specific word order, they are well suited to languages with free word order, such as Czech and Turkish. Link grammar, which is a theory of syntax by Davy Temperley and Daniel Sleator [1] , is similar to dependency grammar, but link

¹ **Principle of Compositionality** is the principle that the meaning of a complex expression is determined by the meanings of its constituent expressions and the rules used to combine them.

grammar includes directionality in the relations between words, as well as lacking a head-dependent relationship.

In this thesis, we study Turkish syntax from a computational perspective. Our aim is to develop a link grammar for Turkish as complete as possible. The reason for us to choose to study Turkish syntax computationally is syntactic analysis underlies most of the natural language applications. Hence, to accelerate new researches on Turkish as a lesser studied language, syntactic analysis is a very important step. One of the reasons for us to choose the link grammar formalism to develop our grammar is that it is based on the dependency formalism which is known to be more suitable for free order languages like Turkish. In addition, link grammar is lexical and this property makes it an easy development environment for a large, full coverage grammar.

In addition to our work, there also some other researches on the computational analysis of Turkish syntax. One of these is a lexical functional grammar of Turkish by Güngördü in 1993 [8]. Demir [18] also developed an ATN grammar for Turkish in 1993. Another grammar is based on HPSG formalism and developed by Sehitoglu in 1996 [7]. Hoffman in 1995 [19], Çakıcı in 2005[21], and Bozşahin in 1995 [20] worked on categorial grammars for Turkish.

In addition to these categorial and context free works, Turkish syntax is studied from the dependency parsing perspective. Oflazer presents a dependency parsing scheme using an extended finite state approach. The parser augments input representation with “channels” so that links representing syntactic dependency relations among words can be accommodated, and iterates on the input a number of times to arrive at a fixed point [13]. During the iterations crossing links, items that could not be linked to rest of the sentence, etc, are filtered by finite state filters. They used this parser for building a Turkish

treebank [22], namely METU-Sabancı Turkish Treebank. The explanatory paragraph, in Figure 1 is directly taken from the web site of the treebank .

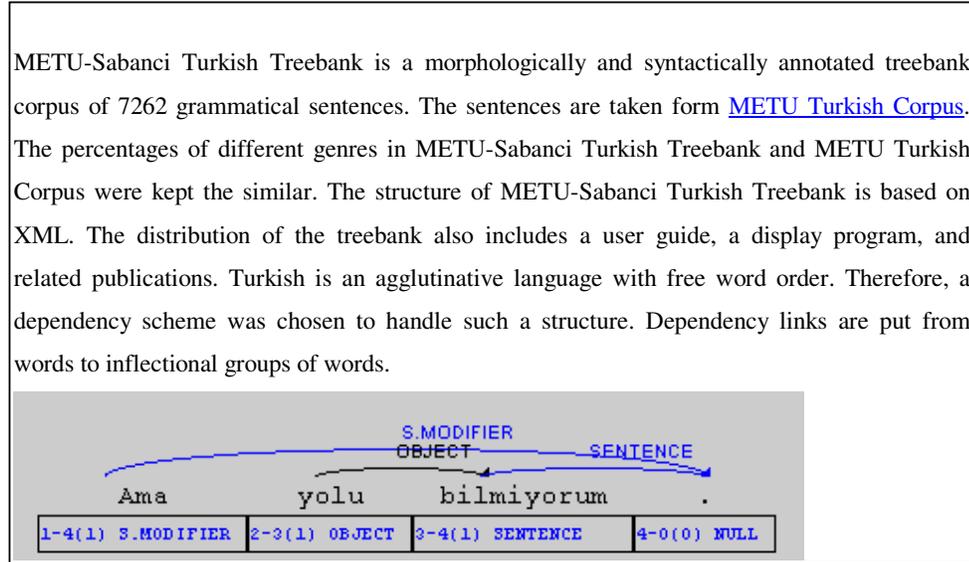


Figure 1 METU-Sabancı Turkish Treebank

The Turkish Dependency Treebank explained above is used for training and testing a statistical dependency parser for Turkish by Oflazer and Eryiğit [12]. In their work, they explored different representational units for the statistical models of parsing.

1.1 Linguistic Background

In this section, linguistic background for necessary for the rest of the thesis together with some terms will be given in detail.

The minimal meaning-bearing unit in a language is defined as a *morpheme*. For example, the word “books” consists of two morphemes, “book”, and “s”. Morphemes can be further categorized into two classes, *stems*, and *affixes*. *Stems* supply the main meaning of the words while *affixes* supply the additional meanings. Hence, in the previous example, the morpheme “book” is the stem of

the word “books”, and the morpheme “s” is an affix. The study of the way that words are built up from morphemes, stem and affixes, is defined as the *morphology*. New words can be formed from stems by *inflection* or *derivation*. The difference between *inflection* and *derivation* is that, the resulting word of inflection has the same class as the original stem, whereas the resulting word has a different class after derivation. For example, “books” is formed by inflection from the stem “book” and the suffix “-s”. In addition, the word “books” and the stem “book” have the same class (noun). On the other hand, the noun “preparation” is derived from the verb “prepare”. *Part of Speech (POS) Tag* of a word represents its class. *Noun* is the POS tag of the word “book”. Therefore, each stem has a POS tag and derivational affixes can change the POS tag of the stems that they are appended. *Orthographic rules* are the *spelling rules* or *phonetic rules* and they are used to model the changes that occur in a word, usually when two morphemes combine. For example “y->ie” spelling rule changes “baby+-s” to “babies” instead of “babys” [16].

Rules specifying the ordering of the morphemes are defined by the term *morphotactics*. For example, in Turkish the plural suffix “-ler” may follow nouns. *Morphological features* are the additional information about the stem and affixes. “Book + Noun+ Plural” contains the morphological features of the word “Books”. *Morphological features* of words are produced through *morphological analysis*. Hence, the terms *morphological features*, *morphological analysis*, and *morphological parse* of a word can be used interchangeably. Any morphological processor needs morphotactic rules, orthographic rules, and *lexicons* of its language. A *lexicon* is the list of stems with their POS tags.

A *sentence* is a group of words that contains *subjects* and *predicates* and expresses assertions, questions, commands, wishes, or exclamations as complete thoughts. Each sentence is thought to have a subject, an object, and a verb, and one of these can be implied. In a sentence with just one complete thought, the

predicate of the sentence is the group of words that collectively modify the subject. In the following examples, the predicate is underlined.

- I. Ali cooks.
- II. Özlem is in the cinema.
- III. He is attractive.

Subject is defined as the origin of the action or undergoer of the state shown by the predicate in a sentence.

Valence (valency) is the number of arguments that a verb takes. Verbs can be categorized according to their valence. *Intransitive* verbs, verbs with valence=1, takes only subject. *Transitive* verbs have a valence of two and they can take a direct object in addition to subject. *Ditransitive* verbs have a valency of three and they can take a subject, a direct object, and an indirect object. Causative forms of verbs can be obtained through causation operation. Causation operation increases the valences of the verbs. After causation, an intransitive verb becomes a transitive one, a transitive verb becomes a ditransitive verb. Each language has its own way of handling causation. Inflectional or derivational suffixes, idiomatic expressions, auxiliary verbs and, lexical causative forms are the tools to causate verbs in the languages.

Sentences can consist of *independent clauses*, i.e. *IC*, and *dependent clauses*, i.e. *DC*. *Independent clauses* express a complete thought and contain a subject and a predicate. On the other hand, since a *DC* (or *subordinate clause*) does not express a complete thought, it cannot stand alone as a sentence. Hence, a *DC* is usually attached to an *IC*. Although a *DC* contains a subject and a predicate, it sounds incomplete when standing alone. In general, a *DC* is started with a *dependent word*. There are two types of dependent words. The first kind of dependent words are *subordinating conjunctions*. Subordinating conjunctions are used to start *DCs* of type adverbial clauses and they act like adverbs.

- I. He left **when** *he saw me* (subordinating conjunction is in bold and the adverbial clause is in italic)

The second kind of dependent words are *relative pronouns*. They are used to start DCs of either adjectival clauses¹ or noun clauses².

- I. The dog **that** *chased me was black*. (The DC “that chased me” modifies “The dog”)
- II. I do not know **how** *he is so crude*. (The DC “he is so crude” functions as a noun)

Sometimes, different parts of the sentences or phrases cross reference to each other. This situation is named as *agreement* in linguistics. If there is *agreement* between the two parts of a sentence (or phrase), changes of form in the first word depends on the changes of form in on the other. For example, in Latin and Turkish, verbs agree in person and number with their subjects. Agreed parts of the sentences are in bold case in the following examples.

- I. Porto “I carry” in Latin
- II. **Portas** “you carry” in Latin
- III. **Ben** **geldim** “I came” in Turkish
I came
- IV. **Sen** **geldin** “You came” in Turkish
You came

¹ They behave like adjectives.

² They behave like nouns.

In some languages, *agreement* allows the constituents to change their default place in sentences without relying on the case endings, i.e. free constituent order. On the other hand, it results in redundancy allowing some pronouns to drop frequently, a situation known as *pro-dropping*. Chomsky[17] also suggests that there is a one-way correlation between inflectional agreement and empty pronouns on the one hand and between no agreement and overt pronouns, on the other hand. More formally, a *pro-drop language* is a language in which pronouns can be omitted since they can be inferred from the context. If a language allows only the subject pronouns to be omitted, it is named to be *partially pro-drop*, e.g. French, and Italian. On the other hand, languages those allow other constituents to drop, like object, in addition to the subject are called *pro-drop*, e.g. Turkish, and Japanese. English is considered a *non-pro-drop* language.

1.2 Thesis Outline

The outline of the thesis is as follows: Chapter 2 presents a detailed description of the link grammar formalism and the utilities provided by the link grammar parser. Chapter 3 presents some distinctive features of Turkish syntax and morphology with special emphasis on the concepts, which affect the design of our link grammar. In Chapter 4, a detailed architecture of our system and some special preprocessing that we do before the parsing step is described. The link grammar specification for Turkish is presented in Chapter 5. Chapter 6 includes an evaluation of our grammar based on results from our tests on a small corpus. Finally, in Chapter 7 we state our conclusions together with some suggestions for improvements to grammar.

Chapter 2

Link Grammar

2.1 Introduction

Link grammar[1] is a formal grammatical system defined by Sleator and Temperley in 1991 together with the development of efficient top-down dynamic programming algorithms to process grammars based on this formalism and construction of a wide coverage link grammar for English. This formalism, unlike to context free grammars, is lexical and it uses neither constituents nor categories. In fact, link grammars can be classified under the category of dependency grammars. In this formalism, a language is defined by a grammar that includes the words of the language and their linking requirements. A given sentence is accepted by the system if the linking requirements of all the words in the sentence are satisfied (connectivity), none of the links between the words cross each other (planarity) and there can exist at most one link between any pair of words (exclusion). A set of links between the words of a sentence that is accepted by the system is called a *linkage*. The grammar is defined in a *dictionary file* and each of the linking requirements of words is expressed in terms of *connectors* in the *dictionary file*.

In this chapter, first, link grammar formalism is explained. Then some special features of the link grammar parser and link grammar dictionary that we used in our Turkish link grammar are described.

2.2 Main Rules of the Grammar

A sequence of words is accepted by the language of a link grammar as a sentence if there exists a way of drawing the links between the words which satisfies the following conditions.

Planarity: Links do not cross.

Connectivity: The linkage for the sentence must include all the words and it must be a connected graph.

Satisfaction: The linkage must satisfy the linking requirements of all the words.

Exclusion: There can be at most one link between any two words.

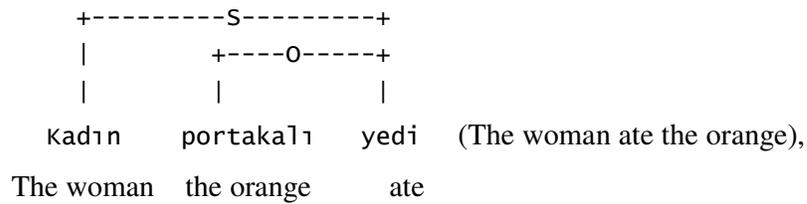
When a sequence of words is accepted, all the links are drawn above the words. Let us consider the following example:

yedi (ate): O- & S-;

kadın (the woman): S+ ;

portakalı (the orange): O+;

Here, the verb “yedi”(ate) has two left linking requirements, one is “S”(subject) and the other is “O”(object). On the other hand, the noun “kadın” (the woman) needs to attach to a word on its right for its “S+” connector and the noun “portakalı”(the orange) has to attach a word on its right for its “O+” connector. Since the word, “yedi”(ate) and “kadın” (the woman) have the same “S” connector, i.e. same linking requirements, with opposite sign they can be connected by an “S” link. A similar situation occurs between the words “portakalı”(the orange) and “yedi”(ate) for the “O” connector. Therefore, if these words are connected in the following way, all of the linking requirements of these words are satisfied.



In this sentence, “kadın”(The woman) links to word “yedi”(ate) with the S (subject) link and “portakalı”(the orange) links to word “yedi”(ate) with the O (object) link.

2.3 Language and Notion of Link Grammars

A *dictionary file* in link grammar consists of words and a block of connectors for each of these words specifying their linking requirements. Connectors can take plus sign meaning pointing to the right, or can take minus sign meaning pointing to the left. A right pointing connector connects to a left pointing connector with the same type and hence forms a link. A set of words are accepted by the grammar if there exist a way to link all the words. In this case, a *linkage*, which is a connected graph, is created.

2.3.1 Rules for Writing Connector Blocks or Linking Requirements

Connector names consist of one or more uppercase letters. They can also contain a sequence of subscripts. *Subscripts* are either lowercase letters or “*”s.

Connectors match to form a link if they have the same name (sequence of uppercase letters part) and their subscripts also match. To test whether two subscripts match, first their lengths are made same by appending necessary number of “*”s to the shorter one. A “*” character matches to any lowercase letter. Then if these two subscripts match and connectors have the opposite sign, being the word with the “+” signed connector on the left hand side of the word

with the “-“ signed connector, a link between these two connectors can be drawn. For example “D-“ matches both “D_n+” and “D_g+”, “S_s-“ matches “S_f+”, “S+” and “S_{ss}+” but not “S_{fp}+” or “S_p+”.

Formulas describing the linking requirements of words can also be combined by the binary associative operators *conjunction* (&) and *exclusive disjunction* (or) [1]. To satisfy the conjunction of two formulas both formulas must be satisfied, whereas to satisfy the disjunction of two formulas only one of the formulas must be satisfied.

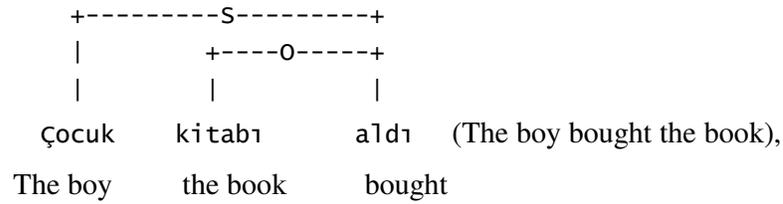
Optional links are contained in curly brackets {...}. An equivalent way of writing an optional expression like "{X-}" is "(X- or ())". This can be useful, since it allows a cost to be put on the no-link option [4]. Undesirable links are contained in any number of square brackets [...].

A multi-connector symbol “@” is used when a word can connect to one or an indefinite number of links of the same type. This is used, for example, when any number of adjectives can modify a noun.

For disjunction expressions, such as “A+ or B+”, and for conjunction expressions between connectors with opposite signs, like “A- & B+”, the ordering of the elements is irrelevant [4]. However when connectors with the same sign are conjoined, order of the operands becomes important. For these operands the further to the left the connector name, the closer the connection must be. For instance, according to the following rule:

aldı (bought): O- & S-;

The verb “aldı” (bought) takes both an object and a subject to its left but the object must be closer to it. Let us consider the following example sentence:



In this sentence, “çocuk”(The boy) links to word “aldı”(bought) with the S (subject) link and “kitap”(the book) links to word “aldı” with the O (object) link.

A dictionary entry consists of one or more words, followed by a colon, followed by a connector expression, followed by a semi-colon. The dictionary consists of a series of such entries. Any number of words can be put on the left of the colon and they are separated by spaces. Then all of them possess the linking requirement in that rule. For example, according to the following rule, all three words possess the same linking requirement "A+".

red small long: A+;

2.3.2 The Concept of Disjuncts

For the mathematical analysis of link grammar and for easy development of the necessary algorithms to process them, Sleator and Temperley[1] introduced another way of expressing link grammar, namely *disjunctive form*. A *disjunct* is a set of connector types that constitutes a legal use of a word and corresponds to one particular way of satisfying the requirements of a word. Therefore, linking requirements of a word can be converted into to set of all the legal uses of the word, namely a *set of disjuncts*. A disjunct has two parts: the left list and the right list. These lists are the ordered list of connector names and left list consists of the connectors with the “-“ sign, whereas the right list consist of the connectors with the “+” sign. Therefore, the left list defines the left hand linking requirements, whereas the right list defines the right hand requirements of a word. A disjunct is denoted as: $((L_1, L_2, L_3 \dots L_x)(R_y, R_{y-1}, R_{y-2} \dots R_1))$. In this

formalism, the list consisting of “L” type connector denotes the left hand side linking requirements of the word, while the second list denotes the right hand side linking requirements. Either “x” or ”y” can be zero. On the left side, the word connected to current word with “L₁” link is closer than to the word with “L₂” link. On the right hand side, the word connected to current word with ”R_y” link is closer than to the word with ”R_{y-1}” link.

A formula can be translated into a set of disjuncts by enumerating all the ways that the formula can be satisfied. In reverse direction, to translate a set of disjuncts into a formula, all the disjuncts should be combined with the “or” operand. For the following rule,

kitab (book) çocuk (child): (S+ or O+) & {D-};

The following disjuncts can be constructed.

((), (S+))

((), (O+))

((D-), (S+))

((D-), (O+))

2.4 General Features of the Link Parser

The following features are used by the link parser and they help the easy development of a link grammar for a natural language [1] .

Macros: Macros can be used in the dictionary. Macros are used for naming the linking requirement formulas those are used many time throughout the dictionary. For example, one can define a macro for the general linking requirements of the nouns with a name <noun-general> and then can use it as an ordinary connector in the formulas of both singular and plural nouns.

Word Files: Word files can be used instead of listing all the words with a particular linking requirement in just one long dictionary file. In this case, instead of a word, the relative path of the file that includes the list of all words with the same disjunct set can be used on the left hand side of the formulas.

Word Subscripts: If a word has more than one part of speech tag, then it can be used in different roles and hence, it should be included in different dictionary entries by following each of them with a different subscript. For example in Turkish, the word “hızlı “ means both “fast” (adjective) and “quickly” (adverb), thus in the dictionary for the word “hızlı” there can be two items; one is “hızlı.e”(e for adverb) with the other adverbs and the other is “hızlı.a” (a for adjective) with the other adjectives.

Cost System: When the parser finds more than one linkage for a given sentence, it looks at the total lengths of the linkages and outputs the one with the lowest length first. In addition to this heuristic, it is possible to design the grammar in such a way that some of connectors are given a cost and hence when outputting the solutions, the linkages with these connectors are not given priority. To assign a cost to a connector it is surrounded by square brackets[4]: For example, the connector “[A+]” receives a cost of 1; “[+[A+]” receives a cost of 2; etc. When outputting the solutions, the parser sorts them first according to the cost system and second according to the total lengths of the linkages.

2.5 Special Features of the Dictionary

In addition to the general features of the parser, the dictionary has also many useful built-in features for solving problems encountered in the development of parsers like unknown words, hyphenated expressions, numeric expressions, idioms, and punctuation symbols.

Capitalization: The parser is case sensitive. But there is a special category in the link grammar file called “CAPITALIZED_WORDS” which is used as the default category for the words those begins with a capital letter and does not included in none of the word lists. The authors assumed that most of the words with the first letter in uppercase were nouns, and hence types of the some unknown words can be estimated in this way. However, when this word is at the beginning of the sentence, it is handled in a bit different way. When such a word is encountered, the parser looks for both its original form and its lowercase form. If the parser finds its both forms in the grammar, then it uses both of them. Nevertheless, if it cannot find any of these forms, then the parser assigns the word to “CAPITALIZED_WORDS” category. A similar situation occurs after colons.

Hyphenated Words: Because in English hyphenated words are used productively, another special category used in the grammar is "HYPHENATED_WORDS" category. If a word contains a hyphen and is not included in the grammar, then it is automatically assigned to this category. In this way instead of listing all the hyphenated words in the grammar, they are recognized automatically.

Number Expressions: To be able to automatically handle the numeric expressions, the parser has the "NUMBERS” reserved category. So, strings consisting entirely of digits, period, decimal point, comma and colon are assigned to this category.

Unknown Words: The parser has a nice feature word guessing the unknown word role in the sentence. To use this feature one can define a category, "UNKNOWN-WORD.x". The authors used “n” (for nouns), “v” (for verbs), ”a” (for adjectives) and “e” (for adverbs) subscripts in their link grammar for English. If these categories are defined in the grammar, when the parser encounters an unknown word in a sentence it tries the linking requirements of all these categories to create a valid linkage for the sentence and hence it outputs the

successful solutions. In other words, in this way, the parser guesses the part of speech tags of unknown words. With the version 4 of the link parser, the parser has another new feature to handle unknown words, namely morpho-guessing for English. It is a system for guessing the part of speech tag of an unknown word by looking at its spelling. Words ending in “-s” are guessed to be plural nouns or singular verbs, those ending in “-ed” are guessed to be past tense or passive verbs, those ending “-ing” present participles and those ending in “-ly” adverbs.

To handle unknown words the parser acts in the following order:

- a) If the word is the first word of a sentence and its first letter is uppercase, then convert it to lowercase and perform the following step on both forms.
- b) If there are special symbols like punctuation symbols in the string, then break the word into sub-strings and perform the following steps on each of them.
- c) Check if it is included in the grammar.
- d) If it is not included, and begins with a capital letter, assign it to the category "CAPITALIZED-WORD".
- e) If it is not included, and contains “-” character assign it to the category "HYPHENATED-WORDS".
- f) If it is not included, and consists of only digits and some special punctuation symbols, assign it to the category "NUMBERS".
- g) If its type cannot be found, try morpho-guessing strategies.

- h) If its type cannot be found, try assigning it to "UNKNOWN-WORD.x" categories.
- i) At the end if the parser cannot find a reasonable solution for the unknown word, the parser gives the "the following words are not in the dictionary: [whatever]" message and stop searching for the solution.

The Walls: In some special cases like question sentences and imperatives, especially when a sentence lacks a subject, to sign the beginning and end of the sentence might be useful. This is provided by the "LEFT-WALL" and "RIGHT-WALL" predefined categories. If the "LEFT-WALL" category is included in the grammar, then a dummy word (LEFT-WALL) is inserted at the beginning of each sentence. In this case, because of the connectivity rule, "LEFT-WALL" is seen as a normal word and it has to be connected to the rest of the sentence. In addition to the "LEFT-WALL", there are cases where "RIGHT-WALL" is needed like some special punctuation symbols but it is not as important as "LEFT-WALL".

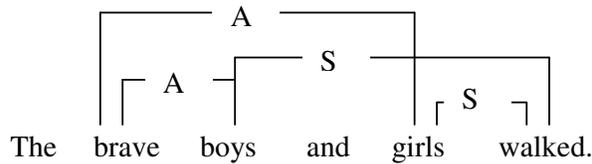
Idioms: In the grammar, an ordered set of words can be defined as a single word. In this way, some special two-word passives like "dealt with"; "arrived-at" and idioms can be handled easily. These expressions should be included in the grammar by joining them with underbars. When the parser encounters the idiomatic expressions, it prints them as different words and links them by special dummy links with arbitrary names of the form IDAB, where A and B characters are arbitrary.

2.6 Coordinating Conjunctions

Coordinating conjunctions have different characteristic that make them very difficult to express in the link grammar formalism. As stated before, the most important rule that link grammar formalism based on is the Planarity rule. Most

of the phenomena in natural languages fit naturally into planarity rule, whereas coordinating conjunctions in some cases seem to result in crossing links.

In the following sentence, the adjective “brave” modifies both of the nouns, “boys“ and “girls”, and because each of these nouns are the subject of the verb “walked”, links are crossed and hence the planarity rule is violated.



Authors solved the problem for English by a hand-wired solution and in the following subsections; the solution devised by the authors is discussed in detail.

2.6.1 Handling Conjunctions

To be able to handle conjunctions in English, authors define some new notions and redefine coordinating conjunctions from their perspective.

Given a sentence “S”, part of this sentence “L” is defined as a “well-formed ‘and’ list” if it satisfies the following conditions. “L” should consist of elements delimited by either “,” or “and”, while the last delimiter being either “and” or “, and”. For example in the sentence “Ali, Ayşe and Veli go to school”, the substring “Ali, Ayşe and Veli” is a “well-formed ‘and’ list”. The delimiters “,” and “and” are not accepted as elements of the list.

- Each string produced by replacing “L” with one of its elements should be a valid sentence of the link grammar.
- In all of the sentences, created by replacing “L” by one of its elements, there should be a way of creating a valid linkage such that for each

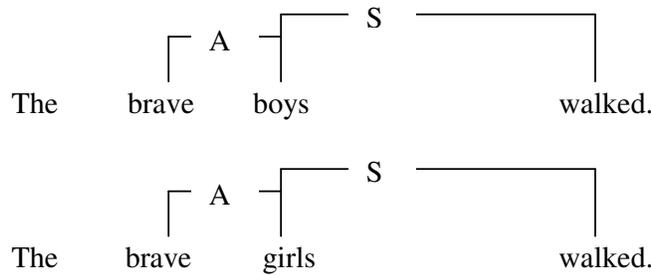
sentence, the element should link to the rest of the sentence with the same set of links to the same set of words.

The following sentence satisfies all these conditions.

S: The brave boys and girls walked.

L: boys and girls

Elements of L: {boys, girls}



As it can be seen, the sentences created by replacing the list with its elements also links to the rest of the sentence with the same set of link to the same set of words.

This definition of “and” and “well formed ‘and’ list” allows many ungrammatical sentences like “Ali bought the apple Ayşe and banana Veli eat”. Hence, the problem with the definition is that it does not impose any relation requirement between the elements of “well-formed ‘and’ list”.

The authors devised two methods to overcome this problem. First is to restrict the set of connectors that can be used while linking the elements of the list to the rest of the sentence by simply adding these connectors to the “ANDABLE-CONNECTORS” list in the grammar.

Second is the refinement of the definition of “well-formed ‘and’ list” with the addition of the following condition: Only one of the words of each element must

be connected to the rest of the sentence. However, the number of links from this word to the rest of the sentence is not limited.

2.6.2 Some Problematic Conjunctional Structures

- Because only one of the words of each element must be connected to the rest of the sentence, the sentence given below cannot be handled.

```

+-----Os-----+
+-----Os-----+      |
+-----Os-----+      |      |
+---Os---+      |      |      |
+--Ss--+  +-DS--+      |      +-DS--+      |
|      |  |      |      |      |      |
Ayşe gave.v a book.n to Ali and a pencil.n to Veli.

```

This problem remains in the Author’s current system for English.

- Embedded clauses creates problem.

```

+-S-+-C-+-S-----+
|  |      |      |
I think John and Dave ran

+-S-+-+
|      |
I think John and Dave ran

```

To prevent these kinds of linkages, Authors have implemented a post processing system. After expanding the conjunction sentences into several sub-sentences by replacing “well-formed ‘and’ list” with its elements, domain structure of each of these sub-sentences are computed. At the end, if the nesting structure of a pair of links, descending from the same link, has the same domain ancestry, then the original linkages is accepted.

- Current system developed for English does not handle different constraints for different conjunctions, e.g. “Ayşe ate apple but orange”.

2.7 Post-Processing

2.7.1 Introduction

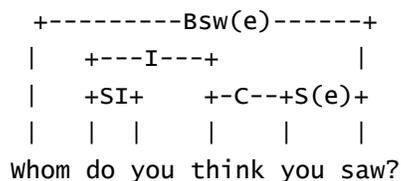
To handle some phenomena that cannot be handled with the link grammar formalism like coordinating conjunctions, the authors developed a post processing system based on *domains*. A domain contains a subset of the links in a sentence. The parser divides the sentence into domains based on the types of the links that start them after finding a linkage for it. It then further divides the sentence into groups and each group consists of links with the same domain membership. Then, the parser decides on the validness of the linkage by testing the rules related with the current group to the links. The post-processing system is partially hand-wired.

2.7.2 Structures of Domains

“Root link” of a domain, in other words a certain type of link starts a domain. The “root word“ is the name given to the word on the left hand side of the “root link”. Most of the time, a domain contains all the links that can be reached from the right end of the root link. The examples given in this subsection are directly taken from [4]

```
+-----CO-----+
+-----XC-----+ |
+-C-+Ss(s)+O(s)+ | +Sp+
| | | | | | |
After he saw us , we left
```

In this example, “C“ link is the root link of (s)-type domain; hence, the links “Ss” and “O” on the right end of the “C” link are the members of “(s)-type” domain. But “Xc”, ”Co” and “Sp” links are not included in the group of “(s)-type” domain, since they cannot be reached from the right end of “C” link.



In this example, because “Bsw” link can be reached from the right end of the “C” link, it is also included in the “(e)-type” domain. Hence, in some cases domains might include the words on the left hand side of the root word.

There are three types of domains. The ordinary domains were explained above. The other two are “ulfr only” domains and “ulfr” domains. “ulfr” is an abbreviation for “Under left from right” and “ulfr only” domains includes all the links that can be reached from the left end of the root link tracing to the right. “ulfr” domains include the unions of the links included by ordinary domains and “ulfr only” domains.

In this domain structure, whether a domain includes its root link or not can be controlled. All the links with the same domain membership are said to create a group. In fact, groups or domains correspond to subject-verb expressions or clauses.

2.7.3 Rules in Post Processing

In natural languages, sometimes there can be constraints on the types of links that should or should not be found in a specific clause. If these constraints are related to links to the same word, with link grammar formalism these constraints can easily be enforced. However, there are cases where these constraints are related to links on different words and pure link grammar formalism is incapable of enforcing these constraints. To overcome this problem, post-processing system provides users with two types of rules. These are *contains-one* and *contains-none* kinds of rules. The general format of rules is:

X, Y Z, “Message!”

If this rule is listed under the *contains-one* category, it means that if a group contains “X” link, it also has to contain at least one “Y” or one “Z” link. If this rule is listed under the *contains-none* category, it means that if a group contains “X” link, it can contain neither “Y” nor “Z” link.

Chapter 3

Turkish Morphology and Syntax

In this chapter, first we explain some important distinguishing properties of Turkish syntax and morphology. Then, we move to the subset of Turkish morphotactical rules some of which are necessary to understand the system and some of which have some important syntactic consequences. Then, a brief description of constituent order in Turkish is given and the chapter is closed with the classification of Turkish sentences. All the material given in this chapter contains the necessary background information for the developed link grammar for Turkish. In addition, it draws the general scope of the work to be done.

3.1 Distinctive Features of Turkish

Turkish belongs to the Altaic branch of the Ural-Altaic language family and it has no *grammatical gender*¹. Other important distinguishing properties of Turkish concerning our link grammar listed in the following items.

- Turkish has *vowel harmony*. For this reason, during the affixation process, the vowels in the suffixes have to agree with the last vowel of the affixed word in certain aspects to achieve vowel harmony. For example, the question morpheme “mi” obeys this rule. The vowels

¹ Marking nominal words for gender (sexuality), e.g. “die blume” (the flowers) and “der tabelle” (the table) in German. Die is a determiner used for female nouns and der is used for male nouns.

related to the vowel harmony rule in each example are shown in bold and “+” is used to mark the related morpheme boundary.

- I. Geld**in** mi? (Did you come?)
- II. Yürü**dün mü**? (Did you walk?)
- III. Sen+**in** (Yours)
- IV. Göz+**ün** (of the eye)

In example I, the vowel “i” in the question morpheme “mi” does not change because it agrees with the last vowel “i” of the word “Geldin”. However, in example II, it turned into the vowel “ü”, to agree with the last vowel “ü” of the word “Yürüdün”. Similarly, in example III, the vowel “i” of the possessive marker suffix “in” did not change, while in example IV, it turned into vowel “ü”.

- In Turkish, the basic word order is *SOV*, but constituent order may vary freely as demanded by the discourse context. For this reason, all six combinations of subject, object, and verb are possible in Turkish.

(He is going to his home)

- | | | | |
|------|----------------|----------------|----------------|
| I. | O (Subject) | evine (Object) | gidiyor (Verb) |
| | He | His home | going |
| II. | Evine (Object) | o (Subject) | gidiyor (Verb) |
| | His home | he | going |
| III. | Evine (Object) | gidiyor (Verb) | o (Subject) |
| | His home | going | he |
| IV. | Gidiyor (Verb) | evine (Object) | o (Subject) |
| | going | His home | he |

V. O (Subject) gidiyor (Verb) evine (Object)
 he going his home

VI. Gidiyor (Verb) o (Subject) evine (Object)
 going he his home

- Turkish is *head-final*[7], meaning that modifiers always precede the modified item. Therefore in a sentence:

- Object of postpositions¹ precede postpositions.

Ayşe ile gittin. (You went with Ayşe)
 Ayşe with (you went)

- Adjectives precede nouns.

Cesur çocuk (The brave child)
 Brave child

- Indirect object precedes direct object.

Sentence: Ayşe took the book from the library.

Ayşe kütüphaneden kitabı aldı.
 Ayşe from the library the book took.

- Subject precedes predicate.

Ben gidiyorum. (I am going)
 I going

- Objects precede verb

¹ Postpositions are like of prepositions in English, but prepositions precede their objects in English while postpositions follows their objects in Turkish.

O evine gidiyor (He is going to his home)
 He His home going

- Adverbs precede verbs or adjectives.

Çok iyi bir iş (A very good work)
 Very good a work

- Turkish is an *agglutinative language*, with very productive inflectional and derivational suffixation¹. A given word form may involve multiple derivations[12]. Description of the morphological features used below can be found at APPENDIX A. In the following examples, the relation between a morpheme and a feature is shown by marking both of them with the same numbered subscript.

I. Sağlam+laş₁+tır₂+mak₃ (sağlamlaştırmak = to strengthen)

Sağlam+Noun+A3sg+Pnon+Nom ^DB+Verb+Become₁

^DB+Verb+Caus₂+Pos^DB+Noun+Inf₃+A3sg+Pnon+Nom

Number of word forms that one can generate from a nominal or verbal root is theoretically infinite[12].

- In Turkish syntax, most of the relations between words, such as those that are provided by some auxiliary words in English are accomplished using suffixes [8]. For example, in English, certain cases of noun phrases are formed by prepositions preceding nouns and verbal phrases are formed by prepositions preceding the verbs. This is because of the fact that in Turkish, inflectional suffixes have grammatical roles. In addition, words may take multiple derivational suffixes changing their POS, and each intermediate derived form can take its own inflectional suffixes

¹ Turkish has no native prefixes apart from the reduplicating intensifier prefix as in beyaz="white", bembeyaz="very white", sıcak="hot", sımıcak="very hot".

each of which contributes to the syntactic roles of the word. Hence, for Turkish, there is a significant amount of interaction between syntax and morphotactics. For example case, agreement, relativization of nouns and tense, modality, aspect, passivization, negation, causatives, and reflexives of verbs are marked by suffixes.

I. yap+tır₁+ama₂+yor₃+muş₄+sun₅ (you were not able to make him do)
 yap+Verb^{^DB+Verb+Caus₁} ^{^DB+Verb+AbleNeg₂} +Neg+Prog₃+Narr₄+A2sg₅

II. Araba+mız₁+da₂+ki₃+nin₄ (of the one that is in our car)
 araba+Noun+A3sg+P1pl₁+Loc₂^{^DB+Adj+Rel₃} ^{^DB+Noun+Zero+A3sg+Pnon+Gen₄}

- In Turkish, a modified item, i.e. head, should agree with its modifier, i.e. dependent, and this agreement is provided with the suffixes affixed to the modified item. For this reason, pronoun drop is encountered as sentences with covert subjects and, compound nouns with covert modifiers frequently[7], i.e. Turkish is a *pro-drop language*.

I. (Benim=my) Elbisem. (My dress)

II. (Ben=I) Geldim (I came)

3.2 Turkish Morphotactics

Morphemes in a language can be categorized into *inflectional morphemes* and *derivational morphemes*. In general, inflectional morphemes are used to mark grammatical information; e.g. case, number, agreement, whereas derivational morphemes create new words from existing ones with new meanings and even with new POS tags. *Morphotactics* specifies the ordering of these inflectional and derivational morphemes in a language. Ordering rules of inflectional morphemes, i.e. *inflectional morphotactics*, and derivational morphemes, i.e. *derivational morphotactics*, of words in Turkish are explained in this subsection.

3.2.1 Inflectional Morphotactics

Since the syntactic roles owed by inflectional morphemes are very important in Turkish, full set of the inflectional morphotactics is given in detail.

3.2.1.1 Verbal Inflectional Morphotactics

Verbs can take the following suffixes in the given order. The suffix responsible for the property and the property in the feature structure are given in bold. Full list of tense suffixes can be found in APPENDIX A.

I. Polarity:

a. Positive:

geldim (I came);

gel+Verb+**Pos**+Past+A1sg

b. Negative:

gel**medim**(I did not came).

gel+Verb+**Neg**+Past+A1sg

II. First Tense Suffixes:

gitti (went “Past tense”)

git+Verb+Pos+**Past**+A3sg

gidi**yor** (is going “Progressive tense”)

git+Verb+Pos+**Prog1**+A3sg

III. Second Tense Suffixes: They are similar to first tense suffixes and they are placed after the first tense suffixes.

Gitmiş**ti**_{2m} (past of narrative tense)

git+Verb+Pos+**Narr**₁+**Past**₂+A1sg

In this example the first tense is the narrative tense and the both the feature and the morpheme responsible for this tense is numbered with the same subscripted number 1. The second tense is the past tense and same marking method is used for this tense. Full list of the second tense features can be found in APPENDIX A.

IV. Person Suffixes:

- a. A1sg for first singular: geldi+m (I came)
gel+Verb+Pos+Past+A**1sg**

- b. A2sg for second singular: geldi+n (you came)
gel+Verb+Pos+Past+A**2sg**

- c. A3sg for third singular: geldi (he/she came)
gel+Verb+Pos+Past+A**3sg**

- d. A1pl for first plural: geldi+k (we came)
gel+Verb+Pos+Past+A**1pl**

- e. A2pl for second plural: geldi+niz (you came)
gel+Verb+Pos+Past+A**2pl**

- f. A3pl for third plural: geldi+ler (they came)
gel+Verb+Pos+Past+A**3pl**

3.2.1.2 Nominal Inflectional Morphotactics

Nominal¹ words can take the following suffixes in the given order. Related suffix in each of the following examples is shown in bold case. Full morphological analyses of the words are given next to words.

I. Plural suffixes:

a. A3sg for singular: kitap, kitap (book)

kitap+Noun+A**3sg**+Pnon+Nom

b. A3pl for plural: kitaplar, kitap+**lar** (books)

kitap+Noun+A**3pl**+Pnon+Nom

II. Possessive marker:

a. P1sg: kitabım, kitap+**ım**(my book)

kitap +Noun +A3sg +**P1sg** +Nom

b. P2sg: kitabın, kitap+**ın** (your book)

kitap +Noun +A3sg +**P2sg** +Nom

c. P3sg: kitabı, kitap+**ı** (his/her book)

kitap +Noun +A3sg +**P3sg** +Nom

d. P1pl: kitabımız, kitap+**ımız** (our books)

kitap +Noun +A3sg +**P1pl** +Nom

e. P2pl: kitabınız, kitap+**ınız** (your books)

kitap +Noun +A3sg +**P2pl** +Nom

¹ Nominal words include nouns, pronouns, adjectives, and adverbs.

f. P3pl: kitapları ,kitap+**ları** (his/her books)

kitap +Noun +A3sg +**P3pl** +Nom

In fact, there is ambiguity in the last example. Same word, has also the following meanings and analyses:

kitap +Noun +A3pl +P3pl +Nom (their books)

kitap +Noun +A3pl +Pnon +Acc (of the books)

kitap +Noun +A3pl +P3sg +Nom (their book)

III. Case Markers:

a. Nominative: kitap, kitap (book)

kitap+Noun+A3sg+Pnon+**Nom**

b. Locative: kitapta, kitap+**ta** (at the book)

kitap+Noun+A3sg+Pnon+**Loc**

c. Ablative: kitaptan, kitap+**tan** (from the book)

kitap+Noun+A3sg+Pnon+**Abl**

d. Dative: kitaba, kitap+**a** (to the book)

kitap+Noun+A3sg+Pnon+**Dat**

e. Accusative: kitabı, kitap+**ı** (the book)

kitap+Noun+A3sg+Pnon+**Acc**

f. Instrumental: kitapla, kitap+**la** (with the book)

kitap+Noun+A3sg+Pnon+**Ins**

- g. Genitive: kitabın, kitap+ın (of the book)
 kitap+Noun+A3sg+Pnon+Gen

A few examples illustrating the usage and order of these markers are given below. The morphological feature-morpheme relation is indicated by numbering them with the same subscript.

- I. Kitaplarımızda, kitap+lar₁+ımız₂+da₃ (at our books)
 kitap+Noun+A3pl₁+P1pl₂+Loc₃

- II. Kitabının, kitap+ın₁+ın₂ (of your book)
 kitap+Noun+A3sg+P2sg₁+Gen₂

- III. Kitabının, kitap+ı₁+nın₂ (of his/her book)
 kitap+Noun+A3sg+P3sg₁+Gen₂

3.2.2 Derivational Morphotactics

In Turkish, both the verbal and nominal words can take many derivational suffixes details of which can be found in [11]. In addition, in a derived word with many derivational steps from a root word, each intermediate derived word may have its own inflectional features. Some of these derivations with important outcomes in language are explained in detail.

3.2.2.1 Verbal Derivational Morphotactics

Through affixation of some derivational suffixes, new verbs, adverbs (*gerunds*), nouns (*infinitives or verbal nouns*), and adjectives (*participles*) can be derived. In this section, derivations that result in changes to syntactic roles of the verbs are explored.

The first types of these derivations are the ones with changes to the POS of the verbs, namely gerunds, participles¹, and infinitives. They are used to construct different types of dependent clauses, i.e. DC's, without subordinating conjunctions or relative pronouns. In the following examples, suffixes deriving gerunds, participles, and infinitives from verbs are shown in bold and the full morphological feature structures of each of these words are given at the end of each example.

Gerunds:

(You left when he saw me)

- I. O beni gör+**ünce** ayrıldın.
 He me when he saw you left
 (gör+Verb+Pos^DB+Adverb+When)

Gerunds are adverbs derived from verbs by affixation of some special derivational suffixes. They are used to construct subordinate clauses² and the derivational suffix that they take plays a syntactic role similar to a subordinating conjunction in English.

Participles:

(The dog that chased me was black)

- II. Beni kovala+**yan** köpek siyahtı.
 Me that chased the dog was black
 (kovala+Verb+Pos^DB+Adj+PresPart)

Participles are similar to gerunds with the last POS being an adjective. They are used for introducing relative clauses. Hence, participle-producing suffixes behaves like the relative pronouns in English.

¹ e.g. participles are used as relative clauses and a relative clause is a subordinate clause that modifies a noun

² Adverbial dependent clauses

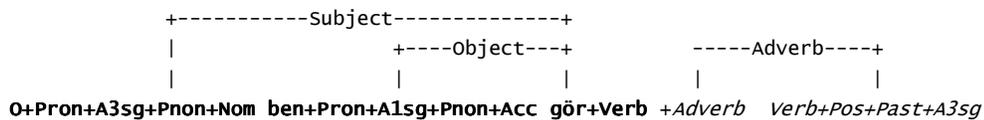
Infinitives:

(I cannot understand why he is so crude.)

III. Bu kadar kaba ol+uş+un+u anlayamıyorum.
 So crude he is I cannot understand
 (ol+Verb+Pos^DB+Noun+Inf3+A3sg+P3sg+Acc)

Similar to participles, *infinitives* are used to introduce noun clauses through suffixation of derivational suffixes and these suffixes can be assumed to correspond to relative pronouns in English.

These structures are a consequence of the morphosyntactic properties of the derivations in Turkish. If a word is assumed as a sequence of derivations each with its own inflectional suffixes, each of its intermediate derivations preserves its syntactic roles as a modified in the modifier-modified relation, and only the last derivation, that is the resulting POS, contributes to the word’s syntactic role as a modifier¹. For the example sentence given in example I, the following is the morphological feature structures of the words.



As it can be seen, this sentence can be assumed to consist of two clauses, first one, which is a DC, in bold and second one in italics. In the example, the verb “gör”(see) is derived to an adverb. Nevertheless, it still plays the role of a modified as a verb (intermediate derivation) and hence, the first clause, “O beni gör” expresses the assertion “He saw me”. On the other hand, because of the adverbial derivational suffix, it plays the role of an adverb modifier on the right hand side. So, the verb² in the second clause “ayrıldı” (He left) is modified by

¹ Please remember, in Turkish, modifier always precedes the modified.
² In fact a sentence with only a verb is possible in Turkish.

this last derivation to get resulting meaning “O beni görünce ayrıldı” (He left when he saw me) by connecting the DC to the main clause.

The second types of these derivations are the ones with changes to the category of verbs according to their valence, namely causative suffixes. In addition, in Turkish, appropriate combinations of multiple causations are allowed.

Valence at the beginning	Initial word	Word after causation	Valence after causation
Intransitive	Dinlenmek (to take rest)	Dinl endir mek (to make somebody to take rest)	Transitive
Transitive	Yazmak (to write)	Yaz dır mak (to make write)	Ditransitive
Intransitive	Ölmek (to die)	Öl dür mek(to kill)	Transitive
Transitive	Öldürmek(to kill)	Öldür t mek(to have someone killed)	Ditransitive

Table 1 Effects of Causation to Verbs

3.2.2.2 Nominal Derivational Morphotactics

One important property of Turkish is that, all adjectives can be used as nouns, i.e. all adjectives can derive into a noun with zero morphemes. Then, the adjective is used as a noun with the property of the adjective.

- I. Çocuk kırmızı giydi. (The child wore something red)
The child red wore

In this example, the adjective “kırmızı”(red) is used as a noun with the meaning “something red”.

In Turkish, nouns, like verbs, have a rich derivational morphology and they take many suffixes that produce new adverbs, nouns, verbs, adjectives, and nominal verbs, i.e. like copula. Some examples to these derivations are given below:

1. Yardım+la₁+ş₂ (to help each other) (“yardım” means help in Turkish)
yardım+Noun+A3sg+Pnon+Nom ^DB+Verb+Acquire₁ ^DB+Verb+Recip₂+Pos
2. hız+l₁+ca₂ (speedy)
hız+Noun+A3sg+Pnon+Nom^DB+Adj+With₁^DB+Adverb+Ly₂

3.2.3 Question Morpheme

In Turkish, question morphemes starting with “mH“ are written as a separate word, but the lexical “H” has to harmonize with the last vowel of the preceding word[11]. In the following examples, question morphemes are in italics and the last vowels of the preceding words are in bold face.

- I. Tezi yazmaya başladın mı? (Did you begin to write the thesis?)
Thesis to write you begin question suffix
- II. Öldü mü? (Did he die?)
He die question suffix

All nominal and verbal words can take question morpheme in Turkish. This basic form of question morpheme, regular question morpheme, just gives a negative meaning to the sentence, and does not change its syntactic structure. Hence, it does not have a syntactic role. Sentences given in I, and II are examples to this form. On the other hand, a question morpheme can also take

tense, person, and copula suffixes. These suffixes derive the question suffix into verb resulting it to take the new syntactic role of verbs. We call this type of question morpheme “*question morpheme with copula*”, hereafter.

I. He is the man who gossip about you.

Senin	hakkında	konuşan	adam.
You	about	who gossip	man, he

II. Am I the one who gossip about you?

Senin	hakkında	konuşan	adam mıyım. (mi+Ques+Pres¹+A1sg)
You	about	who gossip	the one, am I

Note that in the last example, mi question morpheme have both the tense and person suffixes, i.e. (**mi+Ques+Pres+A1sg**).

3.3 Constituent Order in Turkish

Figure 2 summarizes the order of the constituents in Turkish sentences[14]. However, order of the constituents may change rather freely due to a number of reasons:

- Any indefinite constituent immediately precedes the verb[10]:

Sentence: The child read the book on the chair

I. Çocuk	<i>kitabı</i>	sandalyede	okudu.
The child	<i>the book</i>	on the chair	read.

In this example the definite direct object, “*kitabı*” precedes the indirect object “*sandalyede*”.

¹ “Pres” is one of the verb (in the present tense) driving suffixes from nominal words

II. Çocuk sandalyede kitap okudu.
 The child on the chair book read.

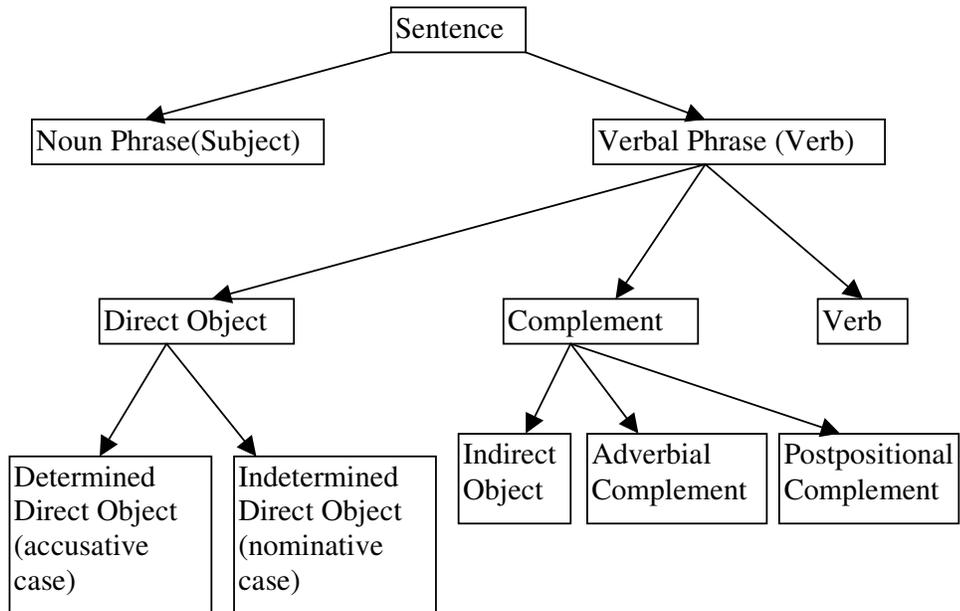


Figure 2 Typical Order of Constituents in Turkish

However, in example II, since the direct object “*kitap*” is indefinite, it follows the definite indirect object “*sandalyede*” and immediately precedes the verb.

- A constituent to be emphasized is placed immediately before the verb.

Sentence: Pınar read the book

I. Pınar *kitabı* okudu.
 Pınar *the book* read

II. *Kitabı* Pınar okudu.
 The book Pınar read

- If the expression to be emphasized is of time, instead of immediately preceding the verb, it is placed at the beginning of a sentence.

Sentence: I came from home yesterday.

- | | | | |
|-----|------------------|------------------|---------|
| I. | Evden | <i>diin</i> | geldim. |
| | From home | <i>yesterday</i> | I came |
| II. | <i>Diin</i> | evden | geldim. |
| | <i>Yesterday</i> | from home | I came |

- In addition, types of adverbial complements can be scramble freely.
- Since daily conversations are directed by the natural flowing of emotions and thoughts, the place of the verb in such sentences is not the end as opposed to normal sentences in which verb is at the end. These kinds of sentences are named as *inverted sentences*. For example, in the colloquial, an imperative often begins a sentence, because someone with urgent instructions to give naturally put the operative word first: "Çık oradan" (Get out of there)[10].

3.4 Classification of Turkish Sentences

Turkish sentences can be classified according to their structure, to the type of their predicates, to the place of their predicates, i.e. according to the order of constituents, and to the meaning of the sentence. Classification of Turkish sentences can be summarized as follows:

- a. **By Structure**
 1. Simple Sentences
 2. Complex Sentences

3. Ordered/Compound Sentences

b. By predicate type

1. Nominal Sentences

2. Verbal Sentences

c. By predicate place

1. Regular Sentences

2. Inverted Sentences

d. By meaning

3. Positive Sentences

4. Negative Sentences

5. Imperative Sentences

6. Interrogative Sentences

7. Exclamatory Sentences

3.4.1 Classification by Structure

Simple sentences contain only one *independent clause*, i.e. IC, with no *dependent clauses*, i.e. DC.

I. Ben okula gidiyorum. (I am going to the school)

A complex sentence is a sentence with one IC and many DC's.

I. Senin yaşadığın ev çok lüks.(The house that you live in is very luxury.)

A conditional sentence is treated as it is in the class of *complex sentences*. In conditional sentences, DC connected to the IC by a condition, result, or reason relation.

I. Sen okula gidersen ben gelmem. (If you go to the school I will not come)

A *compound (ordered) sentence* consists of at least two independent clauses and zero or more dependent clauses joined by conjunctions and/or punctuation¹.

Independent ordered sentence are a subcategory of compound sentences. They consist of independent clauses and there is neither semantic relation between these independent clauses nor common constituents². They are conjoined by either commas or semicolons.

- I. Nöbetçi bile benden korkmaz, isterseniz kendisine sorunuz. (Even the guard does not afraid of me, if you want you can ask him.)³

Dependent ordered sentences are another subcategory of compound sentences. In spite of independent ordered sentences, there is a semantic relation between their independent clauses and this relation is provided through conjunctions or common constituents.

- I. Çocuk konuyu okudu ve anladı. (The child read and understood the subject).

3.4.2 Classification by Predicate Type

A *verbal sentence* is a sentence whose predicate is a finite verb.

- I. Ben okula gidiyorum. (I am going to the school.)

In a *nominal sentence*, the predicate can be either a nominal word or a verb derived from a nominal word by some special suffixes. Copula⁴ is one of these suffixes. However, in informal speech, copula suffix is omitted frequently and

¹ Commas, semicolons or conjunctions

² Except implicit common subject

³ This example is taken from [14]

⁴ “-dır” is the suffix with the copula role in Turkish.

hence, in Turkish, nominal words and phrases; i.e. nouns and noun phrases, pronouns, adjectives and adjectival phrases, adverbs and adverbial phrases can play the role of verbs. This situation is referred as “*suffixless nominal to verbal derivation*”, hereafter. In the following examples, suffixes producing verbs from nominal words as the copula suffix “**dır**” is in bold face and the nominal with the predicate role is in italics.

- | | | | | |
|------|-------|---------|--------------------|---------------------------------------|
| I. | Benim | elbisem | <i>mavidir.</i> | (My dress is blue) |
| | My | dress | is | blue |
| II. | Benim | elbisem | <i>mavi.</i> | (My dress is blue)(Copula is omitted) |
| | My | dress | is | blue |
| III. | O | benim | <i>kitabımdır.</i> | (It is my book) |
| | It | my | is | my book |
| IV. | O | benim | <i>kitabım.</i> | (It is my book) (Copula is omitted) |
| | It | my | my | book |

The words “var” (existent), “yok” (not existent), ”değil” (not) are the special words and they are used to construct nominal sentences.

- | | | | | | | |
|------|---------|---------|---------|---------------|---------------------|------------------------------------|
| I. | Masanın | üstünde | bir | kitap | var. | (There is a book on the table) |
| | Table | on | a | book | there is | |
| II. | Masanın | üstünde | bir | kitap | yok. | (There is not a book on the table) |
| | Table | on | a | book | there is not | |
| III. | O | benim | kitabım | değil. | (It is not my book) | |
| | It | my | my book | is not | | |

3.4.3 Classification by Predicate Place

In Turkish, sentences can be classified according to the place of the verb. If the place of the verb is not the end of the sentence, it is named as an *inverted sentence* and else it is called as a *regular sentence*. All of the following combinations are types of inverted sentences, SVO, OVS, VSO, and VOS. In the following example verb is in bold case.

- I. Kitabı **aldım** ben. (I bought the book)
The book bought I

3.4.4 Classification by Meaning

Declarative sentences are the most common type of the sentences and they are used make statements. *Positive and negative sentences* are types of declarative sentences according to the polarity of the verb. The suffix used to give the negative polarity meaning is in bold case in the example II, i.e. without any suffix, verbs have positive polarity meaning in Turkish.

- I. Ben okula gideceğim (I will go to school) (positive)
I to the school will go
- II. Ben okula gitmeyeceğim (I will not go to school) (negative)
I to the school will not go

Imperative Sentences are used make a demand or a request.

- I. Gel buraya. (Come here.)
Come here

Interrogative Sentences (questions) are used to request information. In the following examples, the question words and suffixes are in bold case.

I. Okula **kim** gidiyor ? (Who is going to the school?)
To the school who going

II. Ayşe okula gidiyor **mu**? (Is Ayşe going to the school?)
Ayşe to the school going question suffix

Exclamatory Sentences are generally more emphatic forms of statements:

I. Ne harika bir gün! (What a wonderful day!)
What wonderful a day!

3.5 Substantival Sentences

Sentences functioning as nouns or adjectives within longer sentences are named as substantival sentences[10]. These are frequently encountered in Turkish, especially in colloquial. Quotations and paraphrases are a sort of substantival sentences.

I. “Güneş daha batmadı” dedi.¹ (“The sun has not yet set”, he said)
The sun yet not set she/he said

Here the quoted words are the direct object of the verb dedi. (She/he said).

II. Kuş uçmaz kervan geçmez bir yer². (An inaccessible place)
Bird does not fly caravan does not pass a place

In the previous example, the substantival sentence “Kuş uçmaz kervan geçmez” is used as an adjective, which modifies the noun “yer”(place).

¹ This example is directly taken from Lewis.

² This example is directly taken from Lewis.

III. Olmaz cevabı (The answer “it is not possible”)
“it is not possible” the answer

In example III, the sentence “olmaz”(it is not possible) is used to construct a noun phrase in which it has the syntactic role of noun modifier.

Chapter 4

Design

4.1 Morphological Analyzer

As mentioned in the previous sections, Turkish is an agglutinative language with very complex morphotactics and morphological features have important syntactic roles. For this reason, the role of a morphological analyzer is very important. Hence, the one developed by Oflazer [11] using PCKIMMO [15], a full two level specification of Turkish morphology, Turkish Morphological Analyzer, TMA hereafter, is used in our system.

4.1.1 Turkish Morphological Analyzer

TMA is developed in PCKIMMO[15] using two-level morphology formalism by Oflazer[11]. It consists of about 23.000 root words and almost all of the morphological rules of Turkish in its lexicon files and 22 two-level orthographic rules in its rule file. Almost all of the special cases and exceptions to orthographic¹ and morphological rules are handled using two level morphology and finite state machines.

Turkish is an agglutinative language with very complex derivational and inflectional morphotactics. Morphemes added to a root word or a stem can convert the word from a nominal to a verbal structure or vice-versa, or can

¹ For example, vowel harmony in Turkish is an orthographic (phonological) rule.

create adverbial constructs[11]. For example, the word “sağlamlaştırma” (to strengthen) can be broken down into morphemes as follows:

sağlam+laş+tır+mak

There are a number of phonetic rules, which constrain and modify the surface realizations of morphological constructions. Vowels in the suffixes of a word have to agree with its the last vowel in certain aspects to achieve vowel harmony, although there are some exceptions. In some cases, vowels in the roots and morphemes are deleted. Consonants in the root words or in the suffixes undergo certain modifications, and they are sometimes deleted in a similar manner. In addition, there are a large number of words that are assimilated from foreign languages; i.e. Persian, Arabic; and English, with exceptions to these rules[11]. Architecture of this TMA, which is based on two-level morphology, is depicted in Figure 3.

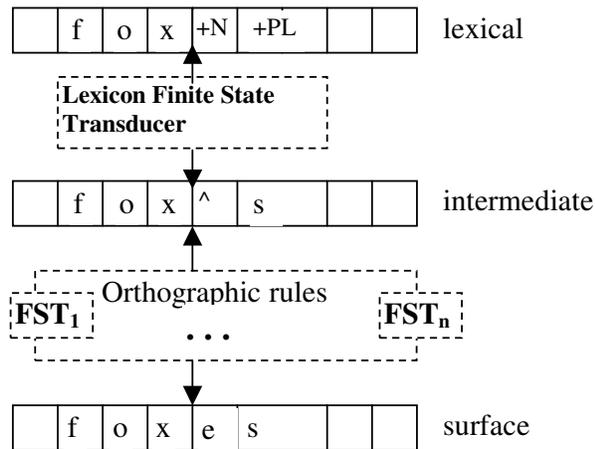


Figure 3 Architecture of a Two Level Morphological Analyzer¹

The lexicon transducer maps between the lexical level, with its stems and morphological features, and an intermediate level, which represents a simple concatenation of morphemes. Then, a set of transducers runs in parallel and they

¹ This figure is taken from [16].

map between the intermediate and surface levels. Each of these transducers represents a single orthographic rule. In Figure 3, a trace of the system accepting the mapping from “fox+N+PL” to “foxes” is given as an example.

4.1.2 Improvements and Modifications to Turkish Morphological Analyzer

Before developing our Turkish Link Grammar, we made some modifications and improvements to this two level Turkish morphological analyzer. First, we make the necessary changes to TMA for handling special Turkish characters, which are Ğ, ğ, Ü, ü, Ş, ş, İ, ı, Ö, ö, Ç, ç. “Çocuk” (child), “şirket” (company), “ırkçılık” (racism), “ürkmek” (to scare), “ölmek” (to die), “soğutucu” (cooler) are some example words with these characters. In addition, Turkish lexicon and rules files are modified to work with the version 4 of PCKIMMO to run on windows platform. Moreover, the followings are the other important modifications to TMA.

- Morphotactics and lexicons are changed to make morphological parser work in two ways. Therefore, from the output of the recognition mode, synthesizer can create the original word by only appending spaces between the morpheme boundaries. Following is an example to this situation.

I. Input to the recognizer mode of TMA:

kitabım (My book)

Output from the recognizer mode of TMA:

kitap+Noun+A3sg+P1sg+Nom

II. Input to the synthesizer mode of TMA:

kitap +Noun +A3sg +P1sg +Nom

Output from the synthesizer mode of TMA:

kitabım

- New root words are added to the lexicons of nouns, proper nouns, postpositions, verbs, and conjunctions.

I. **Word:** Annenler(your mother mainly and her husband)

Output: annenler+Noun+A3pl+P2sg+Nom

II. **Word:** Dahi(even)

Output: dahi+Conj

III. **Word:** hele(just)

Output: hele+Conj

- POS tags of some lexicons are changed.
- Necessary suffixes are added to morphotactics. Following are examples to these derivation morphemes and the words with them. Related morphemes and features are in bold case and examples are self-explanatory.

I. **Word:** rötuşla(to retouch)

Morpheme Boundaries: rötuş+**IA** (rötuş=retouching)

Output:

rötuş+Noun+A3sg+Pnon+Nom[^]**DB+Verb+Acquire**+Pos+Imp+A2sg

II. **Word:** rötuşlan(to become retouched)

Morpheme Boundaries: rötuş+**IAn** (rötuş=retouching)

Output:

rötuş+Noun+A3sg+Pnon+Nom[^]**DB+Verb+Acquire**+Pos+Imp+A2sg

- Output format of the analyzer is reorganized and modified to produce output in a uniform and standard way.

I. **Word:** yedikten(after from eating)

Morpheme Boundaries: ye+dik+ten

Output before the modification:

((CAT*V)(R* "ye")(CONV* ADJ "dik")(CASE* ABL))

Output After the modification:

ye+Verb+Pos^DB+Adj+PastPart^DB+Noun+Zero+A3sg+Pnon+Abl

As it can be seen, we print the full inflectional feature structures of all root, intermediate, and last derived forms of words in any case. For verbs, we output the polarity and for nouns, we output the singular/plural, person, and case information. In this way, we standardize the input into our parser.

- Adjective modifier adverbs are subcategorized.

I. Daha (more), daha+Adverb+AdjMdfy

II. En (most), en+Adverb+AdjMdfy

III. Derhal (immediately), derhal+Adverb

- Rules for the question morpheme “-mi” to take tense and agreement markers are added.

I. miyim, mi+yim,mi+Ques+Pres+A1sg

II. miydin, mi+ydi+n,mi+Ques+Past+A2sg

- Numbers lexicon and their morphotactics are rewritten completely to handle joint number sequences and their inflected and derived forms.

I. Yirmibeş(25), yirmi(20)+beş(5), yirmibeş+Num+Card

- II. Yüzdoksansekiz(198), yüz(100)+doksan(90)+sekiz(8), yüzdoksansekiz+Num+Card
- III. Altıncı (sixth), altı(6)+ıncı(th), altı+Num+Ord
- IV. ikişer (two at a time), iki(2)+şer(th), iki+Num+Dist

4.2 System Architecture

The aim of this work is to develop a syntactic grammar of Turkish in the link grammar formalism. Partially unlexicalized output of a morphological analyzer after some preprocessing is used as input to the grammar. If the morphological analyzer cannot parse a word, it might not be a valid Turkish word or it might be an unknown word. As mentioned in Section 2.5, link grammar parser provides some functionality to handle unknown words. So, these words are used as input to the parser as they are and necessary rules for these unknown words are added to the grammar. Hence, our current system handles unknown words. Currently, the grammar cannot handle punctuation symbols, but they can easily be integrated. The parser uses only morphological and syntactic information, so it makes use of no semantic information.

System architecture is depicted in Figure 4 as a flowchart by labeling the important steps 1 through 5. Our program is developed in C, and it uses the morphological analyzer and the link grammar static libraries externally. The borderlines of these two external processes, morphological analyzer and link grammar, are drawn in bold to distinguish them from the internal parts of the system.

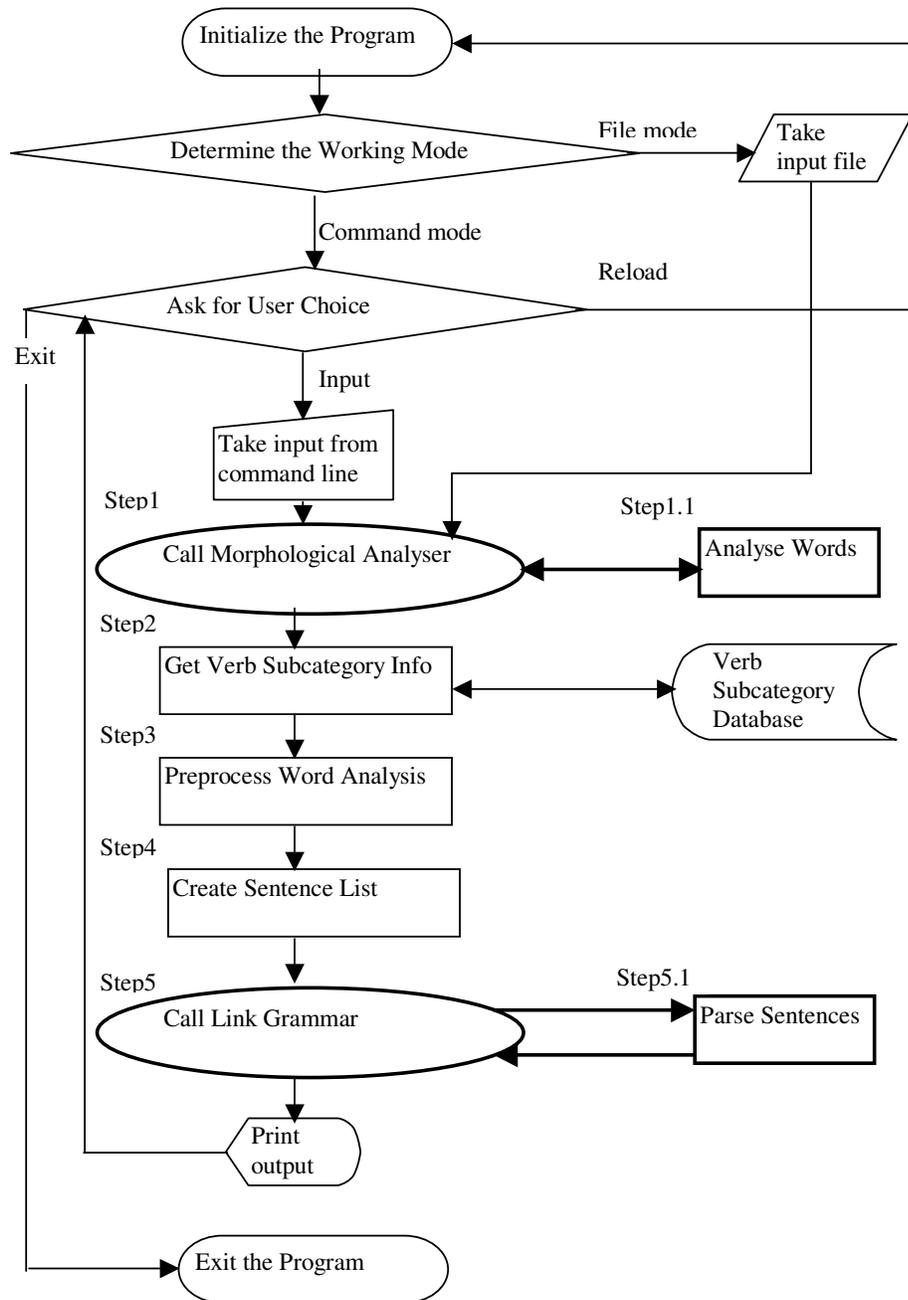


Figure 4 System Architecture

At the beginning, our system initializes by loading the lexicons and morphological rules of the morphological analyzer and grammar rules of the link grammar. All these rules and data are kept in memory until either the

program exits or the user types the reload command. If the system is started in the input file mode by specifying both the mode and input/output files in the configuration file, it parses the file and turns into command mode again. In file mode, the system expects a sentence on each line of the file. In this mode, the system expects input from the user. At this stage, user can also reload all the lexicons and rules into memory to reflect the last changes to the sentences. In addition, it is possible to stop the program by typing exit command.

Step 1: Morphological Analysis of Words in the Sentence

After taking the input sentence, in step 1, our system calls the external morphological analyzer for each word of the sentence to get their morphological analysis. The feature set of the morphological analyzer used in our system is listed in APPENDIX A. The word itself is used in the rest of the system if the morphological analyzer cannot analyze a word.

Input to Step1: sen kitabı okudun (you read the book)

Output from Step 1:

- I. Sen+Pron+A2sg+Pnon+Nom (you)

- I. Kitap+Noun+A3sg+Pnon+Acc (the book)
- II. Kitap+Noun+A3sg+P3sg+Nom

- I. Oku+Verb+Pos+Past+A2sg (read)

Step 1: Getting Verb Subcategory Information

Then in Step 2, subcategory information for the morphological analysis of verbs is loaded from an external lexicon. In this external lexicon, verbs are categorized according to their object requirements. This external verb subcategory information lexicon has nothing to do with the morphological analyzer and its lexicons. Verbs can take nouns as objects. In Turkish, the case of a noun that can be taken by a verb as its object can be locative, ablative, dative, accusative

(nominative), or instrumental. In addition, a verb can be used without taking an object. This information about the types of verbs according to their object is encoded as a six digit binary number in our system, each digit representing a specific case for the nominal object that can be taken by a verb. If the verb can take a nominal object by a specific case, then related digit in the binary number is set to one, else it is set to zero. Numbering digits from right to left, the meaning of each digit in the subcategory information structure is shown below.

–	–	–	–	–	–
6	5	4	3	2	1
Locative (-de) ev+de (at home)	Ablative (-den) ev+den (from home)	Dative (-e) ev+e (to home)	Accusative+Nominative (-i) ev+i (the home)	Instrumental (-le) ev+le (with home)	Objectless

Table 2 Verb Subcategorization Information

For example for the word “anlaşmak” (to come to an agreement), “100011” exists in our external verb subcategory lexicon. Since the first (objectless), the second (instrumental) and sixth digits are equal to one, this verb can be used without an object, or it can take a nominal object with an instrumental or locative case. If subcategory information is found, it is appended to the end of the morphological analysis of the word.

Input to step 2:

I. anlaş+Verb+Pos+Imp+A2sg

Output from step 2:

I. anlaş+Verb+Pos+Imp+A2sg+100011

If a verb cannot be found in the external verb subcategory lexicon, meaning that it is subcategory information is not available, nothing appended to its end, and these kinds of verbs are assumed to take objects in any case.

Step 3: Stripping Lexical Parts of Words

In step 3, the output of step 2 is preprocessed for the parsing stage. In this step for all types of words except conjunctions, lexical parts of the words are removed. In fact, our link grammar for Turkish is designed for the classes of word types and their feature structures, i.e. POS, rather than the words themselves.

Input to step 3:

- I. Sen+Pron+A2sg+Pnon+Nom (you)

- I. Kitap+Noun+A3sg+Pnon+Acc (the book)
- II. Kitap+Noun+A3sg+P3sg+Nom

- I. Oku+Verb+Pos+Past+A2sg (read)

Intermediate output from Step 3:

- I. Pron+A2sg+Pnon+Nom

- I. Noun+A3sg+Pnon+Acc
- II. Noun+A3sg+P3sg+Nom

- I. Verb+Pos+Past+A2sg

The intermediate output of step 3, as shown above, is the list of unlexicalized morphological feature structures of words. If a word is derived from another word by the help of at least one derivational suffix, then its feature structure is said to contain derivational boundary. For example,

“Arabamızdaki nin”(of the one that is in our car) is a word derived from “Arabamızda”(in our car) with the help of suffixes “-ki” and “-nin”.

Araba (car)

Word	Feature Structure (Morphological Analysis)	Meaning in English
Araba+mız+da	Araba+Noun+A3sg+P1pl+Loc	In our car
Araba+mız+da+ ki+nin	Araba+Noun+A3sg+P1pl+Loc ^DB+Adj+Rel ^DB+Noun+Zero+A3sg+Pnon+Gen	Of the one that is in our car

So, “**^DB+Adj+Rel**” and “**^DB+Noun**” are the two derivational boundaries in the feature structure of this word and the intermediate output from step 3 for this word is:

Noun+A3sg+P1pl+Loc**^DB+Adj+Rel****^DB+Noun**+Zero+A3sg+Pnon+Gen

Feature structures of words with derivational boundaries are handled in a special way in our system. Details of this preprocessing applied to derived words in Step 3 are given in Figure 5.

To make the steps better understood an example is given in Figure 6 for each of them. *1* and *2* are two different example words to illustrate the both cases and items related to first example are marked with “Ex1” and items related to second example are marked with “Ex2”.

After this special preprocessing step to derived words, step 3 is completed.

- 1 *If the feature structure of input word has no derivational boundary*
 - 1.1 *Output is equal to input*
- 2 *Else*
 - 2.1 *Replace the derivation special subcategory information with space characters*
 - 2.2 *Preserve the last derivation with its inflectional features to the end*
 - 2.3 *Replace the inflectional features of the intermediate forms with space characters*
 - 2.4 *If there is an intermediate form with a POS the same as with the POS of the last one*
 - 2.4.1 *Replace them with space characters*
 - 2.5 *If the POS of the root form same as with the POS of the last one*
 - 2.5.1 *Replace the root with space characters*
 - 2.6 *If there are more than one intermediate derivation with the same POS,*
 - 2.6.1 *Replace them with space characters except the last one*
 - 2.7 *Append the string “Root” to end of the POS of the root derivation*
 - 2.8 *Append the string “DB” to end of the POS of the intermediate derivation*
 - 2.9 *Preserve the last derivation with its inflectional features as it is*

Figure 5 Special Preprocessing for Derived Words

- 1 ***Noun+A3sg+Pnon+Acc*** (Ex1)
 - 1.1 *Noun+A3sg+Pnon+Acc(output)* (Ex1)
- 2 ***Noun+A3sg+PIpl+Loc^DB+Adj^Rel^DB+Noun+Zero+A3sg+Pnon+Gen*** (Ex2)
- 3 *Noun+A3sg+PIpl+Loc^DB+Adj^DB+Noun+A3sg+Pnon+Gen* (Ex2)
- 4 *Noun+A3sg+Pnon+Gen (preserve this)*(Ex2)
 - 4.1 *Noun Adj Noun+A3sg+Pnon+Gen* (Ex2)
 - 4.2 *Noun Adj Noun+A3sg+Pnon+Gen (intermediate form is Adj, not a noun)* (Ex2)
 - 4.2.1 *Noun Adj Noun+A3sg+Pnon+Gen (so nothing done)* (Ex2)
 - 4.3 *Noun Adj Noun+A3sg+Pnon+Gen (yes, first and last pos are the same, noun)* (Ex2)
 - 4.3.1 *Adj Noun+A3sg+Pnon+Gen (remove the root)* (Ex2)
 - 4.4 *Adj Noun+A3sg+Pnon+Gen (no, there is just intermediate form, Adj)* (Ex2)
 - 4.4.1 *Adj Noun+A3sg+Pnon+Gen (so nothing done)* (Ex2)
 - 4.5 *Adj Noun+A3sg+Pnon+Gen (root is removed already, so nothing done)* (Ex2)
 - 4.6 *AdjDB Noun+A3sg+Pnon+Gen* (Ex2)
 - 4.7 *AdjDB Noun+A3sg+Pnon+Gen* (Ex2)

Figure 6 Example to Preprocessing for Derived Words

Step 4: Creating Sentences for Link Parser

Since a part-of-speech tagger is not used in our system, the number of feature structures found for the words are very large. For this reason, after this step, a separate sentence is created for each of the morphological parse combinations of the words in step 4. For the example sentence given in step 3, “sen kitabı okudun” (you read the book), the output of step 4 is shown below.

Input to Step 4:

- I. Pron+A2sg+Pnon+Nom

- I. Noun+A3sg+Pnon+Acc
- II. Noun+A3sg+P3sg+Nom

- I. Verb+Pos+Past+A2sg

Output from Step 4:

- I. Pron+A2sg+Pnon+Nom Noun+A3sg+Pnon+Acc Verb+Pos+Past+A2sg
- II. Pron+A2sg+Pnon+Nom Noun+A3sg+P3sg+Nom Verb+Pos+Past+A2sg

Step 5: Parsing the Sentences

At the end, for each of these sentences, link grammar is called, and each of the sentences is parsed.

Input to Step 5:

- I. Pron+A2sg+Pnon+Nom Noun+A3sg+Pnon+Acc Verb+Pos+Past+A2sg
- II. Pron+A2sg+Pnon+Nom Noun+A3sg+P3sg+Nom Verb+Pos+Past+A2sg

Output from Step5:

sen kitabı okudun
1.1)

```

+-----Wvss-----+
|               +-----Sss-----+
|               |               +-----Oc-----+
|               |               |               |
LEFT-WALL Pron+A2sg+Pnon+Nom Noun+A3sg+Pnon+Acc Verb+Pos+Past+A2sg

```

cost vector=(UNUSED=0 DIS=0 AND=0 LEN=3)

1^^

sen+Pron+A2sg+Pnon+Nom kitap+Noun+A3sg+Pnon+Acc oku+Verb+Pos+Past+A2sg

2.1)

```

+-----Wvss-----+
|               +-----Sss-----+
|               |               +-----On-----+
|               |               |               |
LEFT-WALL Pron+A2sg+Pnon+Nom Noun+A3sg+P3sg+Nom Verb+Pos+Past+A2sg

```

cost vector=(UNUSED=0 DIS=0 AND=0 LEN=3)

2^^

sen+Pron+A2sg+Pnon+Nom kitap+Noun+A3sg+P3sg+Nom oku+Verb+Pos+Past+A2sg

Chapter 5

Turkish Link Grammar

As explained in the previous sections, Turkish is head-final, hence in a regular Turkish sentence, modifiers of a word are always on the left hand side, and the word it modifies is on the right hand side. For this reason, left-linking requirements of a word corresponds to its modifiers and right-linking requirements corresponds to the word it modifies in Turkish. Let us consider the following example sentence:

1. Sentence

Küçük	top	düştü (The small ball fell down)
Small	ball	fell down

Related Linkage:

+-----A-----+	-----S-----+	
küçük	top	düştü
small	ball	fell down

In this example, the noun “top” (ball) is modified by the adjective “küçük” (small) on the left hand side, hence for the noun “top”(ball) to connect to an adjective is one of its left-linking requirements. On the other hand, the same noun modifies the verb “düştü”(fell down) as its subject on the right hand side. For this reason, to connect to a verb as a subject is one of its right-linking requirements.

Another important observation about the syntax is that although any word can be modified by more than one word (resulting in many conjoined left-linking requirements), each word can modify at most just one word (resulting in disjointed right-linking requirements). Following is an example to this situation:

2. Utterance

Küçük	kırmızı	top	(The small red ball)
Small	red	ball	

Related Linkage:

+-----A-----+		
	+-----A-----+	
küçük	kırmızı	top
Small	red	ball

In this example, the noun “top”(ball) is modified by two adjectives “küçük”(small) and “kırmızı”(red) on its left. This rule is broken only if there exist a set of headwords, i.e. modified by the same set of modifiers, that are connected by a number of punctuation symbols or conjunctions.

In addition, if a word, “L₁“ modifies a word “R₁“ on the right hand side and if there is another modifier word between these two words, say “L₂“, “L₂“ can modify only one of the words between “L₂“ and “R₁“. In fact, these last two observations are not specific to Turkish syntax and named as “planarity property” in computational linguistics. This property is one of the general properties of languages that the link grammar formalism is based on, namely the “planarity rule”.

In the light of these observations, the details of the Turkish Link Grammar (TLG) are explained in the following sections.

5.1 Scope of Turkish Link Grammar

The link grammar developed in this thesis includes most of the rules in Turkish syntax. Noun phrases; postpositional phrases; dependent clauses constructed by gerunds, participles, and infinitives; simple, complex, conditional, and ordered/compound sentences; nominal and verbal sentences; regular sentences; positive, negative, imperative, and interrogative sentences; pronoun drop; freely changing order of adverbial phrases, noun phrases acting as objects, and subject are in our scope. In addition, we can handle quotations, numbers, abbreviations, hyphenated expressions, and unknown words.

However, we do not handle inverted sentences, idiomatic and multi-word expressions, punctuation symbols, and embedded and some types of substantival sentences.

5.2 Linking Requirements Related to All Words

As explained in Section 4.2, in order to preserve the syntactic roles that the intermediate derived forms of a word play, we treat them as separate words in the grammar. On the other hand, to show that they are the intermediate derivations of the same word, all of them are linked with the special “DB”¹ connector. In the following example, the feature structure of each morpheme is marked with the same subscript.

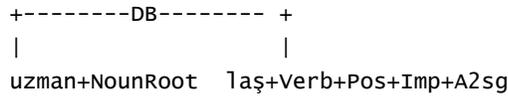
3. Word: uzman₁+laş₂ (specialize)

Full feature structure:

uzman+Noun+A3sg+Pnon+Nom₁ ^DB+Verb+Pos+Imp+A2sg₂

¹ DB used to denote derivation boundary

Linked structure¹:



Here, the noun root “uzman”(specialist) is an intermediate derived form and connected to the last derivation morpheme “-1aş” (to become) by the “DB” link, to denote that they are parts of the same word.

However, these intermediate derived forms, IDF, do not contribute to the right linking requirement of the last derived word. In addition, the “DB” linking requirements of the intermediate derived forms is different according to their order. The first form, which is the root word, intermediate forms placed between the first and the last forms, and the last derived form has different “DB” linking requirements.

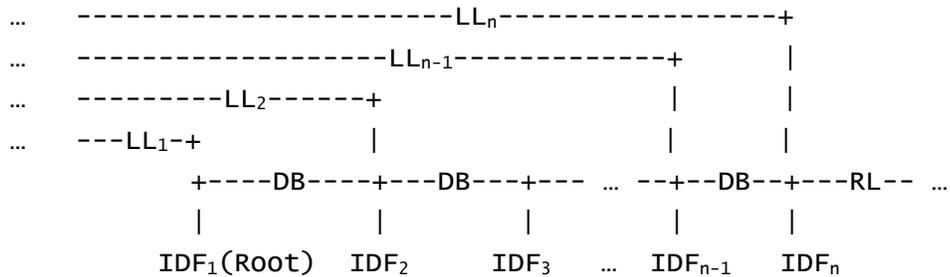


Figure 7 Linking Requirements of Intermediate Forms of a Word, W_x

In Figure 7, linking requirements of a word, “W_x“, with n intermediate derived forms (IDF₁...IDF_n) are illustrated. In Figure 7, “LL“ represents the links to the words on the left hand side of “W_x“, and “RL“ represents the links to the words on the right hand side of “W_x“. IDFs of the word “W_x“ are connected by “DB” links. As it can be seen all n IDFs can connect to the words to the left of “W_x“, i.e. “LL”, but only the last IDF, IDF_n can connect to the words on the

¹ Although we use lexical parts (like “uzman” in “uzmanlaşmak”) in our examples, the lexical parts are not used in actual implementation, i.e. “uzman” as “uzman+NounRoot”, “1aş” as “Verb+Pos+Imp+A2sg”.

right hand side of “ W_x “, i.e. “RL”. In addition, IDF_1 , which is the root stem, needs only to connect to its right with the “DB” connector, whereas the last IDF, IDF_n needs to connect to its left with the same connector. On the other hand, all the IDFs between these two should connect to both to their lefts and to rights with “DB” links to denote that they belong to the same word, “ W_x “. Hence, the same word, in fact the same IDF, has different linking requirements depending on its place in a word. To handle this situation, different items are placed into the grammar representing each of these three places of the same word¹.

The term “derivational linking requirements”, DLR, refers to linking requirements related to “DB” connectors, and “non-derivational linking requirements”, NDLR, refers to the ones that does not related to “DB” connectors, hereafter. In addition, NDRL is used as an abbreviation for “non derivational left linking requirement” and NDRLR is for “non derivational right linking requirement”. In Figure 8, derivational linking requirements are in italics and non-derivational linking requirements are in bold.

//linking requirements of the “intermediate derived form in the beginning”, IDF_{Root} IDF_{Root} : NDLLR & <i>DB+</i> ; //linking requirements of the same “intermediate derived form in the middle”, IDF_{DB} IDF_{DB} : <i>DB-</i> & NDLLR & <i>DB+</i> ; //linking requirements of the same “intermediate derived form at the end”, IDF IDF : <i>DB-</i> & NDLLR & NDRLR ;

Figure 8 Change of Linking Requirements of an IDF According to Its Place

As it can be seen in Figure 8, NDLR’s of an IDF placed at the beginning and in the middle are the same. In addition, NDLR of the IDF for these two positions is a subset of the whole NDRL of the same IDF placed at the end, to be precise, it is equal to NDLLR of it. For this reason, from this point on, we give only

¹ Please remember each intermediate derived form is handled as a separate word in TLG.

NDLR of the words, IDFs placed at the end. However, they are placed as separate entries in the dictionary file of Turkish Link Grammar, TLG. Because of this derivational structure, we do not do anything special either to gerunds, participles, and infinitives, etc.

In addition, as explained in Section 3.2.3, all words can take the question morpheme, i.e. the type without any person or time suffix. We call this type of the question morpheme with only the question meaning as “*regular question morpheme*”, hereafter. Since all question morphemes are written separately in Turkish, the morphological analyzer cannot handle them. For this reason all word categories in the grammar have a right linking requirements to handle regular question morpheme. Linking requirements of all words to regular question morpheme is represented with the “QBr” connector. “QB” is the connector for all question morphemes and the subscript r is used to indicate it is a regular question morpheme, i.e. a question morpheme with no person or tense suffix. Some of the feature structures of words and links of the linkages in the following examples are not shown due to space limitations hereafter.

4. Utterance: Geldin mi? (Did you come?)

Linked structure:

```

+-----QBr-----+
|                   |
ge1+Verb+Pos+Past+A2sg   mi+Ques

```

5. Utterance: Elbise mi(Is it dress?)

Linked structure:

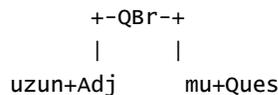
```

+-----QBr-----+
|                   |
elbise+Noun+A3sg+Pnon+Nom   mi+Ques

```

6. Utterance: Uzun mu(Is he/she tall?)

Linked structure:



Since both of these two phenomena, the question morpheme, and derivation boundary phenomena are common to all words we combined them in a macro, and used it in the linking requirements of all words. This macro, <affix-bound>, is given in Figure 9 in rule 1.

```

%rule 1
<affix-bound>:{DB-} & {QBr+};

%rule 2
Noun+A3sg+Pnon+Gen Noun+Prop+A3sg+Pnon+Gen
:(<affix-bound> & <noun-phrase-non> & <g-noun-right>)or
<Sffxlss-N-to-Vrb-Drv-non>;

```

Figure 9 Macro for the Derivation Boundary and Question Morpheme

Rule 1 says that, any last IDF or word can connect to another IDF on its left and can take a regular question morpheme on its right. Rule 2 is one of the rules from our TLG dictionary file showing usage of this macro.

Placing this macro at the beginning results in the word to which Noun+A3sg+Pnon+Gen or Noun+Prop+A3sg+Pnon+Gen is connected with the DB link to be the nearest word on the left hand side. This ensures that IDFs of the same word are all connected together. Similarly, it also ensures that if the word has a regular question morpheme, it should be the nearest linked word on the right hand side.

5.3 Compound Sentences, Nominal Sentences, and the Wall

In Section 3.4.1 , the structures of compound sentences in Turkish are explained. In TLG, we choose the predicates of independent clauses to represent the

clauses. Hence, to combine the independent clauses of a compound sentence, we connect their predicates to the conjunctions.

7. Sentence:

Sen gittin ve Ayşe koştu. (You went and Ayşe ran)
 You went and Ayşe ran.

Related Linkage:

	+-----CLV-----+	+-----CRV-----+	
Sen	git+Verb+Pos+Past+A2sg	ve	Ayşe koş+Verb+Pos+Past+A3sg
You	went	and	Ayşe ran

In this example, the feature structures of only predicates of clauses, i.e. in this case they are verbs, are shown. This sentence is a compound sentence consisting of two ICs, “Sen gittin”(You went) and “Ayşe koştu”(Ayşe ran). These two independent clauses are connected by the conjunction “ve”(and) and in TLG; we combine these two clauses by connecting their predicates “gittin”(went) and “koştu”(ran) to the conjunction “ve” (and) by “CLv” and “CRv” links. “CL” is used to connect a conjunction to the word on the left, and “CR” is used to connect to the word on the right. “v” subscript in “CLv” and “CRv” shows that the links connects to words of type verb on both sides.

If a sentence is represented by a set of links, i.e. linkage, connecting the syntactically related words with each other, then we need a starting point in the sentence to traverse all of its words. This is also necessary for marking the independent clause in a complex sentence with many dependent clauses. In addition, it makes us traverse a whole compound sentence, which is conjoined with some conjunctions or punctuation symbols, “without having to select one of the independent clauses to represent the whole sentence”. For this reason, we use the “LEFT-WALL”¹, i.e. the wall, and connect it either to the predicate of

¹It is explained in Section 2.5

the independent clause in a complex sentence or to the conjunction combining the predicates of two independent clauses.

LEFT-WALL: (Wv+ or Wc+) ;

Figure 10 Linking Requirements of the LEFT-WALL

As it can be seen in Figure 10, linking requirement of the wall is represented with the “W” connector. The subscript “v” is used for verbs and “c” for conjunctions. This formula ensures that the wall has to connect either to a verb with the “Wv” link or to a conjunction with the “Wc” link on the right hand side.

8. Sentence:

Sen gittin. (You went)
 You went.

Related Linkage:

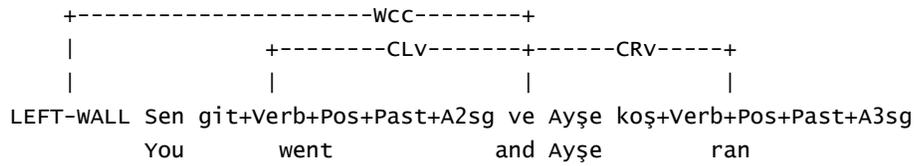
+-----	-WVSS-	-----+
LEFT-WALL	sen+Pron+A2sg+Pnon+Nom	git+Verb+Pos+Past+A2sg
	You	went

In this example, a simple sentence is given. Predicate of this sentence, which is a verb, is connected to the wall. The connector used in linking is “Wvss”. The subscript sequence “vss” shows that the connected word is a verb with second singular person suffix, i.e. “A2sg”.

9. Sentence:

Sen gittin ve Ayşe koştu. (You went and Ayşe ran)
 You went and Ayşe ran.

Related Linkage:



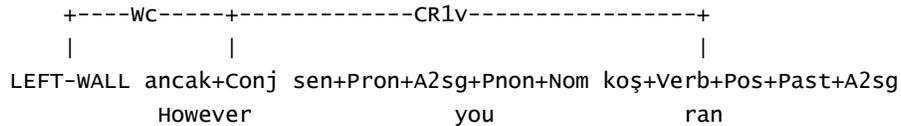
In 9, a compound sentence is given as an example. In this compound sentence, the conjunction “ve” (and) is connected to the wall with the “Wcc” link. The subscript sequence “cc” shows that the wall is connected to a conjunction in the middle. Since we choose predicates to represent the sentences, conjunctions are connected to them. However, in Turkish, some of the conjunctions can also be at the beginning of the sentence:

10. Sentence:

Ancak sen koştun. (However, you ran.)

However, you ran.

Related Linkage:



If there is a conjunction at the beginning of a sentence, then it is connected to the wall with the “Wc” link on the left hand side, and to the predicate with another link¹ on the right hand side. 10 is given as an example to this case.

Nominal sentences with omitted copula are frequently encountered in Turkish as mentioned previously in Section 3.4.2 . For this reason, we added macros to handle *suffixless nominal to verbal derivations* for nouns, pronouns, adjectives, and adverbs to the dictionary of TLG. Then, these macros are disjoined with other rules of the related word categories to ensure that the word

¹ Type of this connector depends on the conjunction type.

plays the role of either a nominal; i.e. an adjective, pronoun, adverb or a noun; or a predicate:

```
%rule 1 for adjectives
Adj:
(<affix-bound> & (({EA-} & <adj-right>) or ([<n-noun-
right>]))) or <Sffx1ss-Adj-to-Verb-Drv>;

%rule 2 Suffixless Adjective to Verb Derivation (Omitted
Copula)
<Sffx1ss-Adj-to-Verb-Drv> :
(<affix-bound>&{EA-}1&({@E-}1 & (St-&{@E-}))) or (Qbc+&{EA-});
```

Figure 11 Rules for Adjectives

In Figure 11, rule 1, the sub-formula in bold case represents the syntactic roles of adjective as an adjective only and it is explained in the related heading. However, what is important here is that it is disjoined with the macro <Sffx1ss-Adj-to-Verb-Drv> to enforce the adjectives to behave either as an adjective or as a predicate (verb).

Like any other derivation, these nominal words preserve their syntactic roles as a modified, i.e. they transfer their left linking requirements to the resulting predicate, while accepting the left and right linking requirements of their new role as a verb.

Rule 2 in Figure 11 says that an adjective can behave like a verb without taking any suffix or by taking the *question morpheme with copula* as described in Section 3.2.3. The first macro “<affix-bound>” is explained in Section 5.2. If the adjective is derived from another word, and if there is an adverb modifying it, it precedes the derived intermediate forms of the adjective on the left hand side. This linking requirement comes from the fact that the word is in fact an adjective. On the other hand, since it is a verb now, it can be modified on the left hand side by any number of adverbial phrases, preceded by a subject and then again by any number of adverbial phrases as the leftmost modifier. For example, Figure 12 is a representative sentence structure allowed by this rule. Optional

constituents are in italics and constituents those can be more than one are underlined.

<p>Adverb_v= Adverb that modifies verbs, Adverb_a= Adverb that modifies adjectives, IDF=Intermediate derived forms of adverb, Ques= Question Morpheme</p> <p><u>Adverb_v</u> Subject <u>Adverb_v</u> <u>Adverb_a</u> <u>IDF</u> Adjective <u>Ques</u></p>
--

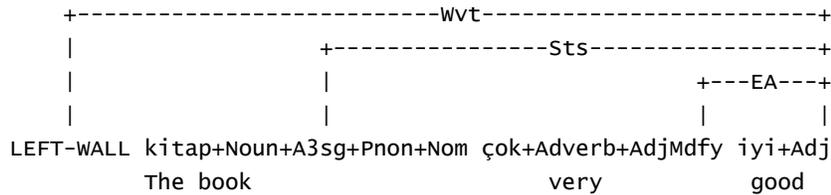
Figure 12 Suffixless Adjective to Verb Derivation, an Example Illustrative Sentence Structure

11. Sentence:

Kitap çok iyi. (The book is very good)

The book very good

Related Linkage:

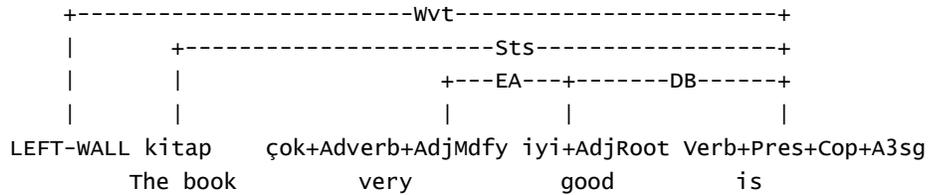


Sentence given in 11 is an example of nominal sentence with omitted copula. Here the adjective “iyi” (good) is the predicate of the sentence and hence connected to the wall with the “Wvt” link. In addition, it is modified by the adverb “çok”(very) as an adjective and it takes the subject “kitab”(the book) as a verb on the left hand side. In the following example, same sentence with the copula is presented to compare it with the previous version of the sentence. The copula suffix is in bold case and it does the exactly the same job of the “to be” verb in English.

12. Sentence:

Kitap çok iyi+**dir**. (The book is very good)
 The book very good **is**

Related Linkage:



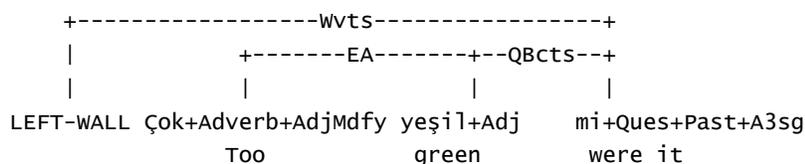
In this example, the adjective “iyi” (good) takes the copula suffix “dir”(to be), and hence it derives into a verb. This derivation is shown by the link “DB” between the adjective “iyi”(good) and the suffix “dir”(to be). When compared to the previous example, we can see that there is an overt morpheme to play the role of the predicate of this sentence and hence “Wvt” link is connected to copula suffix. In addition, unlike to previous example, the subject “kitap”(the book) modifies this new overt verb.

On the right side of the rule, there is another sub-formula, “(QBc+&{EA-})”, disjoined with the rest of the formula. Here, “QBc+” connector is used to link the adjective to the *question morpheme with copula*. Since such an adjective can be modified by an adverb on the left, the connector “{EA-}” is conjoined with the question morpheme connector. If the adjective takes this type of the question morpheme, not the adjective itself but the question morpheme gets the syntactic role of the verb. The following is a sentence accepted by this part of the rule.

13. Sentence:

Çok yeşil miydi? (Was it too green)
 Too, green was it.

Related Linkage:



In example 13, since the question morpheme is the predicate, the wall is connected to it with the “wvts“ link. In addition, the adverb “çok”(too) modifies the adjective “yeşil”(green); hence, they are connected with “EA“ link. The connector “QBctS“ shows that the adjective is connected to a question morpheme affixed by copula and third singular person suffixes.

5.4 Linking Requirements of Word Classes

After giving the material related to all word classes and explaining how we handle different sentence structures, now we explain the linking requirements of each of word categories one by one.

5.4.1 Adverbs

There are mainly three types of adverbs in our TLG dictionary. These are regular adverbs, question adverbs, and adjective modifier adverbs.

- I. Examples to Regular Adverbs: birdenbire (suddenly), akşamleyin (in the evening), evet (yes), hayır (no), sabah (in the morning), asla (never)
- II. Examples to Question Adverbs: Acaba (I wonder), nasıl (how), neden (why),
- III. Examples to Adjective Modifier Adverbs: daha (more), en (most), çok (very), koyu (dark), gayet (extremely)

In Turkish, instead of suffixation, comparative and superlatives are created with the help of adjective modifier adverbs. "Daha güzel" (more beautiful) is an example to comparatives and "en akıllı" (most intelligent) is to superlatives. Linking requirements of all these three types of adverbs is given in Figure 13.

```

Adverb : (<affix-bound> & ({EE-} & (EE+ or Ea+))) or <Sffxlss-
Adverb-to-Verb-Drv>;

Adverb+Ques : (<affix-bound> & ({EE-} & (EEq+ or Ea+))) or
<Sffxlss-Adverb-to-Verb-Drv>;

Adverb+AdjMdfy : ({EE-} & (EE+ or Ea+ or EA+)) or <Sffxlss-
Adverb-to-Verb-Drv>;

```

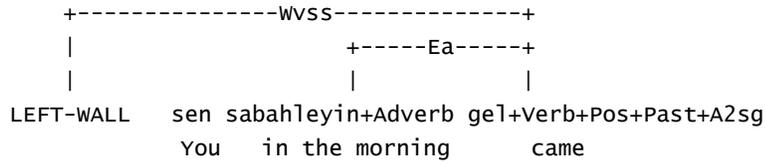
Figure 13 Linking Requirements of Adverbs

As it can be seen in Figure 13, the only the difference between the regular adverbs and the question adverbs is that connectors in the last one are subscripted with the "q" character to denote question type. In addition, regular adverb and adjective modifier adverb linking requirements are the same except that the latter has "EA+" connector to modify adjectives on the right hand side. "EE+" connector shows that adverbs can modify other adverbs on the right and "EE-" is its counterpart to enable an adverb to be modified by another one on the left. "E+" link is used to connect verbs to their modifier adverbial phrases, resulting in creating adverbial complements, and the subscript "a" shows that it is an adverb, i.e. not a postpositional phrase acting as an adverb. Example sentences to all these three adverb types are given below.

14. Sentence illustrating the usage of a regular adverb:

Sen sabahleyin geldin (You came in the morning)
You in the morning came.

Related Linkage:



15. Sentence illustrating the usage of an adjective modifier adverb:

Sen daha güzel bir elbise gördün. (You saw a more beautiful dress)

You more beautiful a dress saw.

Related Linkage:

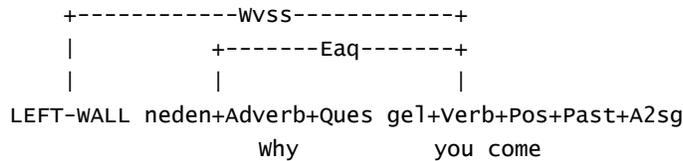


16. Sentence illustrating the usage of a question adverb:

Neden geldin? (Why did you come?)

Why you come

Related Linkage:



5.4.2 Postpositions

When nominal words in nominative, dative, ablative, or genitive forms are connected to postpositions on the right side, postpositional phrases are constructed and they behave like an adjective or adverb. For this reason we subcategorized postpositions according to the case of the nominal that they take on the left hand side.

```

%takes nouns in nominal case
Postp+PCNom: (Jn- & (Ap+ or (Ep+ or EEp+ or EAp+)));

%takes nouns in genitive case
Postp+PCGen: (Jg- & (Ap+ or (Ep+ or EEp+ or EAp+)));

%takes nouns in dative case
Postp+PCDat: (Jd- & (Ap+ or (Ep+ or EEp+ or EAp+)));

%takes nouns in ablative case
Postp+PC Abl: (Ja- & (Ap+ or (Ep+ or EEp+ or EAp+)));

%takes nouns in instrumental case
Postp+PCIns: (Ji- & (Ap+ or (Ep+ or EEp+ or EAp+)));

```

Figure 14 Linking Requirements of Postpositions

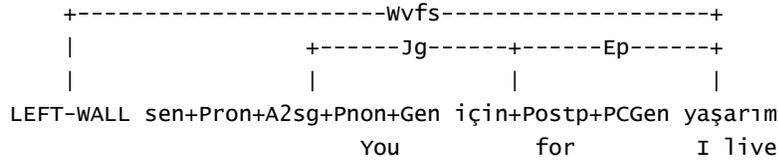
Hence, as it can be seen in Figure 14, there are five rules for each of these postpositions in TLG. The connector “Jn-“ shows that this postpositions need to connect to a noun in nominal case to create a postpositional phrase. The main link type “J“ used for postpositional phrases in general and it is subscripted with the initial letter of the case of the nominal that it connected to, i.e. “Jn“ for nominative, “Jg“ for genitive, “Ji“ for instrumental, “Ja“ for ablative, “Jd“ for dative. Then, the resulting postpositional phrase can play the syntactic role of an adjective by connecting to the following noun with the “Ap1“ link. Other connectors related to the adverbial usage are explained in Section 5.4.1. All the “p“ subscripts to the right of connectors denote the postposition. The word, “için” (for) is a postposition in Turkish and it can take nouns in either nominative or genitive case. The following are two example sentences illustrating these two usages of this word.

17. Sentence illustrating the usage of “için” (for) connected with nominal word in genitive case:

Senin için yaşarım (I live for you)
You for I live

¹ “p” subscript shows that the adjective is a postpositional adjective

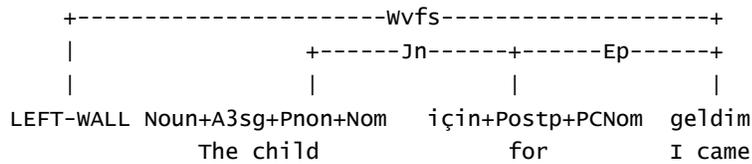
Related Linkage:



18. Sentence illustrating the usage of “için” (for) connected with nominal word in nominative case:

Çocuk için geldim (I came for the child)
The child for I came

Related Linkage:



5.4.3 Adjectives and Numbers

We explored adjectives in two groups, i.e. regular adjectives and question adjectives. The only difference between two is that connectors of the latter one are subscripted with the letter q to indicate that it is linked to a question adjective.

```

Adj :(<affix-bound> & (({EA-} & A+) or
([<n-noun-right>]))) or <Sffxlss-Adj-to-Verb-Drv>;

Num+Ord :(<affix-bound> & (({NN-} & {EA-} & A+) or
([<n-noun-right>]))) or <Sffxlss-Ord-to-Verb-Drv>;

Adj+Ques :(<affix-bound> & (({EA-} & Aq+) or
([<n-noun-right-q>]))) or <Sffxlss-Adj-to-Verb-Drv>;

```

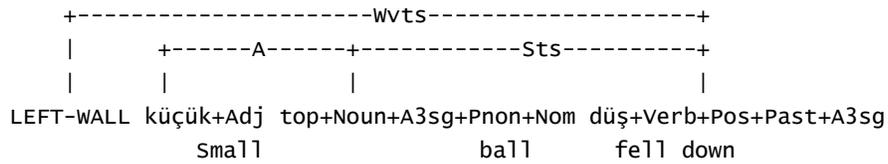
Figure 15 Linking Requirements of Adjectives

In Figure 15, the connector “EA-“ ensures the adjective to be modified by an adverb on its left. This connector is conjoined with the “A+“ connector to indicate that an adjective can modify a noun on the right hand side. In addition, these two connectors are disjoined with the “<n-noun-right>“ macro on the right hand side. In fact, this macro is for the syntactic roles of nouns as a modifier. Disjoinment of this macro to the existing formula of adjectives enables them to behave as nouns. Example sentences illustrating these usages are given below.

19. Sentence

Küçük top düştü (The small ball fell down)
 Small ball fell down

Related Linkage:

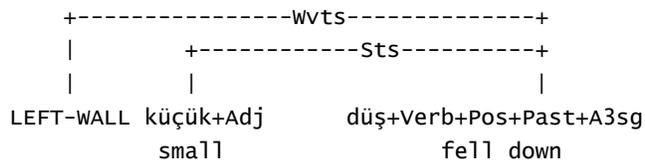


In this example, “top” (ball) is the subject of the verb and hence it is connected to the verb with “Sts” link. The subscript “ts” next to “S” connector show that the verb has third person singular suffix. Hence, the adjective “küçük” (small) modifies the subject noun “top” (ball).

20. Sentence

Küçük düştü (The small one fell down)
 Small fell down

Related Linkage:



In this sentence, the adjective “küçük” (small) is used with the meaning (the one which is small) and it is the subject of the verb.

```
Adj :(<affix-bound> & (({EA-} & A+) or
([<n-noun-right>]))) or <Sffxlss-Adj-to-Verb-Drv>;

Num+Ord :(<affix-bound> & ({NN-} & {EA-} & A+) or
([<n-noun-right>]))) or <Sffxlss-Ord-to-Verb-Drv>;

Num+Card: (<affix-bound> & ({NN-} & (Dn+ or NN+))) or <Sffxlss-
Card-to-Verb-Drv>;
```

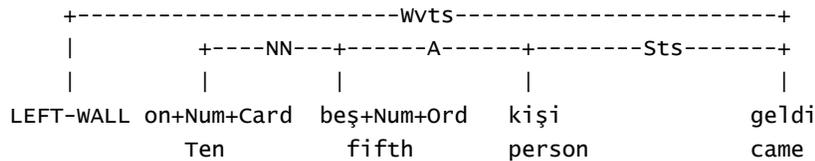
Figure 16 Linking Requirements of Numbers

Linking requirements of ordinal numbers are similar to linking requirements of adjectives. The only difference between the two is that ordinal numbers can take cardinal number words together in series on their left hand side and this situation is marked with the “NN-“ connector. Following example illustrates the usage of cardinal and ordinal number with nouns together.

21. Sentence

On beşinci kişi geldi (The fifteenth person came)
 Ten fifth person came

Related Linkage:

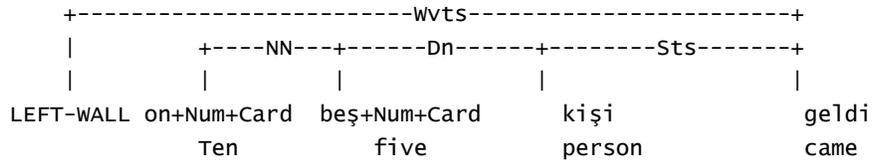


Similarly, “{NN-}” rule ensures that cardinal numbers can take other zero or more cardinal numbers on the left. On the right hand side, cardinal numbers link to either a noun with the “Dn” link modifying it as a determiner of type numeric or it can connect to another cardinal to create number series.

22. Sentence

On beş kişi geldi (Fifteen people came)
 Ten five person came

Related Linkage:



In this example, the numbers “on”(ten) and “beş”(five) came together to create the number series (fifteen) with the NN link. This series then link to the noun “kişi”(person) with the noun determiner link, i.e. “Dn”.

5.4.4 Pronouns

Similar to nouns, pronouns take case suffixes in Turkish and the syntactic role of a pronoun depends on the case suffix that it takes. For this reason, we have rules for pronouns in nominative, genitive, locative, ablative, accusative, dative, and instrumental cases separately.

```
%nominative pronouns can be subject of the verb
Pron+A1sg+Pnon+Nom: (<affix-bound> & (Sfs+)) or
                    <Suffixless-Pron-to-Verb-Drv>;
Pron+A2sg+Pnon+Nom: (<affix-bound> & (Sss+)) or
                    <Suffixless-Pron-to-Verb-Drv>;
Pron+A3sg+Pnon+Nom: (<affix-bound> & (Sts+)) or
                    <Suffixless-Pron-to-Verb-Drv>;
Pron+A1pl+Pnon+Nom: (<affix-bound> & (Sfp+)) or
                    <Suffixless-Pron-to-Verb-Drv>;
Pron+A2pl+Pnon+Nom: (<affix-bound> & (Ssp+)) or
                    <Suffixless-Pron-to-Verb-Drv>;
Pron+A3pl+Pnon+Nom: (<affix-bound> & (St+)) or
                    <Suffixless-Pron-to-Verb-Drv>;
```

Figure 17 Linking Requirements of Nominative Pronouns

In Turkish, subject of a sentence has to be in nominative case. In addition, subject and verb agree in person. The only exception to this rule is that if the subject of a sentence is in third person plural form, then the verb can be in either

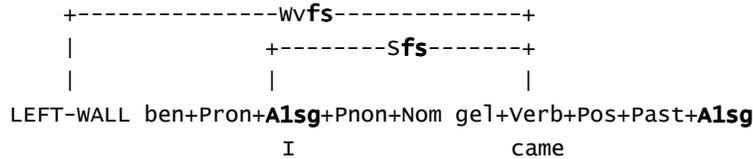
third person singular or third person plural form. Hence, nominative pronouns can be subject of the verb on the right hand side. On the other hand, pronouns do not take modifiers on their left hand side. Following are examples to person agreement between verb and subject of a sentence.

23. Sentence

Ben geldim (I came)

I came

Related Linkage:



In this example, the subject “ben”(I) is first singular person pronoun. Because of agreement rule, the verb “geldim”(came) have also first singular person suffix. Hence, the subject between them is subscripted with “**fs**”(first singular) to denote this agreement, like the link between the wall and the verb, i.e. “**Wvfs**”. A full list of subscripts for the subject link “**S**” is given in Table 3.

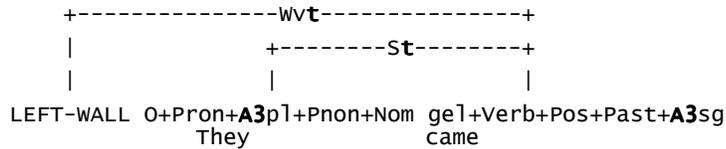
Sfs	First singular person subject
Sss	Second singular person subject
Sts	Third singular person subject
Sfp	First plural person subject
Ssp	Second plural person subject
Stp	Third plural person subject

Table 3 Subscript Set for S (Subject) Connector

24. Sentence illustrating the agreement between third person plural subject and third person singular suffixed verb.

Onlar geldi (They came)
 They came

Related Linkage:



% Linking Requirements of Genitive Pronouns: Rule Part 1

Pron+A1sg+Pnon+Gen: (<affix-bound> & (Dfs+ or Jg+)) or
 <Suffixless-Pron-to-Verb-Drv>;

Pron+A2sg+Pnon+Gen: (<affix-bound> & (Dss+ or Jg+)) or
 <Suffixless-Pron-to-Verb-Drv>;

Pron+A3sg+Pnon+Gen: (<affix-bound> & (Dts+ or Jg+)) or
 <Suffixless-Pron-to-Verb-Drv>;

Pron+A1pl+Pnon+Gen: (<affix-bound> & (Dfp+ or Jg+)) or
 <Suffixless-Pron-to-Verb-Drv>;

Pron+A2pl+Pnon+Gen: (<affix-bound> & (Dsp+ or Jg+)) or
 <Suffixless-Pron-to-Verb-Drv>;

Pron+A3pl+Pnon+Gen: (<affix-bound> & (Dtp+ or Jg+)) or
 <Suffixless-Pron-to-Verb-Drv>;

% Linking Requirements of Accusative Pronouns: Rule Part 2

Pron+A1sg+Pnon+Acc Pron+A2sg+Pnon+Acc Pron+A3sg+Pnon+Acc
 Pron+A1pl+Pnon+Acc Pron+A2pl+Pnon+Acc Pron+A3pl+Pnon+Acc:
 (<affix-bound> & {Oc+}) or <Suffixless-Pron-to-Verb-Drv>;

Figure 18 Linking Requirements of Genitive and Accusative Pronouns

In Figure 18, rule part 1, the connector “Jg” represents the postposition phenomena explained in Section 5.4.2. “D+” connects possessive pronouns (genitive pronouns) to nouns. Like agreement in person between subject and verb of a sentence, possessive pronoun and the noun that it modifies have to agree in person also.

25. Sentence illustrating the agreement between second person plural pronoun and second person plural suffixed noun.

Sizin kitabınız (your book)
 Your your book


```

Pron+A1sg+Pnon+Loc Pron+A2sg+Pnon+Loc Pron+A3sg+Pnon+Loc
Pron+A1pl+Pnon+Loc Pron+A2pl+Pnon+Loc Pron+A3pl+Pnon+Loc:
(<affix-bound> & {IOl+}) or <Suffixless-Pron-to-Verb-Drv>;

Pron+A1sg+Pnon+Abl Pron+A2sg+Pnon+Abl Pron+A3sg+Pnon+Abl
Pron+A1pl+Pnon+Abl Pron+A2pl+Pnon+Abl Pron+A3pl+Pnon+Abl:
(<affix-bound>&{IOa+ or Ja+}) or <Suffixless-Pron-to-Verb-Drv>;

Pron+A1sg+Pnon+Dat Pron+A2sg+Pnon+Dat Pron+A3sg+Pnon+Dat
Pron+A1pl+Pnon+Dat Pron+A2pl+Pnon+Dat Pron+A3pl+Pnon+Dat:
(<affix-bound>&{IOd+ or Jd+}) or <Suffixless-Pron-to-Verb-Drv>;

Pron+A1sg+Pnon+Ins Pron+A2sg+Pnon+Ins Pron+A3sg+Pnon+Ins
Pron+A1pl+Pnon+Ins Pron+A2pl+Pnon+Ins Pron+A3pl+Pnon+Ins:
(<affix-bound> & {Ep+}) or <Suffixless-Pron-to-Verb-Drv>;

```

Figure 19 Linking Requirements of Locative/Ablative/Dative/Instrumental Pronouns

5.4.5 Nouns

Nouns play the second important role in a sentence after verbs. They modify the verb either as a subject or as objects. In addition, they are involved in many adverbial phrases and postpositional phrases to create different types of compliments. Before moving to the syntactic roles of nouns as modifier, we give a brief description of how they are modified by other words on their left hand side.

5.4.5.1 Nominal Groups

Nominal groups consist of a group of nouns those are connected to each other with the possessive relation [14]. In nominal groups, a modified item can be modified by more than one noun. Both the modified item and modifiers themselves can be nominal groups, too. They can be classified into three groups:

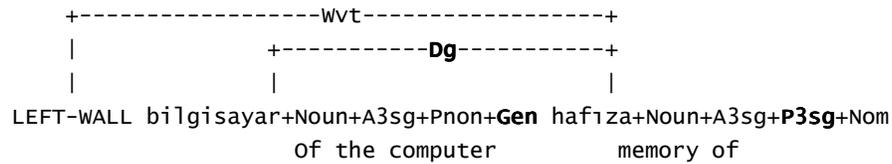
a) Definite (or Possessive) Nominal Groups: In these groups, the modifier takes genitive case suffix and the modified noun takes third person possessive suffix. We used “Dg” link to connect these words as it can be seen in the following example.

27. Sentence

Bilgisayarın hafızası (Memory of the computer)

Of the computer memory of

Related Linkage:



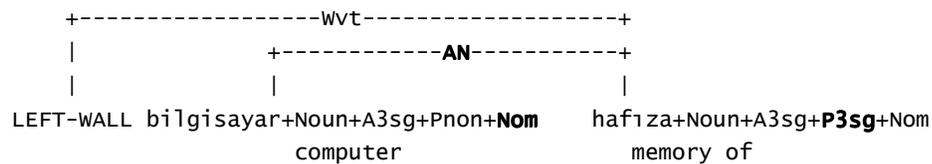
b) Indefinite (or Qualifying) Nominal Groups: In these groups, the modifier takes no case suffix, hence it is in nominative case, and the modified noun takes third person possessive suffix. We used “AN” link to connect these words as it can be seen in the following example.

28. Sentence

Bilgisayar hafızası (Computer memory)

Computer memory

Related Linkage:

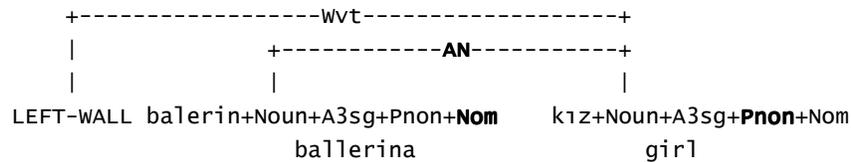


c) Adjectival Nominal Groups: In these groups, the modifier takes no case suffix, hence it is in nominative case, and the modified noun takes no possessive suffix, too. We used the same “AN” link to connect these words as it can be seen in the following example.

29. Sentence

Balerin k1z (Ballerina girl)
 Ballerina girl

Related Linkage:



5.4.5.2 Linking Requirements of Nouns

To sum up, nouns can be explored in two big categories according to their left linking requirements:

a) Nouns in possessive form

These are the nouns with one of the following possessive suffixes or features: “P1sg”, “P2sg”, “P3sg”, “P1pl”, “P2pl” and “P3pl”. These nouns in general take possessive pronouns on their left hand side: Example 25 illustrates this situation in Section 5.4.4 .

However, nouns with third person singular possessive suffixes have a different property that they can also take other nouns on the left hand side to create nominal groups as explained in the previous subsection. Example 27 illustrates this situation.

b) Nouns not in possessive form

These are the nouns with no possessive suffix, i.e. with “Pnon” feature. These nouns together with the nouns in possessive form have the following common left linking requirements.

```
<llr_noun>:{{@AN-}& ({{Dn-}&{@A-}} or {{@A-}&{Dn-}})};
```

Figure 20 Left Linking Requirements Common to All Nouns

Keeping the possibility of taking determiners explained up to this point, on the left most place, nouns can also take the modifiers shown in Figure 20. This rule says that, a noun can take zero or more number of nouns in nominative case on the left to create nominal groups. Following example illustrates this situation:

30. Sentence

Benim ortaokul son sınıf öğrencim.

My secondary school last year my student

(My student who is senior at secondary school)

Related Linkage:

```

+-----Dfs-----+
|                   +-----AN-----+-----AN-----+-----AN-----+
|                   |                   |                   |                   |
ben+A1sg+Pnon+Gen ortaokul+Nom son+Nom sınıf+Nom öğrenci+P1sg
My           secondary school last year my student

```

The sub-formula “{{Dn-}&{@A-}}” allow nouns to take cardinal numbers followed by zero more adjectives optionally. Also inverse of this formula is dijoined with it to enable vice versa. So, sentences in the following two examples have the same meaning and they are both valid.

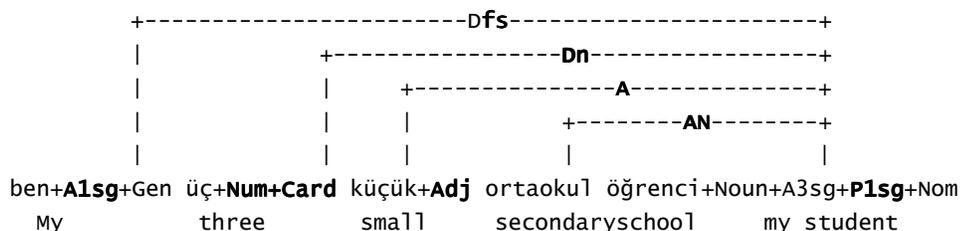
31. Sentence

Benim üç küçük ortaokul öğrencim.

My three junior secondary school my student

(My three junior secondary school students)

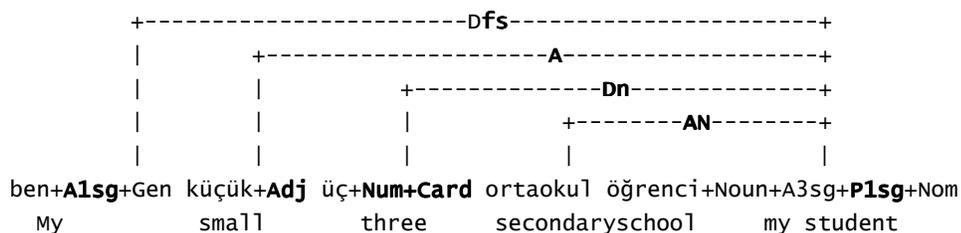
Related Linkage:



32. Sentence

Benim küçük üç ortaokul öğrencim.
My junior three secondary school my student
(My junior three secondary school students)

Related Linkage:



```

<genitive-noun-right>:(Dg+ or Jg+);
<dative-noun-right>:(IOd+ or Jd+);
<ablative-noun-right>:(IOa+ or Ja+);
<accusative-noun-right>:(Oc+);
<locative-noun-right>:(IOl+);
<instrumental-noun-right>:(Ei+);

<nominative-noun-right-A3sg>:(Sts+ or On+ or Jn+);
<nominative-noun-right-A3pl>:(St+ or On+ or Jn+);

<nominative-noun-right-A3sg-PnonP3sg>:
(Sts+ or On+ or Jn+ or [AN+]);

<nominative-noun-right-A3pl-PnonP3sg>:
(St+ or On+ or Jn+ or [AN+]);

```

Figure 21 Right Linking Requirements of Nouns

Right linking requirements of nouns in Figure 21 are same as pronouns as it is explained in Section 5.4.4 . The only difference is “[AN+]” link, which is explained in Section 5.4.5.1. However, since adjectival nominal groups are quite rare, we give it a cost of one by surrounding it one level of square brackets. The

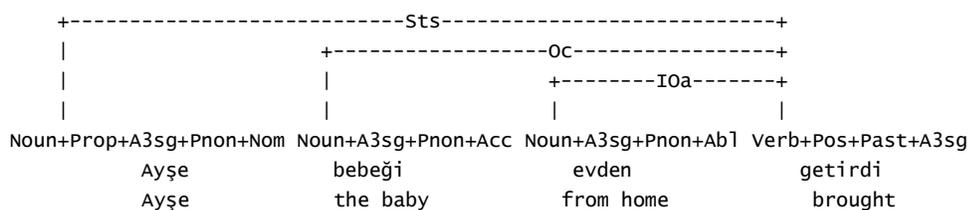
example given below shows many of the syntactic roles of the nouns as modifiers.

33. Sentence¹

(Ayşe brought the baby from home)

Ayşe bebeği evden getirdi.
 Ayşe the baby from home brought

Related Linkage:



5.4.6 Verbs

We explained that we get verb sub-categorization information for the case of their objects in Table 3 in Section 4.2. However, since some of the sub-categorization information for verbs can be wrong, we used it only to decide that whether the verb is intransitive or not. If the verb is not an intransitive one, we allow it to take at most one subject, one direct object, one indirect object in locative case, one indirect object in ablative case, one indirect object in dative case, and zero or more adverbial phrases on its left hand side. However, if the verb is an intransitive verb or if it is a copula, then it is allowed to take at most one subject and any number of adverbial phrases on the left. Hence, we subcategorize the verbs into two big classes according to their object requirements. To allow all combinations of subject, object, indirect objects, and adverbial phrases, we simply add all combinations of them into the TLG dictionary, which results in the rule to be very long. For this reason, we do not

¹ Please note that words are separated from their feature structures and written on the next line due to space limitations.

include the left linking requirements of verbs. Since, the following examples are self-explanatory and these usages are explained in the previous word classes, i.e. pronouns and nouns, no further descriptions are given for them.

34. Sentence illustrating the usage of copula:

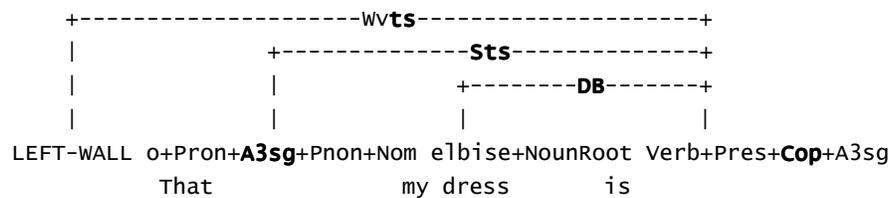
O elbisemdir. (That is my dress.)

That is my dress

Full feature structure:

o+Pron+A3sg+Pnon+Nom e1bise+Noun+A3sg+P1sg+Nom^DB+Verb+Zero+Pres+Cop+A3sg

Related Linkage:

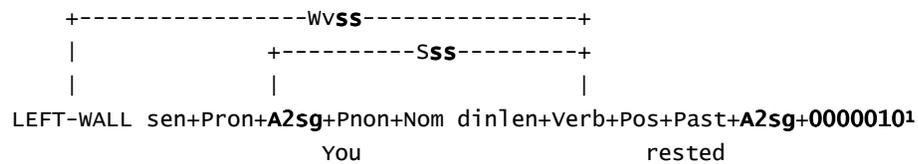


35. Sentence illustrating the usage of an intransitive verb

Sen dinlendin. (You rested.)

You rested

Related Linkage:



36. Sentence illustrating the usage of other verbs

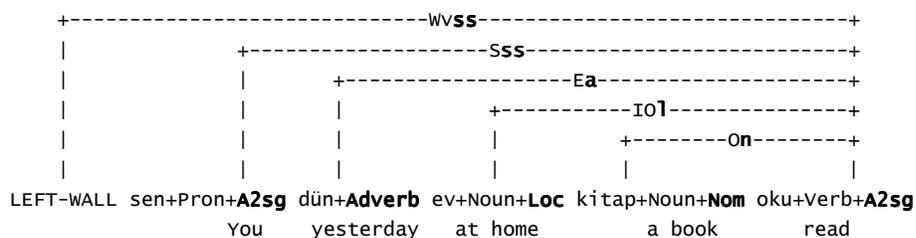
(You read a book at home yesterday.)

Sen dün evde kitap okudun

You yesterday at home book read

¹ It is described in Section 4.2.

Related Linkage:

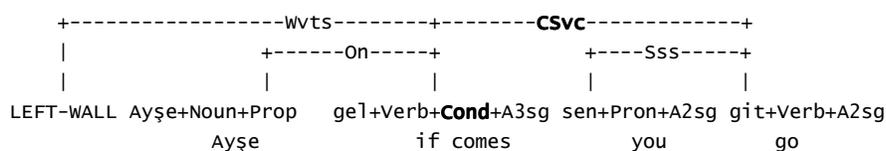


To handle conditional sentences, we connected the verb of the independent clause, i.e. the verb with condition suffix, to the verb of the dependent clause with the “CS” link. In addition, similar to condition suffix, desire suffix do a similar job in Turkish. For this reason, we subscripted this “CS” link with either “d” character to denote desire or “c” character to denote condition suffix. Two examples illustrating both of these usages are given below.

37. Sentence illustrating the usage of a conditional sentence with conditional suffix.

Ayşe gelirse sen gidersin.(If Ayşe comes, you go.)
 Ayşe if comes you go

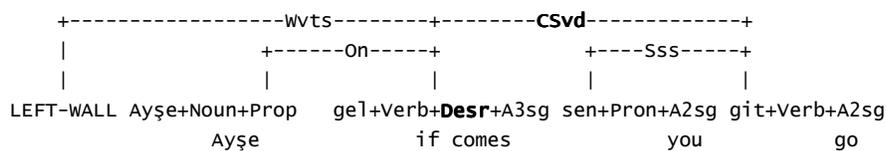
Related Linkage:



38. Sentence illustrating the usage of a conditional sentence with desire suffix.

Ayşe gelse sen gidersin. (If Ayşe comes(i.e. with a desire), you go.)
 Ayşe if comes you go

Related Linkage:



5.4.7 Conjunctions

In Section 5.3 , we described how conjunctions connect sentences together. Here we explain how they connect other constituents of sentences.

As it is mentioned in the previous subsections, Turkish is head-final and hence modifiers always precede the modified item. While connecting words with conjunctions, we tried to choose the words with the same syntactic role. Although, the syntactic role of a word is the combination of the syntactic roles of a word both as a modifier and as a modified, i.e. as head, we choose the words with the same modifier syntactic role to connect. Then, we make the conjunction to play the syntactic role of these conjoined modifiers. The following example illustrates this situation:

39. Sentence illustrating the usage of conjunction:

Sen ve Ayşe geldiniz.(You and Ayşe came)

You and Ayşe came

Related Linkage:



In this example, the conjunction “ve”(and) connects two nouns “sen”(you) and “Ayşe”(Ayşe, a proper name). Since these words are the subject of the verb “gel”(came), they are connected with the links “CLsss” and “CRsts”. The subscript sequence “sss” in “CLsss” denotes that the noun on the left hand side has second singular person suffix. In addition, the subscript sequence “sts” in “CRsts” denotes that the noun on the right hand side has third person singular feature. Then the conjunction “ve” (and) is connected to the verb with the “Ssp” link. The reason for the verb having second plural person feature is the special

usage of the conjunction “ve”(and), i.e. if one of the subjects has second person feature and the other has either second or third person feature, then the verb has to be in second person plural form.

Conjunctions are included in our system in a very detailed fashion, but since their rules are very long and scattered to all word categories, we do not give their detailed descriptions here.

Chapter 6

Performance Evaluation

We tested the performance of our system for coverage with a document consisting of sentences from newspapers. For a better understanding of the results of our test run, first we explore the output of the parser for the following example sentence. In this example, important parts of the cost vector and important links are drawn in bold.

I. Sentence:

Ayşe elbise giydi. (Ayşe wore a dress.)

Ayşe dress wore

Full output of the parser:

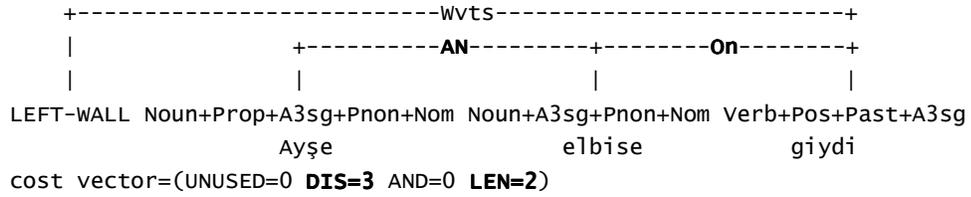
1.1)

```
+-----wvts-----+
|                   +-----Sts-----+
|                   |                   +-----On-----+
|                   |                   |                   |
LEFT-WALL Noun+Prop+A3sg+Pnon+Nom Noun+A3sg+Pnon+Nom Verb+Pos+Past+A3sg
                Ayşe                elbise                giydi
cost vector=(UNUSED=0 DIS=0 AND=0 LEN=3)
```

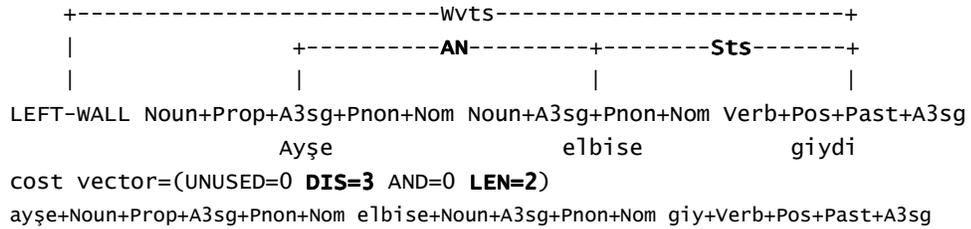
1.2)

```
+-----wvts-----+
|                   +-----On-----+
|                   |                   +-----Sts-----+
|                   |                   |                   |
LEFT-WALL Noun+Prop+A3sg+Pnon+Nom Noun+A3sg+Pnon+Nom Verb+Pos+Past+A3sg
                Ayşe                elbise                giydi
cost vector=(UNUSED=0 DIS=1 AND=0 LEN=3)
```

1.3)



1.4)



In this example, since for each word of the sentence the morphological analyzer finds just one feature structure, there is just one combination of the feature structures of these words to be parsed. For this combination, the grammar outputs four different possible parses. In fact, the first parse is the correct parse. In this parse, the word “Ayşe” is the subject and “elbise”(dress) is the object of the verb “giydi”(wore). As it can be seen in this example, the parser sorts the output first by the costs, i.e. “DIS” parameter, and then by the lengths of the linkages, i.e. “LEN” parameter. “DIS” parameter is determined according to the total costs of the connectors that we assign during the development of the grammar. On the other hand, “LEN” shows the total length of the links in the linkage, i.e. long dependencies have more length, and they are less frequent. In Turkish, although a subject can be between the object and the verb of a sentence, this situation is encountered less frequently. For this reason, for the second position of subject, we give a cost of one and hence the related linkage is printed secondly, i.e. subject is not at the beginning of the sentence. The last two linkages come from the fact that one cannot decide that the string “Ayşe elbise”(Ayşe dress) is not an adjectival nominal group without any semantic knowledge or noun subcategorization information. However, since these kinds of adjectival nominal groups are encountered very rarely in Turkish, we give a

cost of three to these structures. For this reason, they come in last position in this ordering.

Table 4 shows the results of our test run. We collected sentences from domestic, foreign, sports, astrology, and finance news randomly together with sentences from a storybook for children. Before beginning to testing, we removed the punctuation symbols from the sentences and we broke up the sentences into smaller ones to increase the speed of our test. In addition, we removed the incorrect morphological analyses from the results. Our input sentences are given in APPENDIX C and some example outputs from our test run can be found in APPENDIX D.

Number of Sentences	Average number of words in each sentence	Number of sentences for which resulting parses contains the correct parse	Average number of parses	Average ordering of the correct parse
30	4.53	28	5.09	1.92

Table 4 Statistical Results of the Test Run

In the experiment, we used 30 sentences. Average number of words in the sentences was 4.53. Average number of parses per sentences was 5.09. However, for two of the sentences, number of the parses were very high, i.e. 22 and 50, as it can be seen in APPENDIX C. Both of these two sentences contained many consecutive nouns. Since we do not subcategorized nouns for time, place, and title, this resulted in many incorrect indefinite and adjectival nominal groups to be generated and this was the problem in these two sentences. Moreover, one of these sentences consists of words with very complex derivational morphotactics, i.e. many derivational intermediate forms, which results in the number of possible links between these intermediate derived forms to increase. Second, for 28 of the sentences, i.e. 93%, the result set of the parser contained the correct parse. This shows that, though there exists some issues out of our scope, we handle most of the important phenomena, and uncovered issues are encountered very rarely in the language. Lastly, average ordering of the

In fact, for all of these multi-word constructs, a multi-word expression processor is necessary like the one developed by Oflazer, Çetinoğlu, and Say [6].

- Unknown nouns with case suffixes, i.e. not nominative.
 - I. Tigana'nın (Tigana's)
Tigana+Noun+Prop+A3sg+Pnon+Gen

- Abbreviations and numbers with suffixes separated by apostrophe.
 - I. E.K'nın (E.K's)
 - II. saat 02.00'de (at two o'clock)

- Ordered sentences connected by commas.
- We did not use the inflectional features of intermediate derived forms, which is necessary.
 - I. Bizim arabamızdır. (It is our car)
ben+Pron+A1pl+Pnon+Gen(bizim ,our)
araba+Noun+A3sg+P1pl+Nom^DB+Verb+Zero+Pres+Cop+A3sg(
arabamızdır, it is our car)

On the other hand, our system covers many of the syntactic structures in Turkish. These are:

- Noun phrases
- Postpositional phrases
- Dependent clauses constructed by gerunds, participles, and infinitives
- Simple, complex, conditional, and ordered/compound sentences
- Nominal and verbal sentences
- Regular sentences; positive, negative, imperative, and interrogative sentences
- Pronoun drop

- Freely changing order of adverbial phrases, direct and indirect objects, and subject
- Some substantival sentences like quotations.
- Numbers, abbreviations, hyphenated expressions
- Unknown words

Chapter 7

Conclusion

In this thesis, we developed the grammar of Turkish language in the link grammar formalism. Our main aim is to make our scope as complete as possible, i.e. to maximize recall, and we did not concentrate on the running time. In the grammar, we used a fully described morphological analyzer, which is very important for agglutinative languages like Turkish. The grammar that we developed is lexical such that we used the lexemes of only some function words and for the rest of the word classes we used the morphological feature structures. In addition, we preserved the some of the syntactic roles of the intermediate derived forms of words in our system.

The critical connection between language and thought, recent advances in speech recognition and the creation of World –Wide Web resulted in NLP to become a more popular and very important research area. Some of the application areas of NLP include, natural language understanding and generation, informational retrieval, information extraction and machine translation. All of these application areas need some form of syntactic analysis as an underlying process. Although, there is quite a large amount of these applications for some languages like English, Turkish is lesser studied from a computational point of view. For these reasons, we decided to study syntax of Turkish within the light of contemporary linguistic theories and to end up with a syntactic description to be used as a tool in building many useful higher-level applications in the future.

The linguistic theory that we choose to study Turkish syntax is link grammar formalism. It is a very useful tool to develop a syntactic description of a language. It provides user many utilities to handle unknown words, punctuation symbols, hyphenated words, homonyms, number expressions, and idioms. In addition, its cost schema used in the ordering of linkages is very functional. Moreover, the grammar is lexical and this has several important advantages. Since a change in the definition of a word only affects the grammaticality of sentences involving that word, it makes it easier to construct a large grammar incrementally. Furthermore, since the words that are associated semantically and syntactically are linked directly, enforcing agreement, which is encountered frequently in Turkish, is very easy.

As mentioned above, although link grammar formalism is lexical, because of productive morphology of Turkish we used it in a bit different manner. However, instead of using only the morphological feature structures of words, stems of words can also be added to the current system and this results in our current TLG specification to be more precise.

Furthermore, the material that is missing in our system described in Chapter 6 can be added to our system to make our scope as complete as possible. In addition, some statistical information about the relations between the words can be embedded into the system. Lastly, the domain structure and post processing system utilities of the link grammar parser are hard coded and they are not suitable for Turkish. For this reason we cannot use them in our grammar as they are and the implementation of the parser can be changed in the future.

BIBLIOGRAPHY

- [1] Sleator ,D. D. K. and Temperley,D. 1993. Parsing English with a Link Grammar, Third International Workshop on Parsing Technologies.
- [2] Grinberg, Dennis; Lafferty, John and Sleator, Daniel. 1995. A Robust Parsing Algorithm for Link Grammars, Proceedings of the Fourth International Workshop on Parsing Technologies, pp. 111-125.
- [3] Sleator ,D. D. K. and Temperley,D. 1998a. Guide to Links, Provided together with the system or available online at:
<http://www.link.cs.cmu.edu/link/>
- [4] Sleator ,D. D. K. and Temperley,D. 1998a. An Introduction to the Link Grammar Parser, Provided together with the system or available online at:
<http://www.link.cs.cmu.edu/link/dict/introduction.html>
- [5] Lafferty, John; Sleator, Daniel and Temperley, Davy. 1992. Grammatical Trigrams: A Probabilistic Model of Link Grammar. Proceedings of the AAAI Conference on Probabilistic Approaches to Natural Language, October, 1992.
- [6] Oflazer, K.; Çetinoğlu, Ö. And Say,B. 2004. Integrating Morphology with Multi-Word Expression Processing in Turkish. Proceedings of the ACL 2004 Workshop on Multiword Expressions: Integrating Processing, July 2004, Barcelona, Spain.
- [7] Şehitoğlu, O. Tolga. 1996. A Sign-Based Phrase Structure Grammar for Turkish.M.S. Thesis, Middle East Technical University, 1996.

- [8] Güngördü, Zelal. 1993. A Lexical Functional Grammar for Turkish, M.S. Thesis, Bilkent University, 1993.
- [9] Underhill, Robert. 1976. Turkish Grammar. Cambridge: MIT Press.
- [10] Lewis, G. L.. 1988. Turkish Grammar. Oxford University Press.
- [11] Oflazer, K. 1994. Two Level Description of Turkish Morphology. Literary and Linguistic Computing.
- [12] Eryiğit, G., and Oflazer, K. 2006. Statistical Dependency Parsing of Turkish. In Proceedings of EACL 2006 11th Conference of the European Chapter of the Association for Computational Linguistics, Trento, Italy, April.
- [13] Oflazer, K. 1999. Dependency Parsing with an Extended Finite State Approach. In Proceedings of 37th Annual Meeting of the Association for Computational Linguistics, Maryland, USA, June 1999.
- [14] Eker, S. 2005. Çağdaş Türk Dili. Grafiker Yayınları, Ankara, Turkey, 2005.
- [15] Antworth, E.L. 1990. PC-KIMMO: A Two-level Processor for Morphological Analysis, Summer Institute of Linguistics, 1990.
- [16] Jurafsky, D. and Martin, J. H. 2000. Speech and Language Processing. Prentice Hall, New Jersey, USA, 2000.
- [17] Chomsky, N. 1981. Lectures on Government and Binding: The Pisa Lectures. Holland: Foris Publications. Reprint. 7th Edition, Berlin and New York: Mouton de Gruyter, USA, 1993.

- [18] Demir, Coşkun. 1993. An ATN Grammar for Turkish, M.S. Thesis, Bilkent University, 1993.
- [19] Hoffman, Beryl. 1995. The Computational Analysis of the Syntax and Interpretation of 'Free' Word Order in Turkish, PhDthesis, University of Pennsylvania, 1995.
- [20] Bozşahin, C. and Göçmen, E. 1995. A Categorical Framework for Composition in Multiple Linguistic Domains, In Proceedings of the Fourth International Conference on Cognitive Science of NLP, Dublin, Ireland, July 1995.
- [21] Çakıcı ,R.. 2005. Automatic Induction of a CCG Grammar for Turkish, ACL Student Research Workshop , Ann Arbor, MI, July 2005.
- [22] Kemal Oflazer, Bilge Say, Dilek Zeynep Hakkani-Tür, Gökhan Tür, Building a Turkish Treebank, Invited chapter in Building and Exploiting Syntactically-annotated Corpora, Anne Abeille Editor, Kluwer Academic Publishers, 2003. The treebank is available online at:
<http://www.ii.metu.edu.tr/~corpus/treebank.html>
- [23] Nart B. Atalay, Kemal Oflazer, Bilge Say.2003. The Annotation Process in the Turkish Treebank, in Proceedings of the EACL Workshop on Linguistically Interpreted Corpora - LINC, April 13-14, 2003, Budapest, Hungary.

APPENDIX A

Turkish Morphological Features

^DB	Derivation boundary
A1sg	First person singular agreement
A2sg	Second person singular agreement
A3sg	Third person singular agreement
A1pl	First person plural agreement
A2pl	Second person plural agreement
A3pl	Third person plural agreement
Abl	Ablative case for nominal
Acc	Accusative case for nominal
Adj	Adjective
AdjMdfy	Adjective modifier adverbs
Adverb	Adverb
Aor	Aorist tense for verbs
Card	Cardinal numbers
Cond	Conditional for verbs
Conj	Conjunctive
Cop	Copula
Desr	Desire for verbs
Dat	Dative case for nominal
Fut	Future tense for verbs
Gen	Genitive case for nominal
Imp	Imperative for verbs
Ins	Instrumental case for nominal
Interj	Interjection
Loc	Locative case for nominal
Narr	Narrative tense for verbs
Neces	Necessity for verbs
Neg	Negative Polarity
Nom	Nominative case for nominal
Noun	Noun
Num	Number
Ord	Ordinal numbers

P1sg	First person singular possessive agreement
P2sg	Second person singular possessive agreement
P3sg	Third person singular possessive agreement
P1pl	First person plural possessive agreement
P2pl	Second person plural possessive agreement
P3pl	Third person plural possessive agreement
Past	Past tense for verbs
PCNom	Postpositions that take nominative nominal
PCAbI	Postpositions that take ablative nominal
PCDat	Postpositions that take dative nominal
PCIns	Postpositions that take instrumental nominal
PCGen	Postpositions that take genitive nominal
Pnon	No possessive agreement
Pos	Positive Polarity
Postp	Postposition
Pres	Present tense for verbs
Prog1	Progressive time for verbs
Prog2	Another type of progressive time for verbs
Pron	Pronoun
Prop	Proper Name
Opt	Optative for verbs
Verb	Verb
Ques	Question

APPENDIX B

Summary of Link Types

A		connects adjectives to following nouns: Akıllı çocuk (smart child).
AN		connects noun-modifiers to following nouns: Tahta kale (wooden castle)
CL, CLM, CL1, CLKI		connects conjunctions of different types to preceding clauses: Ali ve Veli (Ali and Veli)
CR, CRM, CR1, CRKI		connects conjunctions of different types to following clauses: Ali ve Veli (Ali and Veli)
D	Dn for numbers	Connects determiners (genitive nouns, genitive pronouns and numbers to nouns: Ayşe'nin kitabı (Ayşe's book), üç elma (three apple), Benim kitabım (my book)
	Dg for genitive nouns	
	Dfs for first singular genitive pronouns (g.p)	
	Dss for second singular g.p.	
	Dts for third singular g.p.	
	Dfp for first plural g.p.	
	Dsp for second plural g.p.	
	Dtp for third plural g.p.	
DB		connects words that represent the intermediate, root or the last derivation of the same word.
E	Ea for adverbs	connects adverbs to verbs: Sen hızlı koşuyorsun (You are running quickly)
	Ep for postpositional phrases with adverbial role (w.a.r.)	

	Ei for instrumental nouns (w.a.r.)	
EA	EAp for postpositional phrases (w.a.r.)	connects adverbs to adjectives: O çok akıllı bir çocuk. (He is a very intelligent child)
EE	EEp for postpositional phrases (w.a.r.)	connects adverbs to other adverbs: Sen çok hızlı koşuyorsun. (You run very quickly)
J	Jn for nominative nouns Jg for genitive nouns Jd for dative nouns Ja for accusative nouns	connects postpositions to their objects: Ayşe ile gidiyorum (I am going with Ayşe)
NN		connects number words together in series: Dört yüz bin (Four hundred thousand)
NO		dummy link used for interjections.
O	On for nominative nouns Oc for Accusative nouns	connects verbs to their direct objects: Sen kitabı okuyorsun (You are reading the book).
IO	IOl for locative nouns IOd for dative nouns	connects verbs to their indirect objects: Sen kitap okuyorsun (You are reading book).

	IOa for ablative nouns	
QB	QBr for regular question morpheme(q.m.)	connects the question morpheme “-mi” to preceding word: Ayşe geliyor mu ? (Is Ayşe coming?).
	QBv q.m. connected to verbs	
	QBc q.m. connected to copula(with or without copula suffix)	
CQ		connects the question morpheme “-mi” to following special conjunctions: Ali mi yoksa Ayşe mi geliyor? (Is Ayşe or Ali coming?)
S	Sfs for first singular subject	connects subject noun phrases to finite verbs: Ayşe geliyor . (Ayşe is coming)
	Sss for second singular subject	
	Sts for third singular subject	
	Sfp for first plural subject	
	Ssp for second plural subject	
	Stp for third plural subject	

W	Wc for conjunctions (Wcc, Wcm, Wc, Wck, etc, for different types of conjunctions)	connects predicate of main clause or conjunction, which connect verbs, to the wall.
	Wv for verbs (Wfs, Wss, Wts, Wfp, Wsp, Wtp)	

APPENDIX C

Input Document and Statistical Results

A	B	C	D	E
5	İsrail Lübnan'a yönelik saldırılarını durdurdu	1	4	1
5	Elif'den ailesi haber alamadı	1	4	1
5	Ağabey Polat Elif'in işyerine gitti	1	8	3
4	Kardeşinin işyerinden çıktığını öğrendi	1	1	1
4	Mensa konfeksiyon odaklı çalışacak	1	8	3
5	hava saldırılarını 48 saat süreyle durdurdu	1	5	4
3	Mazlumder üyeleri yerleştirdiler	1	4	3
3	ellerindeki fotoğrafları yerleştirdiler	1	2	1
3	Üyeler fotoğrafları yerleştirdiler	1	2	1
5	daha önceden hazırlanan şövalyelere yerleştirdiler	1	22	1
4	ağabey kaçırıldığı iddiasıyla başvurdu	1	14	12
5	Gönülsüz bir iş olmasın istedik	1	4	1
3	Kardeşimi geri getirsinler	1	1	1
5	kardeşimi getirmelerini istiyorum diye konuştu	1	1	1
4	İsrail polisi haberleri yalanladı	1	4	1
5	gerillaların kuzeye saldırdığı haberlerini yalanladı	1	3	3
4	Hisse senetleri değer kazandı	1	2	1
4	sahaya Çıkan Cimbom isteksizdi	1	4	1
3	Akşama doğru rahatlayacaksınız	1	3	3
5	yaşamınıza daha çok vakit ayıracaksınız	1	2	1
5	Ayrıca küçük bir hediye alacaksınız	1	1	1
3	huzurlu olduğunuz görülüyor	1	8	3
2	KİBRİTÇİ KIZ	1	1	1
3	Bir yılbaşı gecesiydi	1	1	1
6	Dondurucu ve kavurucu bir soğuk vardı	0	1	0
5	Yoldan geçenler paltolarının yakasını kaldırmışlar	1	1	1
6	Çocuklar koşuyorlar ve birbirlerine kartopu atıyorlardı	1	4	1
6	Gecenin zevkini en çok onlar çıkarıyorlardı	1	3	1
7	Ufak bir kız çocuğu tir tir titriyordu	0	10	0
9	Kekikli yağlı çorbanın üzerinde gezdirip sıcak olarak servise hazırlayın	1	50	1

A = Number of words in the sentence

B = Sentence

C = Does the resulting parse set contain the correct parse (1 is YES and 0 is NO)

D = Number of possible parses found for the sentence

E = Place of the correct parse in the result set

APPENDIX D

D Example Output from Our Test Run

In the following sentences, incorrect morphological features structures are not given and the right answer is given in bold. In addition, the input sentences are given in italics and underlined.

1. *İsrail Lübnan'a yönelik saldırılarını durdurdu*

1)

```
+-----Wvts-----+
|                   +-----Sts-----+
|                   |                   +-----Jd-----+-----Ap-----+-----Oc-----+
|                   |                   |                   |                   |                   |
LEFT-WALL Noun+Prop+A3sg+Pnon+Nom Noun+Prop+A3sg+Pnon+Dat Postp+PCDat Noun+A3sg+P3pl+Acc Verb+Pos+Past+A3sg
```

cost vector=(UNUSED=0 DIS=0 AND=0 LEN=7)

2)

```
+-----Wvts-----+
|                   +-----Sts-----+
|                   |                   +-----Ep-----+
|                   |                   |                   +-----Oc-----+
|                   |                   |                   |                   |
LEFT-WALL Noun+Prop+A3sg+Pnon+Nom Noun+Prop+A3sg+Pnon+Dat Postp+PCDat Noun+A3sg+P3pl+Acc Verb+Pos+Past+A3sg
```

cost vector=(UNUSED=0 DIS=0 AND=0 LEN=8)

3)

```
+-----Wvts-----+
|               +-----AN-----+-----Jd-----+-----Ap-----+-----Oc-----+
|               |               |               |               |               |
LEFT-WALL Noun+Prop+A3sg+Pnon+Nom Noun+Prop+A3sg+Pnon+Dat Postp+PCDat Noun+A3sg+P3pl+Acc Verb+Pos+Past+A3sg
```

cost vector=(UNUSED=0 DIS=4 AND=0 LEN=4)

4)

```
+-----Wvts-----+
|               +-----AN-----+-----Jd-----+-----Ep-----+
|               |               |               |               |
LEFT-WALL Noun+Prop+A3sg+Pnon+Nom Noun+Prop+A3sg+Pnon+Dat Postp+PCDat Noun+A3sg+P3pl+Acc Verb+Pos+Past+A3sg
```

cost vector=(UNUSED=0 DIS=4 AND=0 LEN=5)

israil+Noun+Prop+A3sg+Pnon+Nom lübnan+Noun+Prop+A3sg+Pnon+Dat yönelik+Postp+PCDat
saldırı+Noun+A3sg+P3pl+Acc dur+Verb^DB+Verb+Caus+Pos+Past+A3sg

2. Elif'den ailesi haber alamadı

1)

```
+-----Wvts-----+
|               +-----IOa-----+
|               |               +-----Sts-----+
|               |               |               +-----On-----+
|               |               |               |
LEFT-WALL Noun+Prop+A3sg+Pnon+Abl Noun+A3sg+P3sg+Nom Noun+A3sg+Pnon+Nom Verb+Neg+Past+A3sg
```

cost vector=(UNUSED=0 DIS=1 AND=0 LEN=6)

2)

```
+-----Wvts-----+
|               +-----IOa-----+
```

```

|           |           |           |           |           |           |
|           |           |           |           |           |           |
|           |           |           |           |           |           |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|           |           |           |           |           |           |           |           |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|           |           |           |           |           |           |           |           |
LEFT-WALL Noun+Prop+A3sg+Pnon+Abl Noun+A3sg+P3sg+Nom Noun+A3sg+Pnon+Nom Verb+Neg+Past+A3sg

```

cost vector=(UNUSED=0 DIS=2 AND=0 LEN=6)

3)

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|           |           |           |           |           |           |           |           |
|           |           |           |           |           |           |           |           |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|           |           |           |           |           |           |           |           |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|           |           |           |           |           |           |           |           |
LEFT-WALL Noun+Prop+A3sg+Pnon+Abl Noun+A3sg+P3sg+Nom Noun+A3sg+Pnon+Nom Verb+Neg+Past+A3sg

```

cost vector=(UNUSED=0 DIS=4 AND=0 LEN=5)

4)

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|           |           |           |           |           |           |           |           |
|           |           |           |           |           |           |           |           |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|           |           |           |           |           |           |           |           |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|           |           |           |           |           |           |           |           |
LEFT-WALL Noun+Prop+A3sg+Pnon+Abl Noun+A3sg+P3sg+Nom Noun+A3sg+Pnon+Nom Verb+Neg+Past+A3sg

```

cost vector=(UNUSED=0 DIS=4 AND=0 LEN=5)

elif+Noun+Prop+A3sg+Pnon+Abl aile+Noun+A3sg+P3sg+Nom haber+Noun+A3sg+Pnon+Nom
al+Verb^DB+Verb+AbleNeg+Neg+Past+A3sg

3. Ağabey Polat Elif'in işyerine gitti

1)

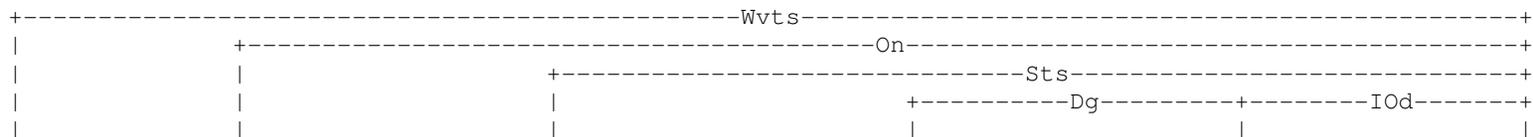
```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|           |           |           |           |           |           |           |           |
|           |           |           |           |           |           |           |           |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|           |           |           |           |           |           |           |           |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|           |           |           |           |           |           |           |           |
|           |           |           |           |           |           |           |           |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|           |           |           |           |           |           |           |           |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|           |           |           |           |           |           |           |           |

```

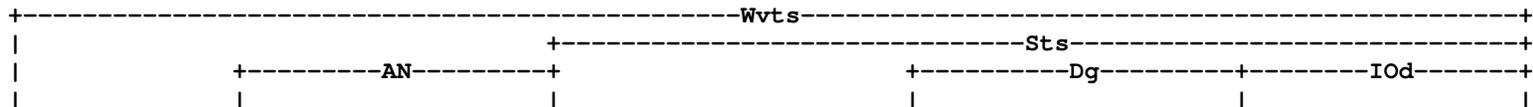
LEFT-WALL Noun+A3sg+Pnon+Nom Noun+Prop+A3sg+Pnon+Nom Noun+Prop+A3sg+Pnon+Gen Noun+A3sg+P3sg+Dat
 Verb+Pos+Past+A3sg

cost vector=(UNUSED=0 DIS=0 AND=0 LEN=9)
 2)



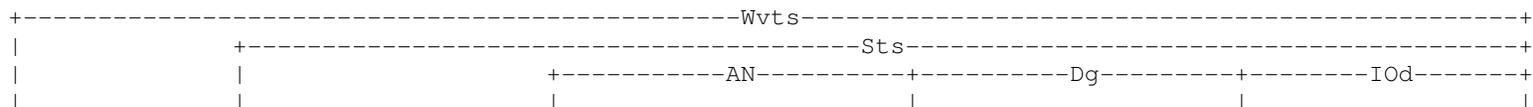
LEFT-WALL Noun+A3sg+Pnon+Nom Noun+Prop+A3sg+Pnon+Nom Noun+Prop+A3sg+Pnon+Gen Noun+A3sg+P3sg+Dat
 Verb+Pos+Past+A3sg

cost vector=(UNUSED=0 DIS=1 AND=0 LEN=9)
 3)



**LEFT-WALL Noun+A3sg+Pnon+Nom Noun+Prop+A3sg+Pnon+Nom Noun+Prop+A3sg+Pnon+Gen Noun+A3sg+P3sg+Dat
 Verb+Pos+Past+A3sg**

cost vector=(UNUSED=0 DIS=3 AND=0 LEN=6)
 4)



LEFT-WALL Noun+A3sg+Pnon+Nom Noun+Prop+A3sg+Pnon+Nom Noun+Prop+A3sg+Pnon+Gen Noun+A3sg+P3sg+Dat
 Verb+Pos+Past+A3sg

cost vector=(UNUSED=0 DIS=3 AND=0 LEN=7)
 5)



```

|          +-----AN-----+
|          |          +-----AN-----+          |          |          |          |
|          |          |          |          |          |          |          |          |
LEFT-WALL Noun+A3sg+Pnon+Nom Noun+Prop+A3sg+Pnon+Nom Noun+Prop+A3sg+Pnon+Gen Noun+A3sg+P3sg+Dat
Verb+Pos+Past+A3sg

```

cost vector=(UNUSED=0 DIS=4 AND=0 LEN=5)

6)

```

+-----Wvts-----+
|          +-----AN-----+          |          |          |          | | | | |
|          |          +-----AN-----+          |          |          |          |
|          |          |          |          |          |          |          |          |
LEFT-WALL Noun+A3sg+Pnon+Nom Noun+Prop+A3sg+Pnon+Nom Noun+Prop+A3sg+Pnon+Gen Noun+A3sg+P3sg+Dat
Verb+Pos+Past+A3sg

```

cost vector=(UNUSED=0 DIS=4 AND=0 LEN=6)

7)

```

+-----Wvts-----+
|          +-----AN-----+          |          |          |          | | | | |
|          |          +-----AN-----+          |          |          |          |
|          |          |          |          |          |          |          |          |
LEFT-WALL Noun+A3sg+Pnon+Nom Noun+Prop+A3sg+Pnon+Nom Noun+Prop+A3sg+Pnon+Gen Noun+A3sg+P3sg+Dat
Verb+Pos+Past+A3sg

```

cost vector=(UNUSED=0 DIS=4 AND=0 LEN=7)

8)

```

+-----Wvts-----+
|          +-----AN-----+          |          |          |          | | | | |
|          |          +-----AN-----+          |          |          |          |
|          |          |          |          |          |          |          |          |
LEFT-WALL Noun+A3sg+Pnon+Nom Noun+Prop+A3sg+Pnon+Nom Noun+Prop+A3sg+Pnon+Gen Noun+A3sg+P3sg+Dat
Verb+Pos+Past+A3sg

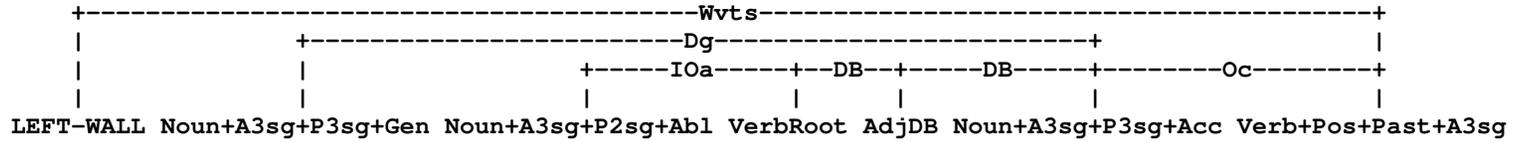
```

cost vector=(UNUSED=0 DIS=7 AND=0 LEN=4)

ağabey+Noun+A3sg+Pnon+Nom polat+Noun+Prop+A3sg+Pnon+Nom elif+Noun+Prop+A3sg+Pnon+Gen
işyeri+Noun+A3sg+P3sg+Dat git+Verb+Pos+Past+A3sg

4. Kardeşinin işyerinden çıktığıını öğrendi

1)

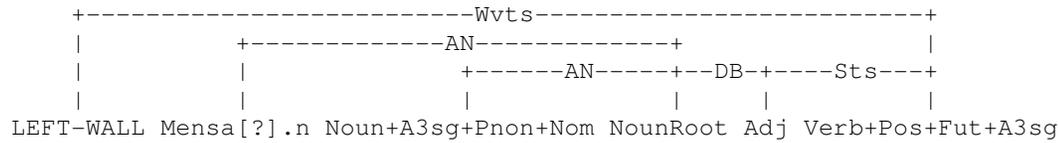


cost vector=(UNUSED=0 DIS=2 AND=0 LEN=8)

kardeş+Noun+A3sg+P3sg+Gen işyeri+Noun+A3sg+P2sg+Abl çık+Verb+Pos^DB+Adj+PastPart^DB+Noun+Zero+A3sg+P3sg+Acc
öğren+Verb+Pos+Past+A3sg

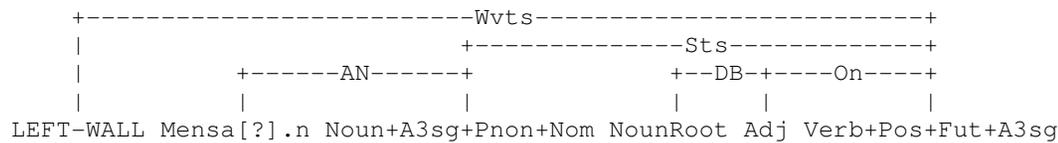
5. Mensa konfeksiyon odaklı çalışacak

1)



cost vector=(UNUSED=0 DIS=4 AND=0 LEN=5)

2)



cost vector=(UNUSED=0 DIS=4 AND=0 LEN=6)

3)



```

|           +-----Sts-----+
|           |           +-----AN-----+---DB-+---On-----+
|           |           |           |           |           |
LEFT-WALL Mensa[?].n Noun+A3sg+Pnon+Nom NounRoot Adj Verb+Pos+Fut+A3sg

```

cost vector=(UNUSED=0 DIS=4 AND=0 LEN=7)

4)

```

+-----Wvts-----+
|           +-----AN-----+
|           |           +-----AN-----+---DB-+---On-----+
|           |           |           |           |
LEFT-WALL Mensa[?].n Noun+A3sg+Pnon+Nom NounRoot Adj Verb+Pos+Fut+A3sg

```

cost vector=(UNUSED=0 DIS=5 AND=0 LEN=5)

5)

```

+-----Wvts-----+
|           +-----On-----+
|           |           +-----AN-----+---DB-+---Sts-----+
|           |           |           |           |
LEFT-WALL Mensa[?].n Noun+A3sg+Pnon+Nom NounRoot Adj Verb+Pos+Fut+A3sg

```

cost vector=(UNUSED=0 DIS=5 AND=0 LEN=6)

6)

```

+-----Wvts-----+
|           +-----On-----+
|           |           +-----AN-----+---DB-+---Sts-----+
|           |           |           |           |
LEFT-WALL Mensa[?].n Noun+A3sg+Pnon+Nom NounRoot Adj Verb+Pos+Fut+A3sg

```

cost vector=(UNUSED=0 DIS=5 AND=0 LEN=7)

7)

```

+-----Wvts-----+
|           +-----AN-----+-----AN-----+---DB-+---Sts-----+
|           |           |           |           |

```

LEFT-WALL Mensa[?].n Noun+A3sg+Pnon+Nom NounRoot Adj Verb+Pos+Fut+A3sg

cost vector=(UNUSED=0 DIS=7 AND=0 LEN=4)

8)

```
+-----Wvts-----+
|           +-----AN-----+-----AN-----+---DB-+-----On-----+
|           |           |           |           |           |
|           |           |           |           |           |
```

LEFT-WALL Mensa[?].n Noun+A3sg+Pnon+Nom NounRoot Adj Verb+Pos+Fut+A3sg

cost vector=(UNUSED=0 DIS=8 AND=0 LEN=4)

Mensa konfeksiyon+Noun+A3sg+Pnon+Nom odak+Noun+A3sg+Pnon+Nom^DB+Adj+With çalış+Verb+Pos+Fut+A3sg

6. hava saldırılarını 48 saat süreyle durdurdu

1)

```
+-----Wvts-----+
|           +-----Oc-----+
|           |           +-----Sts-----+
|           |           |           +-----Ei-----+
|           |           |           |           +-----AN-----+
|           |           |           |           |           +-----Dn-----+
|           |           |           |           |           |
```

LEFT-WALL Noun+A3sg+Pnon+Nom Noun+A3pl+P3sg+Acc 48 Noun+A3sg+Pnon+Nom Noun+A3sg+Pnon+Ins Verb+Pos+Past+A3sg

cost vector=(UNUSED=0 DIS=1 AND=0 LEN=9)

2)

```
+-----Wvts-----+
|           +-----Sts-----+
|           |           +-----Oc-----+
|           |           |           +-----Ei-----+
|           |           |           |           +-----AN-----+
|           |           |           |           |           +-----Dn-----+
|           |           |           |           |           |
```

LEFT-WALL Noun+A3sg+Pnon+Nom Noun+A3pl+P3sg+Acc 48 Noun+A3sg+Pnon+Nom Noun+A3sg+Pnon+Ins Verb+Pos+Past+A3sg

cost vector=(UNUSED=0 DIS=3 AND=0 LEN=12)

3)

```
+-----Wvts-----+
```

```

|           +-----Sts-----+
|           |                   +-----Oc-----+
|           |                   |                   +-----Dn-----+
|           |                   |                   |                   +-----AN-----+
|           |                   |                   |                   |                   +-----Ei-----+
LEFT-WALL Noun+A3sg+Pnon+Nom Noun+A3pl+P3sg+Acc 48 Noun+A3sg+Pnon+Nom Noun+A3sg+Pnon+Ins Verb+Pos+Past+A3sg

```

cost vector=(UNUSED=0 DIS=3 AND=0 LEN=13)

4)

```

+-----Wvts-----+
|                   +-----Oc-----+
|                   |                   +-----Dn-----+
|                   |                   |                   +-----AN-----+
|                   |                   |                   |                   +-----Ei-----+
LEFT-WALL Noun+A3sg+Pnon+Nom Noun+A3pl+P3sg+Acc 48 Noun+A3sg+Pnon+Nom Noun+A3sg+Pnon+Ins Verb+Pos+Past+A3sg

```

cost vector=(UNUSED=0 DIS=4 AND=0 LEN=8)

5)

```

+-----Wvts-----+
|                   +-----Oc-----+
|                   |                   +-----Dn-----+
|                   |                   |                   +-----AN-----+
|                   |                   |                   |                   +-----Ei-----+
LEFT-WALL Noun+A3sg+Pnon+Nom Noun+A3pl+P3sg+Acc 48 Noun+A3sg+Pnon+Nom Noun+A3sg+Pnon+Ins Verb+Pos+Past+A3sg

```

cost vector=(UNUSED=0 DIS=4 AND=0 LEN=9)

hava+Noun+A3sg+Pnon+Nom saldırı+Noun+A3pl+P3sg+Acc 48 saat+Noun+A3sg+Pnon+Nom süre+Noun+A3sg+Pnon+Ins
dur+Verb^DB+Verb+Caus+Pos+Past+A3sg

7. daha önceden hazırlanan şövalyelere yerleştirdiler

1)

```

+-----Wvtp-----+
|                   +-----Ea-----+
|                   |                   +-----EE-----+
|                   |                   |                   +-----DB-----+
|                   |                   |                   |                   +-----DB-----+
|                   |                   |                   |                   |                   +-----A-----+
|                   |                   |                   |                   |                   |                   +-----IOd-----+

```

LEFT-WALL Adverb+AdjMdfy Adverb NounDB VerbDB Adj Noun+A3pl+Pnon+Dat Verb+Pos+Past+A3pl

cost vector=(UNUSED=0 DIS=2 AND=0 LEN=7)

2)

```
+-----Wvtp-----+
|                   +-----Ea-----+   +-----Sts-----+
|                   +-----EE-----+   +---DB---+DB--+   +-----IOd-----+
|                   |                   |                   |                   |
LEFT-WALL Adverb+AdjMdfy Adverb NounDB VerbDB Adj Noun+A3pl+Pnon+Dat Verb+Pos+Past+A3pl
```

cost vector=(UNUSED=0 DIS=2 AND=0 LEN=8)

...
...
...

21)

```
+-----Wvtp-----+
|                   +-----Ea-----+
|                   |                   +-----Ea-----+
|                   |                   |                   +-----On-----+
|                   |                   +---DB---+DB--+   +-----IOd-----+
|                   |                   |                   |                   |
LEFT-WALL Adverb+AdjMdfy Adverb NounDB VerbDB Adj Noun+A3pl+Pnon+Dat Verb+Pos+Past+A3pl
```

cost vector=(UNUSED=0 DIS=3 AND=0 LEN=16)

22)

```
+-----Wvtp-----+
|                   +-----Ea-----+
|                   |                   +-----Ea-----+
|                   |                   |                   +-----Sts-----+
|                   |                   +---DB---+DB--+   +-----IOd-----+
|                   |                   |                   |                   |
LEFT-WALL Adverb+AdjMdfy Adverb NounDB VerbDB Adj Noun+A3pl+Pnon+Dat Verb+Pos+Past+A3pl
```

cost vector=(UNUSED=0 DIS=4 AND=0 LEN=16)

daha+Adverb+AdjMdfy önceden+Adverb hazır+Adj^DB+Noun+Zero+A3sg+Pnon+Nom^DB+Verb+Acquire+Pos^DB+Adj+PresPart
şövalye+Noun+A3pl+Pnon+Dat yerleş+Verb^DB+Verb+Caus+Pos+Past+A3pl