

**T.C.
Mersin Üniversitesi
Sosyal Bilimler Enstitüsü
Yabancı Diller Eğitimi Ana Bilim Dalı**

**DESIGN OF A FINITE-STATE TRANSDUCER
FOR PARTS OF SPEECH TAGGING OF TURKISH**

Ümit MERSİNLİ

YÜKSEK LİSANS TEZİ

Mersin,2010

Mersin Üniversitesi, Sosyal Bilimler Enstitüsü Müdürlüğüne,

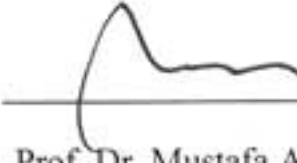
Okt. Ümit MERSİNLİ tarafından hazırlanan "Design of a Finite-State Transducer for Parts of Speech Tagging of Turkish" başlıklı bu çalışma, jürimiz tarafından Yabancı Diller Eğitimi Ana Bilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Başarılı

Başarısız



Başkan



Prof. Dr. Mustafa AKSAN
(Danışman)



Üye



Prof. Dr. Yeşim AKSAN



Üye



Yrd. Doç. Dr. Şaziye YAMAN

Onay

Yukarıdaki imzaların, adı geçen öğretim elemanlarına ait olduklarını onaylarım.

09/06/2010


Prof. Dr. Mustafa AKSAN
Enstitü Müdürü

ACKNOWLEDGMENTS

First and foremost, I would like to thank to my advisor Prof. Dr. Mustafa AKSAN for his patience during the great learning experience I had, and Prof. Dr. Yeşim AKSAN for her encouragement and trust.

I'm also indebted to all colleagues at the Linguistics Department of Mersin University for their invaluable support.

I'm also grateful to all other members of the Turkish National Corpus (TNC) Team for their courage and enthusiasm in building TNC that this study would never be possible without the data they provided.

It would be impossible to name all those who inspired, taught, and supported me. In brief, my special thanks go to all my colleagues, friends, students and my family.

ÖZET

Bu çalışma, Türkçe’de sözcük türü etiketleme amaçlı ve kural-tabanlı bir Sonlu-Durum çeviricinin kökten-ekle bir yaklaşımla tasarlanabileceğini önerir.

Çalışmanın ilk bölümü Doğal Dil İşlemede Sonlu-Durum çeviricilerin kullanımını içeren çalışmaları özetlemekte ve Türkçe’deki bazı uygulamalara değinmektedir.

Yöntem bölümünde, çalışmanın yazılım değerlendirme, veri toplama ve sözlük oluşturma aşamaları ayrıntılandırılmaktadır.

Sonlu-durum Çevirici Tasarımı başlıklı bölümde ise Türkçe’nin türetim ve çekim süreçlerinin modellenmesinin aşamaları ve güçlüklerine değinilmektedir.

Sonuç bölümü ise çalışmanın kısa değerlendirmesini ve önerileri sunar.

Anahtar sözcükler: Sözcük Türü Etiketleme, Türkçe’nin Biçimbilimi, Biçimbirim Sıralaması, Nooj, Sonlu-Durum Çevirici Düzenegi

ABSTRACT

This study proposes that a rule-based Finite-State Transducer for Turkish Part of Speech Tagging can be designed in a root-to-affix approach.

The Introduction part of the study summarizes the studies on Finite-State Transducers for Natural Language Processing and mentions some applications for Turkish.

In the Methodology section, details of the software evaluation, data collection and dictionary compilation stages are given.

In the Components of the Transducer section, the stages of modeling the inflectional and derivational processes and the challenges are mentioned.

Conclusion section presents the overview of the study and recommendations.

Keywords: Part of Speech Tagging, Turkish Morphology, Morpheme Order, Nooj, Finite-State Transducer Automata

CONTENTS

ACKNOWLEDGMENTS.....	i
ABSTRACT.....	ii
CONTENTS.....	iv
TABLES AND FIGURES.....	vi
INTRODUCTION.....	1
Preliminaries	1
Morphotactics of Turkish.....	3
Finite-State Morphology of Turkish	4
Statement of the Problem	5
Purpose of the Study	5
Importance of the Problem	6
Research Questions	6
Hypotheses.....	6
Data Collection Techniques	6
Operational Definitions.....	7
Limitations	8
I. METHODOLOGY.....	9
I.1. Software Evaluation	11
I.2. Data Collection	12
I.2.1. Corpus	12
I.2.2. Tokenization	13
I.2.3. Lemmatization	14

II. COMPONENTS OF THE TRANSDUCER.....	15
II.1. Overview of Nooj Grammars	15
II.2. Architecture of Turkish Module	16
II.3. Dictionaries	16
II.3.1. Lexical Categories	18
II.3.2. Phonemic Alternations	19
II.4. Graphs	22
II.4.1. Derivation	23
II.4.2. Inflection (Nominal Paradigm)	24
II.4.3. Inflection (Verbal Paradigm)	26
III. IMPLEMENTATION AND TESTING.....	27
CONCLUSION	31
Summary	31
Results of the Study	31
Recommendations	31
REFERENCES	33
APPENDICES	36

TABLES AND FIGURES

Table 1. Statistics For The Affixation Of Turkish	4
Table 2. Possible decompositions with left-to-right and right-to-left processing ..	10
Table 4. Rule specifications for in-root phonemic alternations in Turkish	20
Table 5. Operators in Nooj grammar formalism	21
Figure 1. Corpus interface of Nooj.....	12
Figure 2. Sample tokens ordered by frequency	13
Figure 3. Nooj grammars	15
Figure 4. Dictionary compilation interface.....	17
Figure 5. Sample graph representing allomorphs of an affix.....	22
Figure 6. Main graph with the nominal and verbal paradigms	23
Figure 7. Sample derivational graph	24
Figure 8. Sample derivational subgraph	24
Figure 9. Main Graph for Nominal Inflection in Turkish.....	25
Figure 10. Stems for nominal inflection	25
Figure 11. Nominal Inflection in Turkish	26
Figure 12. Verbal Inflection in Turkish	26
Figure 13. Sample Annotation 1	27
Figure 14. Sample Annotation 2	27

INTRODUCTION

The persuasiveness of *Syntactic Structures* had the effect that, for many decades to come, computational linguists directed their efforts towards more powerful formalisms. Finite-state automata as well as statistical approaches disappeared from the scene for a long time. Today the situation has changed in a fundamental way: statistical language models are back and so are finite-state automata, in particular, finite-state transducers (Karttunen, 2001).

Finite-State Transducers (FST) in Natural Language Processing (NLP) have a rather suppressed history. This study will also begin with providing a brief outline of that history.

Preliminaries

As stated in Joshi (1997), the use of finite-state automata for rule-based NLP begins with Transformations and Discourse Analysis Project (TDAP) directed by Zellig S. Harris at Pennsylvania University in 1958.

Another implementation was the DeComp module of MITalk system dating back to the early 1960s. As the name implies, it was a decomposer, morphological analyzer that used affix-stripping or right-to-left approach to analyze the morphemes of a given word in English (Sproat, 1992: 185).

The *keçi* system designed by Hankamer for Turkish was also unique in that it was implemented specifically for Turkish. As Sproat (1992) points out, it was designed with a motivation “to check the typographical accuracy of a corpus of Turkish text that had been typed into a computer” and thus originally checking the harmony rules of Turkish. Its tags for each morpheme was like N0, N1, indicating only the ordering of them.

Then, the method *two-level morphology* is studied by Ronald M. Kaplan and Martin Kay in the 1970s and implemented by Kimmo Koskenniemi in 1983 as KIMMO

(Roark, 2007). In a conference on parsing organized by Lauri Karttunen in 1981, “the four Ks” discussed and then formed the basics of “the first general model in the history of computational linguistics for the analysis and generation of morphologically complex languages” (Karttunen, 2005).

As Karttunen (2005) states, the system was “a new way to describe phonological alternations in finite-state terms”. The power of KIMMO-style programs was that - especially for languages like Turkish, Finnish — they are able to simulate the phoneme alternations of highly concatenative languages by parallel working rule specifications. Thus, the system was used later widespreadly, especially for rule-based processing of phonological phenomena of Turkish. In terms of morphological analysis or morpheme tagging, multi-level transducers were regarded as more appropriate and practical. As Karttunen points out, “it became evident that lexical transducers are easier to construct with sequentially applied replace rules than with two-level rules”.

Among the mentioned approaches, Hankamer’s *keçi* is of special interest since it shows the significance of Turkish as a language with “purely concatenative morphology” even in the earliest phases of finite-state morphology (Sproat, 1992). However, it was, “while certainly more powerful in what it can accomplish than was DeComp, is more limited than KIMMO” (Sproat, 1992).

As the most successful of the approaches above, KIMMO-style programs went on being improved for various languages such as Finnish, Turkish etc.

In the methodology section, theoretical considerations related to the early software mentioned above will be discussed again in more detail.

Morphotactics of Turkish

Turkish is often cited as a representative example to highly concatenative or agglutinative languages. It is almost a tradition to cite rather exaggerated samples from the affixation process of Turkish as in (1-5);

- (1) öl-üm-süz-leş-tir-t-tir-il-e-me-yebil-in-en-ler-de-ki-ler-den-mi-ymiş-ler-ce-sin-e

Is it as if they are of those that belong to the ones which one may not be able to get immortalized?

(Sebüktekin, 1974)

- (2) çöp + lük +ler +imiz +de +ki +ler +den +mi +y +di

Was it from those that were in our garbage cans?

(Hankamer, 1986)

c.f. (Sproat, 1992: 20)

- (3) osmanlı +laş +tır +ama +yabil +ecek +ler +imiz +den +miş +sınız

Behaving as if you were of those whom we might consider not converting into an Ottoman.

(Oflazer, 1994a)

- (4) uygar +laş +tır +ama +dık +lar +ımız +dan +mış +sınız +casına

Behaving as if you are among those whom we could not civilize.

(Oflazer, 1994b)

c.f. (Jurafsky, 2006)

- (5) masa +lar +ım +da +ki +ler +in +ki +nde

At those (things) which belong to those (other things) at my tables.

(Oflazer, 1994a)

Although the examples above are statistically rare, as the values in **Table 1** presents, Turkish is *extremely* concatenative when compared to languages like English.

maximum number of suffixes	8
average number of suffixes for all words	0.94
average number of suffixes for affixed words	1.85
maximum suffix length	7

Table 1. Statistics for the affixation of Turkish. (Güngör, 2003)

As Sebüktekin (1974) states, “morphotactics, *then*, should have an important place in any discussion of Turkish structure”. Work of linguists on the complex morphology of Turkish provided insights especially into the modeling process of this study.

Sebüktekin (1974), as an earlier study, demonstrates the morpheme order of the verbal paradigm in Turkish in a linear manner. All possible combinations of verbal affixes are presented in detail. Sebüktekin also argues that algebraic formulae is a more adequate formalism than geometric graphs. Although relying upon graphical representations while simulating the affixation of Turkish, this study also took benefit of the combination lists provided by the mentioned study.

Other studies on the morphotactics of Turkish are mostly computational finite-state models and discussed in the next chapter of the study.

Finite-State Morphology of Turkish

Turkish, has been an area of special interest even in the beginning phases of finite-state morphology as seen in Hankamer (1989).

In terms of two-level morphology, studies on Turkish begin with the publication of PC-KIMMO rule specifications (Oflazer, 1994a) and their implementation

(Oflazer, 1994b) for Turkish. Another similar application based on the same rule specifications was implemented in Prolog programming language and argued to be “more efficient than the PC-Kimmo system” (Çiçekli, 1997).

The affix-stripping or right-to-left approaches to the morphological analysis of Turkish are the implementations of Sever (2003), Adalı (2002, 2004) and Çilden (2006).

Among the stochastic studies, we can mention the study of Sak (2009).

Another natural language processing application that we can mention among the finite-state approaches to Turkish is the open-source *Zemberek* project which is basically designed as a spell-checker (Akın, 2007).

Statement of the Problem

Although the previously mentioned studies provided various models on the morphotactics of Turkish, a graph-based, open-source finite-state transducer with pre-tagged, corpus-based dictionaries and adequate models of Turkish morphology is needed in order to lead to further developments in Turkish NLP. Such an implementation will provide graphical representations of Turkish morphotactics, easily adaptable to other formalisms and together with its corpus-based, pre-tagged dictionaries, be completely accessible to linguists and researchers for review, testing and modification.

Purpose of the Study

The aim of this study is to describe the processes and principles of constructing an open-source, graph-based, root-driven, non-stochastic finite-state transducer and its components for Parts of Speech Tagging of Turkish.

Importance of the Problem

The resulting graphical representations of Turkish morphology and the pre-tagged dictionaries of the transducer will provide input for further studies. This study will also make it easier to construct annotated user corpora in Turkish without requiring much computational background and lead to improvements in parallel/bilingual corpora design for ELT researchers as well as linguists.

Research questions

1. Can a finite-state transducer for parts of speech tagging of Turkish be designed?
2. Can finite-state transducers be used for the morphological analysis and tagging of Turkish language corpora with a left-to-right or root-to-affix approach?

Hypotheses

1. A finite-state transducer for parts of speech tagging of Turkish can be designed.
2. Finite-state transducers can be used for the morphological analysis and tagging of Turkish language corpora with a left-to-right or root-to-affix approach.

Data collection techniques

Data for this study are derived from the Turkish National Corpus (TNC) project held by English Language and Literature Department at Mersin University and funded by the Scientific and Technological Research Council of Turkey (TÜBİTAK) for a three-year period (2008-2011).

Operational Definitions

Morpheme: “The minimal distinctive unit of grammar”. “The smallest functioning unit in the composition of words” (Crystal, 2003).

Parts-of-speech tagging: In Jurafsky’s (2000) terms, “parts-of-speech tagging is the process of assigning a part-of-speech or other lexical class marker to each word in a corpus”. However, when highly agglutinative languages like Turkish is considered, we prefer using the term in a broad sense such as ‘the procedure for assigning pre-defined linguistic information to the pre-defined set of morphemes in a corpus’. This definition is also compatible with the standards of The European Commission's Expert Advisory Group on Language Engineering Standards (EAGLES) as described in Kahrel (1997). According to EAGLES guidelines, the most common two types of tagging are morphosyntactic annotation and syntactic annotation (Treebanks). Semantic annotation as in WordNet implementations is another type or level of tagging. In this study, all the terms *parts of speech tagging*, *morphosyntactic annotation* and *grammatical tagging* refer to the same procedure defined above which does not include any sentential, function-denoting, constituent or dependency tagging as done in Treebanks.

Finite-state transducer (FST): Formalism or method used for simulating, computerizing finite processes, rules of natural languages. Like all Finite-State Automata (FSA), a FST includes a limited number of states between an initial and a final state, and the transitions between these states. Unlike other FSAs, a FST includes both an input (words, morphemes) and an output (tags denoting linguistic information) within each state.

Token: Any sequence of letters followed and preceded by delimiters, such as space characters, tabs, carriage returns, apostrophes, digits or any punctuation marks.

Lemma: “the item which occurs at the beginning of a dictionary entry; more generally referred to as a headword” (Crystal, 2003). To be more specific, ‘dictionary entries without any pre-defined inflected or derived forms’. The term is used to cover all phonological alternates of a given root as in ‘*akıl, akl*’ in Turkish. Both roots belong to the same lemma ‘*akıl*’. However, ‘*akıllı*’ is not considered as a lemma in this study since affix +II is a pre-defined derivational affix and recognized by the transducer being not regarded as part of the lexicon. In this respect, the term is used more specifically than Crystal’s, which refers to traditional dictionaries rather than electronic dictionaries.

Limitations

Diachronic or etymological analyses are beyond the scope of this study. It is limited to the description of the design process of a finite-state transducer for Parts of Speech (POS) tagging of Turkish. Syntactic and morphological disambiguation and analysis of compound words or multi-word units are also beyond the scope of the study.

I. METHODOLOGY

In this study, methods of finite-state morphology are used. Theoretical considerations underlying the strategic and technical details of the methodology are discussed below.

Natural Language Processing studies including POS tagging can be regarded as works of Artificial Intelligence and thus as attempts to simulate human mind. Psycholinguistic arguments, in this respect, has an important role in the methodological choices of such studies including ours. The following binary oppositions illustrate some key concepts that shape the methodological details of the study.

a) Decomposition / Full-listing

Debate on whether our minds store and process the morphemes of a given word as a whole -full listing- or uses the linguistic knowledge about each morpheme separately -decomposition- was one of the key issues of both psycholinguistics and computational morphology. As argued in Hankamer (1989) in detail, a full-listing approach cannot be an adequate model of cognitive processes especially for agglutinative languages like Turkish. Although experiments in Gürel (1999) present that “some multimorphemic words that consist of frequent affixes are processed as fast as monomorphemic words”, this study relies upon the decompositional point of view methodologically since it is difficult to decide which affixations in Turkish let a whole-word lexical access and to what extent processing time can be regarded as evidence for whole-word processing. In this study, besides having a decompositional point of view, we also share, for practical reasons mostly, the arguments on lexical organization in Şehitoğlu & Bozşahin (1996), indicating that “bound morphemes are not” even “part of the lexicon”.

b) Affix-stripping / Root-driven

Another debate is on the direction of morphological parsing between root-driven (left-to-right) and affix-stripping (right-to-left) strategies. As stated in Hankamer (1989), “the set of suffixes determined by a stem is a finite set, whereas the set of stems determined by a suffix is always very large, and not necessarily even finite”. Thus, according to Hankamer, stripping an affix does not narrow the choices of stems and every time an affix is stripped, the parser should look for almost an infinite set of allowed stems which is actually a wasteful process. In addition, the ambiguities as the ones exemplified in Table 2 will require additional operations which is again at least not practical.

	Left-to-right	Right-to-left
Kayı ₁ s ₁	Kayı ₁ s ₁	Kayı ₁ + s ₁
Elma ₁ s ₁	Elma + s ₁	Elma + s ₁
Elma ₁ s ₁	Elma + s ₁	Elma + s ₁

Table 2. Possible decompositions with left-to-right and right-to-left processing.

c) Rule-based / Probabilistic

The primary distinction among approaches to POS tagging are often figured out as between unsupervised and supervised methods, according to the usage of distributional, contextual, syntactic, morphological or lexical rules or, in supervised methods, the tagging probabilities taken from a training corpus and computed by a variety of formulae. Through this dichotomy, we generalize all stochastic methods using probabilities, frequencies or statistics and the ones using only distributional, contextual or lexical rule specifications. This study, presupposing that there isn't evidence for the use of probabilities or other statistical information in human lexical processing, is based on rule-based methods not involving any training data or statistical techniques.

This study uses methods of finite-state morphology with an unsupervised, non-stochastic, decompositional, rule-based, root-driven approach.

I.1. Software Evaluation

Based on the methodological considerations discussed in the previous section, we have evaluated the finite-state compilers listed in Laitinen (2008).

Bearing in mind that “with existing taggers, automatic perfect tagging is not possible”, as Mihalcea (2003) showed in his performance analysis of mostly stochastic POS taggers, our primary criterion was the opportunities of rule declaring and semi-automatic or manual tagging that are provided. Besides, the capacity of processing large amount of textual corpora was another criterion.

In this respect, a graph-based corpus processor, Nooj, presented in Silberztein (2003) is chosen since it has both graphical tools for modeling, simulating the affixation of Turkish and the power to process large amount of texts simultaneously. In addition, the architecture of Nooj is pre-determined to be based on dictionaries as inputs to FSTs included, so again suitable for the root-driven approach of this study.

“Nooj is a development environment used to construct large-coverage formalized descriptions of natural languages, and apply them to large corpora, in real time.” It uses basically two resources for the annotation of textual input: “electronic dictionaries” and “grammars represented by organized sets of graphs” (Silberztein, 2005).

In brief, the dictionaries of Nooj allow assigning unlimited linguistic information to the entries and it can also handle multi-word units. The graphs allow the user construct a FST visually in a cascaded manner. Each graph can contain or refer to other graphs and this makes the modeling task simpler. The features and advantages of Nooj dictionaries and graphs will be discussed again in detail.

I.2. Data Collection

Data of the study are derived from the Turkish National Corpus (TNC) project held by the Linguistics Department at Mersin University and funded by the Scientific and Technological Research Council of Turkey.

I.2.1. Corpus

Sub-corpus for tokenization are extracted randomly from TNC. It included 100 text files including both fictional and non-fictional books. Number of word forms is computed as 3,323,853 by Nooj.

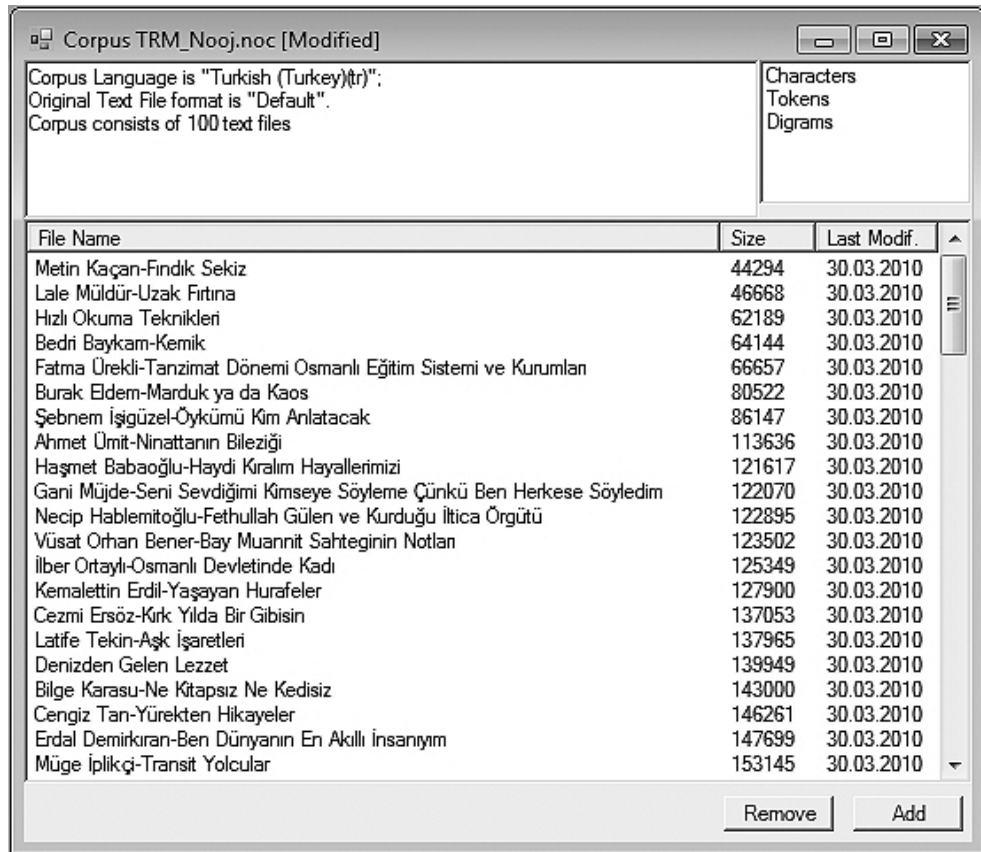


Figure 1. Corpus interface of Nooj

I.2.2. Tokenization

As presented in Figure 2, Nooj has a built-in tokenization tool that is accessible through the corpus interface in Figure 1 (see page 12). It also provides frequencies of each token.

Freq	Tokens
92920	bir
57959	ve
31966	da
31750	de
29874	bu
17037	için
16643	gibi
14124	Bu
13202	çok
12338	o
11750	daha
11137	ne
10443	sonra
9845	olarak
9786	kadar
9010	her
8974	ile
8153	ki
8112	Bir

Figure 2. Sample tokens ordered by frequency

As seen in the first and last lines of the tokenization pane in Figure 2, Nooj's tokenization is case-sensitive. This feature makes the given frequencies of tokens problematic but it helps extracting proper nouns, acronyms and abbreviations since they are all assumed to begin with upper-case letters. The exceptions were tokens like 'ntv, atv' that are acronyms in Turkish with lower-case initials. Another problem was homophonous tokens such as 'Deniz, Ümit' etc. being both proper nouns and common nouns. These tokens, as all other homophonous root forms with different POS, added to both noun and proper noun dictionaries. Finally, the task of filtering proper nouns, acronyms and abbreviations are done semi-automatically with the help of case-sensitive tokenization and Unicode sorting.

I.2.3. Lemmatization

Lemmatization, by definition, depends on which word forms are considered as lemmas. Since each analyzer has its own set of recognized affixes, the lemmas included in the dictionaries may differ. In this study, both inflectional and derivational affixes of Turkish are in the scope of analysis and, if not an archaic derivation or include a non-productive affix, only non-derived word forms or roots are considered as lemmas. In this respect, the dictionaries formed are root-dictionaries rather than stem-dictionaries.

Forming a stem dictionary of Turkish is problematic since some homophonous affixes such as +mA (negative & nominalizer), +(I)r, +AcAk (tense & nominalizer) etc. serve both in inflectional and derivational processes, and some derivational affixes such as +IA, +II, +IIk, +CI are very productive that they can easily be overused in forming lemmas.

After elicitation of mostly inflectional, non-homophonous, non-problematic affixations in a reverse ordered token list, lemmatization is done manually with the above stated considerations.

II. COMPONENTS OF THE TRANSDUCER

II.1. Overview of Nooj grammars

Nooj system includes three types of grammars that accept both graphical and textual rule specifications as presented in Figure 3. In M. Silberztein (p.c., September 20, 2009) terms, the difference between an “Inflection/Derivation” and “Productive Morphology” grammar is that the former is used to compile a dictionary with all inflected and derived forms of a root and thus adds the output to the dictionary with its annotations whereas the latter only annotates the given input text from the corpus by matching it with the corresponding lemma and category in the compiled dictionary and assigning them the linguistic information specified in the graphs.

The screenshot shows a window titled "Create a new Grammar" with a close button in the top right corner. The window is divided into three main sections:

- (1) Select Languages:** This section contains two side-by-side lists labeled "Input:" and "Output:". Both lists contain the following language codes: ro, ru, sa, sy, sp, sq, sr, sw, tm, tr. Each list has a scroll bar and a small menu icon at the bottom.
- (2) Optional Information:** This section contains:
 - A text input field for "Author".
 - A text input field for "Institution".
 - A checkbox labeled "Lock Grammar:" with two radio button options: "No display" (which is selected) and "Community".
 - A text input field for "Password".
- (3) Select Grammar Type:** This section contains:
 - Text: "Morphological Graphs recognize sequences of letters. Syntactic Graphs recognize sequences of words."
 - Two radio buttons: "graphical editor" (selected) and "rule editor".
 - Three buttons at the bottom: "Inflection & Derivation", "Productive Morphology", and "Syntax".

Figure 3. Nooj grammars

II.2. Architecture of Turkish Module

Our strategy, based on a similar approach to Bisazza (2009), is to use;

- i. Dictionaries (.nod files) for assigning a POS or Lexical Category tag to each lemma with all its morphophonemic alternations.
- ii. Inflection/Derivation grammars (.nof files) for specifying in-root phonemic alternation rules, in other words to compile the dictionaries including all possible alternations of the lemmas.
- iii. Productive morphology grammars (.nom files) for modeling both derivational and inflectional affixations of Turkish, namely for morphological tagging.
- iv. Syntactic grammars (.nog files) for contextual disambiguation of the annotated corpora.

Although M. Silberztein (p.c., September 20, 2009) notes that from a Hungarian dictionary of over 50,000 entries, researchers generated a .nod file that recognizes over 150 million word forms, the affix combinations of Turkish, considering the recursive nature of some affixes, are theoretically almost infinite. Hence, at least for the present version of the module, we have preferred .nom files for morphological tagging. In the following sections, the design process of each component of the module and the difficulties encountered will be presented.

II.3. Dictionaries

In Nooj formalism, dictionaries are formed in two formats. The Raw-Dictionaries consist of lemmas, related lexical categories and rule tags. Then, in our case, the Raw-Dictionaries are compiled through Nooj interface in Figure 4 with the triggered phonemic alternation rules specified in the Inflection/Derivation Grammar files.

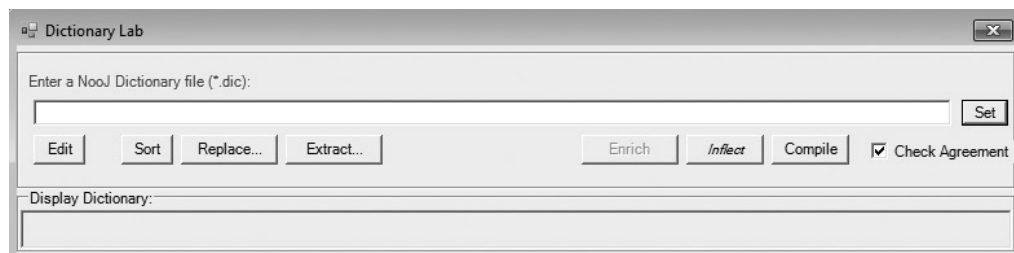


Figure 4. Dictionary compilation interface

During the compilation process, the operations declared in (.nof grammars) are implemented and, the Raw-Dictionaries (.dic files) and the variant lemmas formed by Inflection/Derivation grammars (.nof files) are combined as Dictionaries (.nod files).

The entries in (9-11) exemplify the formalism of Nooj raw-dictionaries.

(9) akıl,NN+FLX=drop+abstract

(10) güneş tutulması,NN+FLX=compound1

(11) abi, ağbi, ağabey, NN

The keyword “ +FLX= ” triggers the rules specified in .nof files. Although out the scope of the present study, Nooj can also handle multi-word units within a single dictionary format as in (10). As Silberztein (2005) notes, this is the major superiority of Nooj over its predecessor Intex. Users can add extra properties to the lemmas or subclasses to the Lexical Categories as in (9) and refer to them as constraints in the graphs. This feature is not used in this version since a tag set for subclasses of lexical categories is subject to further studies. Nooj also lets the user associate more than one word form to the same lemma that, in (11), the variant “abi” and “ağbi” both belong to the lemma “ağabey”. We have also reserved this feature for future releases of the module. Also, regular Optical Character Reading errors or character encoding problems may be handled by including the erroneous form of the lemma in the dictionary if not homophonous with another entry.

II.3.1. Lexical Categories

While forming the raw dictionaries, lemmas are tagged with the following POS

Tags listed in Table 3.

TAG	POS	EXAMPLE
<VB>	Verb	<i>git, gel, dur, bak, kal, sus, gör, dök</i>
<NN>	Noun	<i>gece, hava, renk, fark, dost, oyun</i>
<PN>	Pronoun	<i>bu, kendi, hepsi, herkes, kim, öteki</i>
<NB>	Number	<i>iki, üç, beş, sekiz</i>
<AJ>	Adjective	<i>mavi, yeni, düz, dürüst, zeki</i>
<AV>	Adverb	<i>acaba, asla, bazen</i>
<DET>	Determiner	<i>bu, şu, o</i>
<PP>	Postposition	<i>gibi, göre, için, kadar, karşı, rağmen</i>
<ITJ>	Interjection	<i>aferin, sağol, haydi, hoşçakal, lütfen</i>
<CJ>	Conjunction	<i>ama, çünkü, meğer, üstelik</i>
<ON>	Onomatopoeia	<i>takır, vızıl, gürül</i>
<NP>	Proper Noun	<i>Atatürk, Mersin, Ümit</i>
<AB>	Abbreviation Acronym	<i>TBMM, TDK</i>
<MI>	Affirmative particle	<i>mi, mı, mu, mü</i>

Table 3. Part-of-Speech Tags

As Trask (1999) notes, “over centuries, at least four different types of criteria have been proposed for identifying parts of speech” namely meaning, distribution, inflection and derivation and, as Haspelmath (2001) suggests, “there is universal agreement among linguists that language-particular word classes need to be defined on morphosyntactic grounds for each individual language”. With a similar point of view, we also relied upon distributional, inflectional and derivational features of a given lemma while assigning the appropriate lexical categories. Schachter & Shopen (2007) also proposes that “the primary criteria for parts-of-speech classification are grammatical, not semantic”.

In terms of lexical tagging, especially the Adjective/Noun, Conjunction/Adverb distinctions were challenging and required collaborative work with linguists.

The tags ‘ON’ for Onomatopoeic words and ‘MI’ for ‘Affirmative Particle’ are also used although not common in other languages. Onomatopoeic words have unique derivational features and the affirmative particle is also considered as a lexical category since it has its own affixation and must have a Category tag, for practical reasons in terms of computational analysis.

II.3.2. Phonemic Alternations

After the pre-tagging stage, lemmas are also annotated with the following tags which denote the in-root phonemic alternations listed in Table 4. As stated in (Göksel & Kerslake, 2005: 14), “certain of these changes are confined to specific lexical items, whereas others occur as part of a general phonological process in the language”. The changes confined to specific lexical items are subject to the pre-tagging stage of this study whereas others, being mostly “part of a general phonological process”, are handled through the graphs. The phonemic alternations highly irregular with a computational point of view are discussed below;

Aorist +(A)r and +(I)r

As Lewis (1967: 116) stated, for all monosyllabic verbs ending with a consonant -with 13 exceptions - , the aorist affix is “+(A)r”, whereas for all others it is “+(I)r”. The verbs requiring +(A)r affixation for the Aorist are tagged with rules “add_er” or “add_ar”.

Imperfective +(I)yor

When +(I)yor combines with stems ending with a vowel, the final –e, -a of the verb stems changes to -i,-i,-u,-ü (Lewis, 1967) (Göksel & Kerslake, 2005).

The in-root phonemic alternation caused by the affix ‘+(I)yor’ causes ambiguities as shown in (5-8).

(5) biliyor > bil+(I)yor / bile+(I)yor

(6) yıkıyor > yık+(I)yor / yıka+(I)yor

(7) atıyor > at+(I)yor / ata+(I)yor

(8) uyuyor > uy+(I)yor / uyu+(I)yor

Although the ambiguities in (5-8) are acceptable for native speakers of Turkish, formulating this operation as *substitution of a phoneme* causes some artificial ambiguities since the resulting word form with substituted final vowel can also be homophonous as in (9).

(9) bıçakla > bıçaklı

To avoid false-processing of the non-affixed forms of those homophones as Verbs, we preferred to declare the rule as *deletion* and include the buffer vowel (I) in the graphs, in order to reduce the number of artificial ambiguities. Underhill (1976: 112) also suggests this option in his note “If you prefer, you may simply learn that the suffix –Iyor causes a preceding vowel to drop”.

Other alternations such as the addition of buffer phoneme ‘(n)’ to pronominals “*bu, şu, o*” before case markers and the plural, as mentioned in Underhill (1976:90) as an in-stem variation, are all included in the graphs and not considered as in-root variations.

As Kornfilt (1997: 214) states “Turkish does not have internal morphophonemic alternations that ... are not conditioned by suffixation”. Thus, our tendency was to use the deletion operator where applicable and include the alternations in the graphs rather than the lexicon.

tag	rule	example
double	<D>	af > affi, zam > zamma
drop	<L><R>	akıl > aklını, fikir > fikrimin
dropsoften	<B2>b	kayıp > kaybına, kutup > kutbuna
compound1		anaokulu > anaokulları
compound2	<B2>	elyazısı > elyazıları, başağrısı > başağrıları
compound3	<B2>ç	ipucu > ipuçları
compound4	<B2>k	ayçiçeği > ayçiçekleri
soften_ch	c	ağaç > ağacı, süreç > süreci
soften_k	ğ	emek > emeği, diyalog > diyaloğu
soften_p	b	kitap > kitabı, mektup > mektubu
soften_t	d	cilt > cilde, dört > dördünü
soften_t_er	d + de	et > eder, git > gider
soften_t_ar	d + da	tat > tadar
softendouble	b<D>	tıp > tıbbın, muhip > muhibbi
change_an	<B2>an	ben > bana, sen > sana
change_i		iste > istiyor
change_ı		kapa > kapıyor
change_ü		özle > özlüyor
change_u		boşa > boşuyor
add_er	e	üz > üzer
add_ar	a	yap > yapar

Table 4. Rule specifications for in-root phonemic alternations in Turkish.

The rules in Table 4 follows the formalism of Nooj as in Table 5.

	delete last character, backspace	<L>	go left
<B2>	delete last two characters	<R>	go right
<D>	duplicate last character	+	OR

Table 5. Operators in Nooj grammar formalism (Silberztein, 2003: 92).

Any character following the mentioned operators is *added* to the resulting or existing form of the lemma.

The final format of the dictionary entries before compilation is shown in (2):

- (10) akıl,NN+FLX=drop
af,NN+FLX=double
kitap,NN+FLX=soften_p

The “ +FLX= ” operator declares that the following tag should lead to the predefined operation in the Nooj Inflection file (.nof). Finally, Nooj compiles the Raw-Dictionary and adds the variations of the lemma into the Nooj Dictionary file (.nod).

II.4. Graphs

After compiling the Dictionaries with Infection/Derivation grammars, Nooj needs Productive Grammars that will accept tokens as inputs and match them with the corresponding lemmas in the dictionary.

Since the transducer will not be used as a spell-checker or generator, all allomorphs are included in the same input of each state as the GENITIVE in Figure 5 represents. This strategy, although causing some false-annotations as in “*altu* > *al_VB+DI*”, is preferred in the present version of the module since including allomorphs in separate graphs would require pre-tagging of lemmas according to the phonological paradigm they belong to.

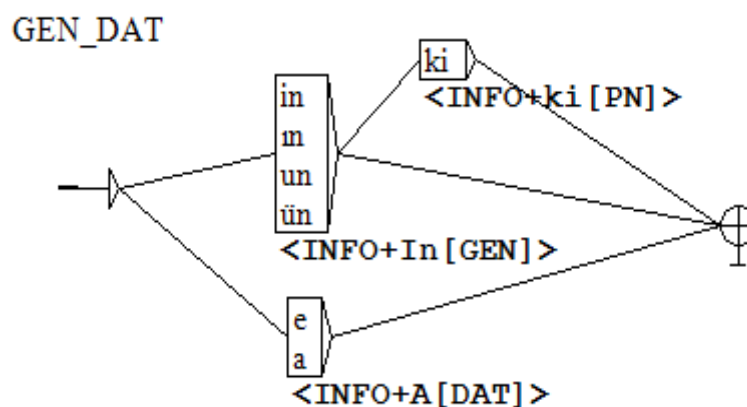


Figure 5. Sample graph representing allomorphs of an affix.

In Figure 5, the <INFO> stands for all previous annotations. All tags declared between the “ < ” and “ > ” characters form the output for the processed token. So the input “okulunki” returns an annotation as “okul,NN+In[GEN]+ki[PN]”.

Homonymy was again a challenge in forming the graphs because Turkish also includes homophonous suffixes like “In[GEN]” and “I+n[POSS]”. So, an input such as “okulun” returns two annotations as “okul,NN+In[GEN]” and “okul,NN+I+n[POSS]”. First one with a context such as “okulun kapısı” and the second “okulun nerede?”. As stated in the introduction, ambiguity resolution at the syntactic level is out the scope of this study.

All the subgraphs described in the following sections are designed in a single .nom file as presented in Figure 6.

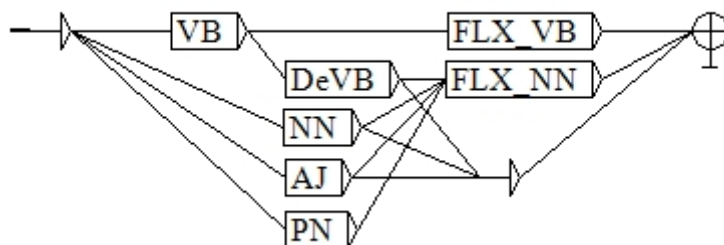


Figure 6. Main graph with the nominal and verbal paradigms.

The subgraphs VB, NN and AJ includes all derivational affixations forming stems of the given lexical category. Together with the deverbals in DeVB subgraph, they form inputs for the verbal and nominal inflectional paradigms, namely FLX_VB and FLX_NN.

II.4.1. Derivation

A detailed discussion of constraints governing derivation in Turkish can be found in Uzun (1992, 1993, 2008). However, the scope of this study is limited to the lexical categories of the derivational input and output word forms. Lexical subcategories related to derivational constraints are subject to further studies. We have organized the derivational graphs in Appendix D as exemplified in Figure 7. and Figure 8.

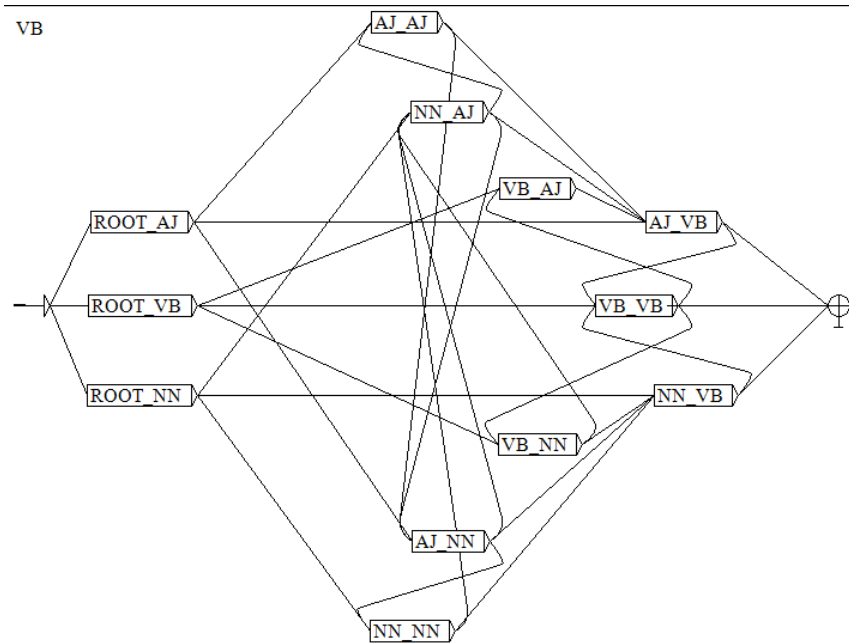


Figure 7. Sample derivational graph

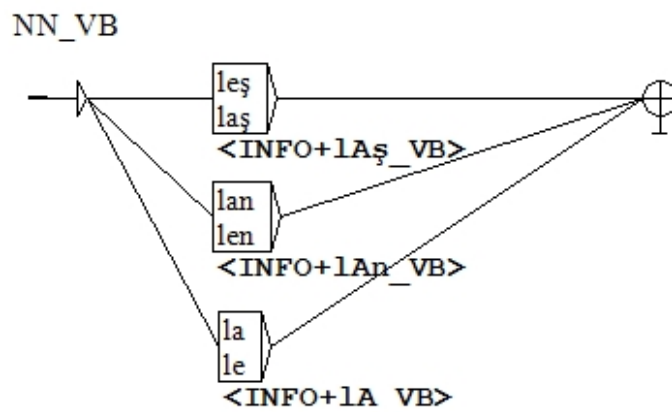


Figure 8. Sample derivational subgraph

The affixes in the derivational graphs and their distributional properties are listed in Uzun (1992).

II.4.2. Inflection (Nominal Paradigm)

Various finite-state models for Turkish nominal paradigm are provided by Oflazer, Göçmen & Bozşahin (1994a), Külekçi & Özkan (2001), Adalı & Eryiğit (2004), Makedonski (2005). Our strategy for Nominal Inflection is to form two subgraphs for stems ending in a vowel and in a consonant. By separating the graph into two, we have

reduced the number of homophonous affixes or buffer phonemes occurring in the same graph and thus causing artificial ambiguities. We have also included a RARE subgraph for irregular or archaic affixations in TNC.

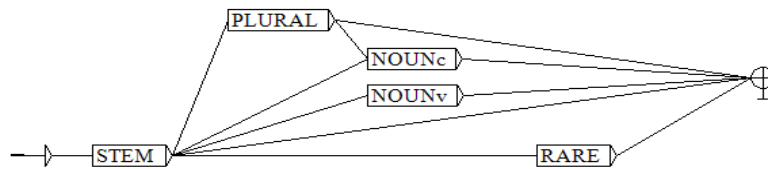


Figure 9. Main Graph for Nominal Inflection in Turkish.

Both graphs NOUN_c and NOUN_v in Figure 9. are the same except for some affixes such as buffer phoneme “(s)” for 3rd Person Possessive are included in NOUN_v as in “gece+si” but not in NOUN_c as in “asker+I”.

The STEM subgraph in Figure 10. includes the deverbals to avoid transition between verbal and nominal paradigms. Also the special case of pronominal “bu, şu, o” requiring a buffer phoneme “n’ is included in this subgraph.

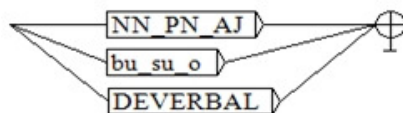


Figure 10. Stems for nominal inflection

The NOUN_c graph in Figure 11. presents the classification and transitions of nominal inflectional affixes.

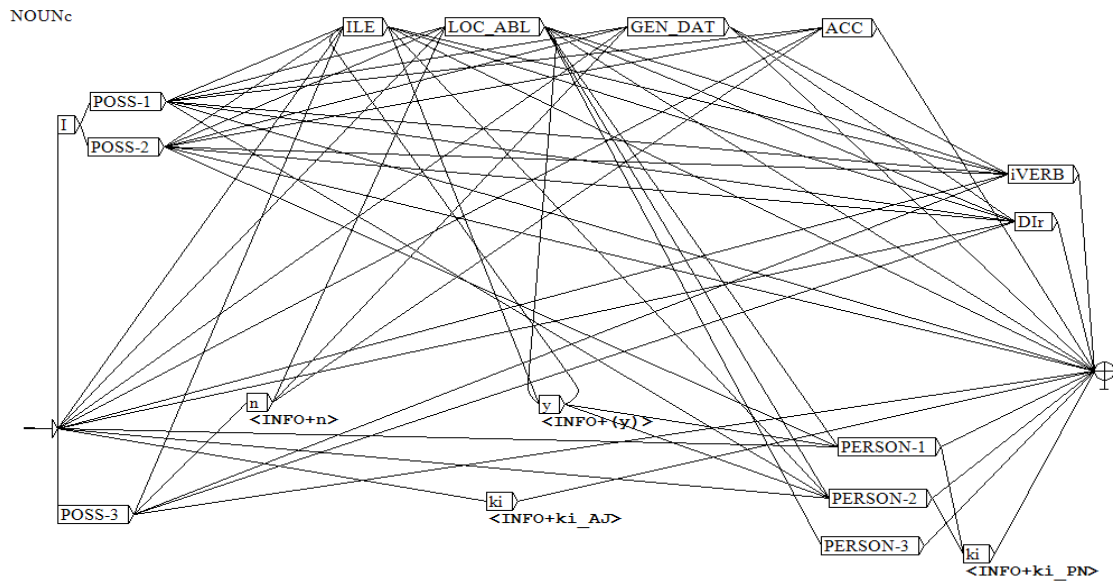


Figure 11. Nominal Inflection in Turkish

II.4.2. Inflection (Verbal Paradigm)

The morphotactics of verbal inflection in Turkish is modeled as in Figure 12. adopting the combinational features explained in Sebüktekin (1974) and arguments for Turkish Tense 1 and Tense 2 slots in Sezer (2001). As mentioned in that study, Tense 2 slot is only for;

- i. i-DI - indicating only witnessed past but not present perfect.
- ii. i-mIş - with an only inferential function but not present perfect.
- iii. i-sE - conditional with an indicative function but not subjunctive.
- iv. i-ken - adverbializer 'while'.

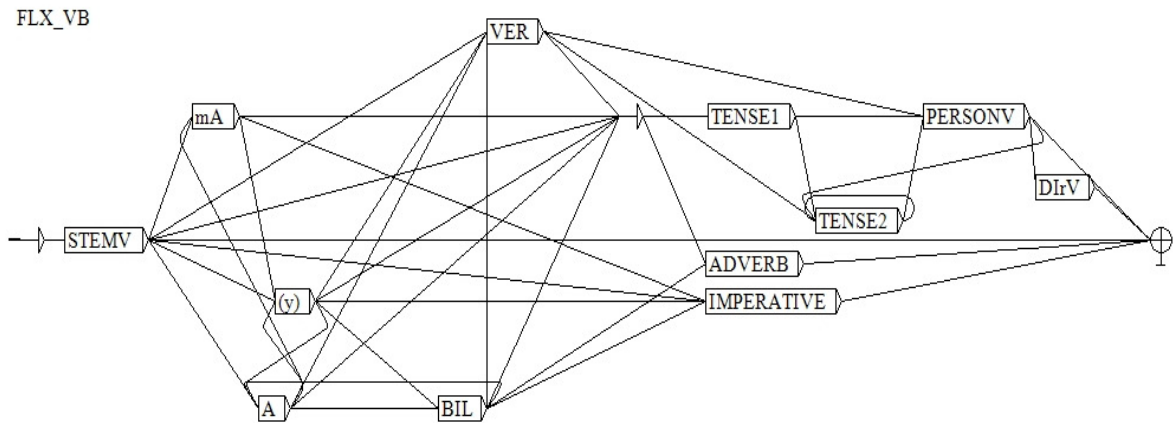


Figure 12. Verbal Inflection in Turkish

We haven't reserved a Tense 3 subgraph since it can be represented with a recursive transition from and to Tense 2 and can analyze the sequences as in (11);

(11) bil + ir + se + ymiş

III. IMPLEMENTATION AND TESTING

Nooj has a built-in analyzer and a concordancer for the implementation and testing of the modules. We have implemented the module, first on a test file as in Figure 13. and 14. and then the subcorpora of fiction and journalism from Turkish National Corpus Project.

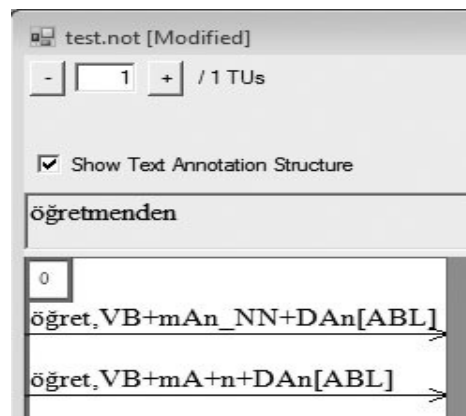


Figure 13. Sample Annotation 1.

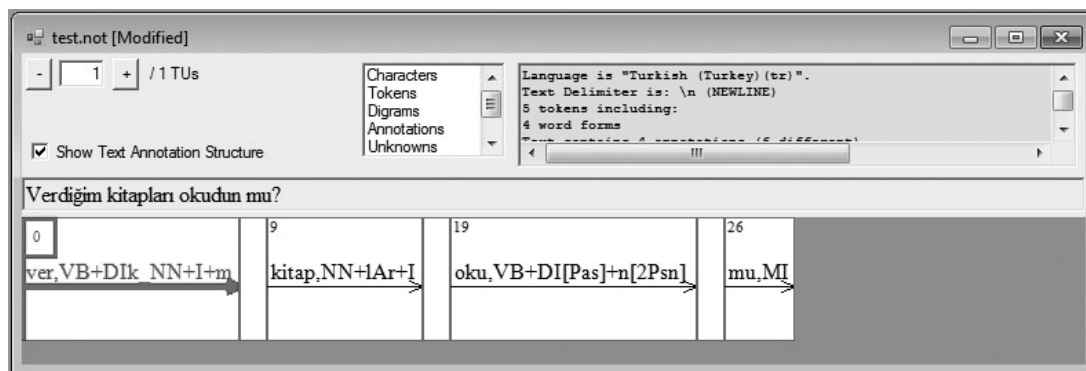


Figure 14. Sample Annotation 2.

Implementation and testing procedure involved the following steps;

- i. annotating text or corpora – in NooJ terminology *Linguistic Analysis*
- ii. checking the lists of *Unknowns* and *Ambiguities* to review the dictionaries and graphs.
- iii. concordancing in test-corpora – the *Locate* tool in NooJ terminology

Below are some concordances provided by the preceding queries. The operators ‘<’ and ‘>’ returns all affixed and bare forms of the given word. Without them, as in Figure 16., the results are only the bare forms. Space character is the ‘then’ operator in NooJ formalism and denotes sequential ordering of the given queries as in Figures 17, 18 and 19. Each affix is preceded with a ‘+’ operator and an affix search as in Figures 18 and 19 is also possible through the NooJ ‘Locate’ menu (Silberztein, 2003).

<oku,VB>

bir teorik yapının sunduğu yöntemlerle	okuyor	. Kullanım değeri'nin yerine çoktan
bağlantılan kurdurabiliyor muyuz "Raskolnikov"u	okurken	? Kurdumak zonunda mıyız? "Karanlığın Yüreği
durduran, bir romanın kısaltılmış versiyonunu	okumayı	kendine hakaret olarak algılayan "okur
okumayı kendine hakaret olarak algılayan "	okur	"un klasikten ne umduğunu, ne
zaman çarşıda olur yakından iştirdik.	Okuma	yazması olmayan biriydi. Nedenini bilemem
bir destan. Eve gidip iyice	okumak	ve anlamak isterdim. Okuduğumda bir
iyice okumak ve anlamak isterdim.	Okuduğumda	bir türlü pazar yerindeki o
Sony Reader PRS-500, elektronik kitap	okumanızı	. RSS haberlerini takip etmenizi ve
seyirlik unsura dönüştürme biçiminde uygulandığını	okuduklarımızdan	öğreniyoruz. Foucault'un "Hapishanenin Doğuğu
zaman çarşıda olur yakından iştirdik.	Okuma	yazması olmayan biriydi. Nedenini bilemem
bir destan. Eve gidip iyice	okumak	ve anlamak isterdim. Okuduğumda bir
iyice okumak ve anlamak isterdim.	Okuduğumda	bir türlü pazar yerindeki o
bir teorik yapının sunduğu yöntemlerle	okuyor	. Kullanım değeri'nin yerine çoktan
bağlantılan kurdurabiliyor muyuz "Raskolnikov"u	okurken	? Kurdumak zonunda mıyız? "Karanlığın Yüreği
durduran, bir romanın kısaltılmış versiyonunu	okumayı	kendine hakaret olarak algılayan "okur
okumayı kendine hakaret olarak algılayan "	okur	"un klasikten ne umduğunu, ne
elbette. Hele bir parça Osmanlıca	okuyabilen	varsa tanıdık çevrede, kesin söylüyorum
mi bileceksin? Bakanlıkta şu kadar	okumuş	kişi geceler boyu çalışmış didinmiş
para eder de bir rahmet	okur	bana diye. Ama ben satamam
uzak dumalı. Televizyon seyretmeyim, gazete	okumayım	. İyi de maaş zammına itiraz
ya canım! O kadar biyoloji	okumuşum	, kendi laboratuvarın davar. Olmadı mutfakta
seyirlik unsura dönüştürme biçiminde uygulandığını	okuduklarımızdan	öğreniyoruz. Foucault'un "Hapishanenin Doğuğu
belirtmişim. "Bir de bu kitabını	oku	!" dercesine Grass'ın kitabını verdiler
kez de ayrıntı abartısıyla karşılaştım.	Okumaya	ara verdim. Gerek şişirilmiş gözlemler
algılama zorluğu çekenler için. Üniversite	okumanın	maliyeti nedir? 22 yaşında okul bitecek
verilere ulaşamıyor. Ürün, saniyede 24 MB	okuyabiliyor	ve 10 MB yazabiliyor. 512MB, iGB

Figure 15. Sample Concordance 1.

oku

belirtmişim. "Bir de bu kitabını	oku	!" dercesine Grass'ın kitabını verdiler
bir ok saplanır. Kral bu	oku	çıkardığında sivri kanatları geriye doğru
dokuz hayvan kullanılıyor. Aklımdan geçenleri	oku	! Bazı beyin hücreleri başkalarının aklından
türlü kayıtsız kalınamayan; gergin yaya	oku	takmak gibi, onun hirsini da
çalışınca bana çok dua ettim	oku	diye, ama sen de çok

Figure 16. Sample Concordance 2.

<NN> <oku,VB>

çoklu bir teorik yapının sunduğu	yöntemlerle okuyor	. Kullanım değeri'nin yerine çoktan
sulan durduran, bir romanın kısaltılmış	versiyonunu okumayı	kendine hakaret olarak algılayan "okur
ürünü Sony Reader PRS-500, elektronik	kitap okumanızı	, RSS haberlerini takip etmenizi ve
çoklu bir teorik yapının sunduğu	yöntemlerle okuyor	. Kullanım değeri'nin yerine çoktan
sulan durduran, bir romanın kısaltılmış	versiyonunu okumayı	kendine hakaret olarak algılayan "okur
çok para eder de bir	rahmet okur	bana diye. Ama ben satamam
sıkıntıdan uzak dumalı. Televizyon seyretmeyim,	gazete okumayım	. İyi de maaş zammına itiraz
ölmüyordu ya canım! O kadar	biyoloji okumuşum	, kendi laboratuvarın davar. Olmadı mutfakta
olduğunu belirtmişim. "Bir de bu	kitabını oku	!" dercesine Grass'ın kitabını verdiler
şekilde algılama zorluğu çekenler için.	Üniversite okumanın	maliyeti nedir? 22 yaşında okul bitecek
ülke yoktur." Sonra da şu	gözlemleri okuyalım	: "Japonya, Güney Kore, Tayvan, Hong
hedefler açıldı Araştırmacılar Homo sapiens	kalitmini okumaya	öğreniyor: Doktorlar her kanser türüne
Bundan sonraki görevleri, sadece bu	bilgileri okuyabilmeyi	öğrenmek. Ne var ki bildik
Yabancılarla konuşmak için mi? Yabancı	yayınları okumak	için mi? Yoksa amacımız ikisini
üyeleri olan Fikret Adaman'ın	kaleminden okuyalım	: "BU'deki eğitimin İngilizce olmasının
de "uyanıklık" olduğu söylenir. Uyumayan,	mevdan okumalara	hazır, çevik, dayanaklarını bilen, gelebilecek
Bunun için çok ve çeşitli	kaynaklardan okumanız	gerekir. Çabuk okuma teknikleri arasında
insanlar bu ihtiyaçlarını yolculuk yaparak,	kitap okuyarak	, spor yaparak, aşk olarak, sosyal
yetişmekte olan kuşakların bilgisayar ve	internet okur	-okuyazarlığı oranı artacaktır. Bu da

Figure 17. Sample Concordance 3.

<VB+r[Aor]> <VB+mA+z[Aor]>

<p>metin" de "kültürsüz kumazlığın" eline sümüyor. Dışarı çıkınca yola adınızı metin" de "kültürsüz kumazlığın" eline oluşturdu. Devrimin ağırlıklı bir anlayış bilim adamları insanların bir yanlış ülke bir ekonomik kriz ortamına Kurumsal inceleme heyeti'nden onay solunum merkezleri baskılanır ve bebek olduğu kadar kaçınılması ve bebeğin gibi, tek eşliler çiftleşme sona verilen gezegenle ilgili bir çevredir. böyle bir çevreyle karşı karşıyayız. uranyum kullanarak yapılan fizyonun keşfedildiğini hekimler hastanın gözlerinde bu isteği hekimler olduğuna göre, promosyon da Onu da kansı Klytemnestra eve Amerika kaynaklı siteye bulaşmış durumda. e-kitabımı okuyabiliyim. Bu da türünden yararlanacak. Kamera rasपालayan sinyali aileye konuk olur, ama fırsatını asla sokmayınız!" Hani çoğumuz banyodan göre isim vermek onların isimlerine gibi, tek eşliler çiftleşme sona verilen gezegenle ilgili bir çevredir. böyle bir çevreyle karşı karşıyayız. en kolay yolu bu. Servise</p>	<p>geçer geçmez atar atmaz geçer geçmez ister istemez bulur bulmaz girer girmez alır almaz doğar doğmaz doğar doğmaz erer eremez Doğar doğmaz Doğar doğmaz duyar duymaz sezer sezmez ister istemez döner dönmez Bulaşır bulaşmaz ister istemez iletir iletmez bulur bulmaz çıkar çıkmaz bakar bakmaz erer eremez Doğar doğmaz Doğar doğmaz biner binmez</p>	<p>bir şekilde "toplumsallaşabiliyor", ama nasıl cenk başlıyor. Çocuklarımız da bu bir şekilde "toplumsallaşabiliyor", ama nasıl beden yapının dengesini sağlama sorununu gülmediklerine, ancak püf noktasını yakalayıp toplumdaki "günah keçisi" olmaya başlarlar donör araştırmalarına başlayacaklarını bildirdiler. Böyle ağlayamaz. Bu da oksijensiz kalmasına annesinin sütü ile beslenmesi gereklidir yeni bir eşin peşine düştüler böyle bir çevreyle karşı karşıyayız , belki bütün organizmalarda, gelişmiş canlılarda , bu elementin zincir reaksiyonunu yaratmakta onlardan önce davranıp "sizin bir hekimlere yöneliyor Ne yazık ki banyoda işleyerek öldürüyor. Aşıl (Achilleus sistemi) çökerten ve böylece en özel okuyucu cihazları gündeme getiriyor , en kumaz gangster bile içgüdüsel arkadaşları Sina'nın omuzuna dönüverir kulak temizleme çubuklarına saldırmaz ya bazı özelliklerinin bilinmesine olanak tanıdığı yeni bir eşin peşine düştüler böyle bir çevreyle karşı karşıyayız , belki bütün organizmalarda, gelişmiş canlılarda bizi kulak tımalayıcı bir kementçe</p>
--	---	---

Figure 18. Sample Concordance 4.

<VB+Ip[AV]> <dur>

<p>hep iyiyi yaratan ruh ile tanımına kitlenmiş, hatta "mitosa" geri kendi iktidar temsilcileri kadrolarına "çelişkiler" sarsılmaz temel eserler olarak "okura" Kum fırtınalarında tutunamayıp top gibi tıpkı şemsiye gibi, bizimle koşarlar. Toplam 184 kuş türü başınızın üstünde hep iyiyi yaratan ruh ile tanımına kitlenmiş, hatta "mitosa" geri kendi iktidar temsilcileri kadrolarına "çelişkiler" sarsılmaz temel eserler olarak "okura" kimse olmasa da 6 ay boyunca körfezine yuvarlanabilir. Düşünün bir, dünyanın birden fazla insan-öncesi türü ve iri gövdesiye tramvayın önünü çevresindeki herkes her çeşit konuyu dansına devam etti. Biz hâlâ bir de o hız yapınca biliyorsunuz, ne kamçılar yemiştik ki Bey bugünlerde pek hiddetli. Kızıp olmadığı için konuşmayı da unuttu, sanki. Kültür ve Turizm Bakanlığı, Bakanlığı, durup dururken -belki de sinemaya ve Lydu'ya tutkusunun</p>	<p>karşılaşıp duruyor dönüp duran üretip dururken sunup dururken durduklandıran Yorulup durunca dolaşıp durur karşılaşıp duruyor dönüp duran üretip dururken sunup dururken dönüp durabilecek dönüp durması dolanıp duruyordu kesip durmasını tartışıp dururken tartışıp duruyorduk ötüp duran şahlanıp dumuştuk kızıp duruyor susup duruyor durup dururken durup dururken çatışıp durması</p>	<p>. Tarihin (politik ekonominin) diyalektik süreci (Aydınlanmanın Diyalektiği) bir "akıl" olduğu başka bir anlam taşıyor. Ve , tahayyüllerimizdeki muhatap kitle nasıl bir çocukları gülmekten kırır geçirirler. Bakmayın onlar da gölgemizle beraber dururlar : Beyaz akbaba, tuma, balık kartalı . Tarihin (politik ekonominin) diyalektik süreci (Aydınlanmanın Diyalektiği) bir "akıl" olduğu başka bir anlam taşıyor. Ve , tahayyüllerimizdeki muhatap kitle nasıl bir . Mühendisler ayrıca güvenlik önlemlerini de , gezegen ve yıldız dengelerinin sürdürülmesi . İşte insanlar şimdi bunları içinden sağlayacaktır. Adam ölecek ama 5 kişinin , kendi kendine bir şarkı mıldanıp , "Yok efendim IMF geri adım aletten var. Hemen başlıyor çık . Binicilerimiz ustalaşmaya kadar epeyce koşup , herhalde haklıdır. İyi de bizim . Ne zaman ki kendinden birilerini -belki de durup dururken değil değil, siyasal yandaşlarından gelen uyanlara . Çiftin hep kritik durumda olan</p>
--	--	---

Figure 19. Sample Concordance 5

CONCLUSION

Summary

In this study, the nominal and verbal inflectional paradigms and derivational affixation of Turkish is modeled through finite-state transducers with an unsupervised, decompositional, root-driven, graph-based approach. A corpus-driven electronic dictionary of Turkish including lexical categories is formed.

Results of the Study

This study showed that a finite-state transducer automaton for parts of speech tagging of Turkish can be designed in a root-to-affix approach and discussed the difficulties specific to Turkish in the lexicon and transducer design.

It is also shown that Turkish morphology can be simulated with the approach of the study. However, this study also showed the need for further studies, like the constraints on the derivational processes of Turkish and common ambiguities with their context dependent solutions.

Recommendations

The challenges in the design procedure of the FST for Turkish POS tagging showed that further studies concerned with the architecture of electronic dictionaries for Turkish are needed. The linguistic information that should be involved in the dictionary of Turkish for each specific lexical category is subject to those studies. With the adequate features added to each lemma, the analysis will return less ambiguities and be more accurate.

Ambiguity in Turkish needs more detailed documentation. Homophonous roots, affixes and root-affix combinations need to be listed and classified. Their contextual information is also needed for disambiguation.

Disambiguation in Turkish will be possible upon the findings of studies focused on the Natural Language Processing of Turkish like the one here. Rule-based disambiguation can be done through the NooJ Syntactic Grammars and the exported annotated corpora may then again be disambiguated stochastically.

Compound words and reduplications in Turkish are topics that are studied by various researchers. A corpus-driven database for those multi-word units will form the necessary dictionaries and add new findings to the available data sets.

The transducer developed in this study may be improved and be used as a generator to identify constraints governing the affixation of Turkish.

Parallel or bilingual corpora, especially including English and Turkish texts, will lead to new findings on the problematic areas specific to Turkish learners of English. Syllabus design and material development for Turkish learners will be more specific by the help of mentioned studies.

Parallel corpora will also lead to improvements in material development for Turkish as a Foreign Language.

REFERENCES

- Adalı, E., & Cebirođlu, G. (2002). Sözlüksüz köke ulaşma yöntemi. In *Proceedings of the 19th TBD Bilişim Kurultayı* (pp. 155-160). İstanbul.
- Adalı, E., & Eryiđit, G. (2004). An affix stripping morphological analyzer for Turkish. In *Proceedings of the IASTED International Conference ARTIFICIAL INTELLIGENCE AND APPLICATIONS*. Innsbruck, Austria.
- Akın, M. D., & Akın, A. A. (2007). Türk dilleri için açık kaynaklı doğal dil işleme kütüphanesi: ZEMBEREK. *Elektrik Mühendisliđi*, 431, 38.
- Bisazza, A. (2009). *Designing a Nooj module for Turkish*. Paper presented at the Nooj Conference 2009.
- Crystal, D. (2003). *A dictionary of linguistics & phonetics*. Malden, MA: Blackwell Pub.
- Çiçekli, İ., & Temizsoy, M. (1997). Automatic creation of a morphological processor in logic programming environment. In *Proceedings of the 5th International Conference on the Practical Application of Prolog (PAP'97)*. London, UK.
- Çilden, E. K. (2006). Stemming Turkish words using Snowball. Retrieved 12 March 2010, from <http://snowball.tartarus.org/algorithms/turkish/stemmer.html>
- Göksel, A., & Kerslake, C. (2005). *Turkish: A comprehensive grammar*. London & New York: Routledge.
- Güngör, T. (2003). *Lexical and morphological statistics for Turkish*. Paper presented at the International Twelfth Turkish Symposium on Artificial Intelligence and Neural Networks.
- Gürel, A. (1999). Decomposition: to what extent? The case of Turkish. *Brain and language*, 68(1-2), 1-15.

- Hankamer, J. (1989). Morphological parsing and the lexicon. In W. Marslen-Wilson (Ed.), *Lexical representation and process* (pp. 392-408): MIT Press.
- Haspelmath, M. (2001). Word classes and parts of speech. In P. B. Baltes & N. J. Smelser (Eds.), *International encyclopedia of the social and behavioral sciences* (pp. 16538–16545). Amsterdam, The Netherlands: Pergamon.
- Joshi, A., & Hopely, P. (1997). A parser from antiquity. *Natural language engineering*, 2(4).
- Jurafsky, D., & Martin, J. H. (2000). *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition*: Prentice Hall.
- Kahrel, P., Barnett, R., & Leech, G. N. (1997). Towards cross-linguistic standards or guidelines for the annotation of corpora. In R. Garside, G. N. Leech & T. McEnery (Eds.), *Corpus annotation : linguistic information from computer text corpora*. London; New York: Longman.
- Karttunen, L. (2001). Applications of finite-state transducers in natural-language processing. In A. P. Sheng Yu (Ed.), *Implementation and application of automata*. Berlin: Springer.
- Karttunen, L., & Beesley, K. R. (2005). Twenty-five years of finite-state morphology. In *Inquiries Into Words, a Festschrift for Kimmo Koskenniemi on his 60th Birthday, CSLI Studies in Computational Linguistics*. Stanford CA: CSLI.
- Kornfilt, J. (1997). *Turkish*. London; New York: Routledge.
- Külekçi, M. O., & Özkan, M. (2001). Turkish word segmentation using morphological analyzer. In *Proceedings of EuroSpeech*. Aalborg, Denmark.

- Laitinen, H.-R. (2008). FsmReg: A registry of finite-state technology. Retrieved 10 March 2010, from <https://kitwiki.csc.fi/twiki/bin/view/KitWiki/FsmReg>
- Lewis, G. L. (1967). *Turkish grammar*. Oxford: Oxford University Press.
- Makedonski, P. (2005). *Finite state morphology: the Turkish nominal paradigm*. Universitat Tübingen, Tübingen.
- Mihalcea, R. (2003). Performance analysis of a part of speech tagging task. In A. Gelbukh (Ed.), *Computational linguistics and intelligent text processing Proceedings of 4th International Conference, CICLing 2003, Mexico City, Mexico, February 16-22, 2003*. Berlin [etc.]: SpringerLink.
- Oflazer, K., Göçmen, E., & Bozşahin, C. (1994a). *An outline of Turkish morphology*: Technical Report, Middle East Technical University.
- Oflazer, K. (1994b). Two-level description of Turkish morphology. *Literary and linguistic computing*, 9(2), 137-148.
- Roark, B., & Sproat, R. W. (2007). *Computational approaches to morphology and syntax*. Oxford; New York: Oxford University Press.
- Sak, H., Güngör, T., & Saraçlar, M. (2008). Turkish language resources: morphological parser, morphological disambiguator and web corpus. In *Advances in natural language processing* (Vol. 5221/2008, pp. 417-427): Springer Berlin / Heidelberg.
- Sak, H., Güngör, T., & Saraçlar, M. (2009). A stochastic finite-state morphological parser for Turkish. In *Proceedings of the ACL-IJCNLP 2009 Conference short papers*. Suntec, Singapore: Association for computational linguistics.
- Schachter, P., & Shopen, T. (2007). Parts-of-speech systems. In T. Shopen (Ed.), *Language typology and syntactic description : Clause structure*. Leiden: CUP.

- Sebüktekin, H. I. (1974). Morphotactics of Turkish verb suffixation. *Boğaziçi Üniversitesi Dergisi*, 2, 87-116.
- Sever, H., & Bitirim, Y. (2003). FindStem: Analysis and evaluation of a turkish stemming algorithm. In *10th International Symposium on string processing and information retrieval (SPIRE'03), Manaus, Brazil, October 8-10, 2003. Lecture notes in computer science (LNCS)* (pp. 238-251): Springer.
- Sezer, E. (2001). Finite inflection in Turkish. In E. E. Taylan (Ed.), *The verb in Turkish* (pp. 1-47). Amsterdam: John Benjamins Publishing.
- Silberztein, M. (2003). Nooj manual. Retrieved 10 March 2010, from <http://www.nooj4nlp.net>
- Silberztein, M. (2005). Nooj's dictionaries. In *Proceedings of the 2nd language and technology conference*: Poznan University.
- Sproat, R. (1992). *Morphology and computation*. London: MIT.
- Şehitoğlu, O., & Bozşahin, C. (1996). Morphological productivity in the lexicon. In *Proceedings of the ACL SIGLEX workshop at Santa Cruz*.
- Trask, R. L. (1999). Parts of speech. In K. Brown & J. Miller (Eds.), *Concise encyclopedia of grammatical categories* (pp. 278-284). Oxford: Elsevier.
- Underhill, R. (1976). *Turkish grammar*. Cambridge, Mass.: MIT Press.
- Uzun, E., Uzun, L., Aksan, M., & Aksan, Y. (1992). *Türkiye Türkçesinin türetim ekleri: Bir döküm denemesi [Derivational suffixes of Turkish: A morpheme inventory]*. Ankara: Şirin.
- Uzun, E. (1993). *Türkiye Türkçesinde sözlüksel yapı: Bir eleştirel çözümleme*. Ankara Üniversitesi, Ankara.

Uzun, E. (2008). Türetim eklerinin türetkenliğini ölçme önerileri üzerine. In Y. Çotuksöken & N. Yalçın (Eds.), *XX. Dilbilim kurultayı bildirileri 12-13 Mayıs 2006*. İstanbul: Maltepe Üniversitesi.

APPENDIX A
Affix tagset (derivational)

1	mA_NN	VB_NN
2	AlgA_NN	VB_NN
3	KA_NN	VB_NN
4	(A)ç_NN	VB_NN
5	mAç_NN	VB_NN
6	gIç_NN	VB_NN
7	(I)nç_NN	VB_NN
8	KAç_NN	VB_NN
9	sI_NN	VB_NN
10	(y)+IcI_NN	VB_NN
11	KI_NN	VB_NN
12	(I)ntI_NN	VB_NN
13	tI_NN	VB_NN
14	(A)nAk_NN	VB_NN
15	(ş)+Ak_NN	VB_NN
16	Am_NN	VB_NN
17	(y)+(I)m_NN	VB_NN
18	mAn_NN	VB_NN
19	KAn_NN	VB_NN
20	KIn_NN	VB_NN
21	(I)t_NN	VB_NN
22	(A)y_NN	VB_NN
23	tay_NN	VB_NN
24	iye_NN	NN_NN
25	Ar_NN	NN_NN
26	GAr_NN	NN_NN
27	at_NN	NN_NN
28	keş_NN	NN_NN
29	mAn_NN	NN_NN
30	baz_NN	NN_NN
31	lak_NN	NN_NN
32	ist_NN	NN_NN
33	DAş_NN	NN_NN
34	DAr_NN	NN_NN
35	dIz_NN	NN_NN
36	CAk_NN	NN_NN
37	dIrIk_NN	NN_NN
38	cAğIz_NN	NN_NN
39	tay_NN	NN_NN
40	CIk_NN	NN_NN
41	CI_NN	NN_NN
42	IIk_NN	NN_NN

43	zade_NN	AJ_NN
44	yet_NN	AJ_NN
45	CI_NN	AJ_NN
46	IIk_NN	AJ_NN
47	AğAn_AJ	VB_AJ
48	KAn_AJ	VB_AJ
49	AcAn_AJ	VB_AJ
50	III_AJ	VB_AJ
51	KIn_AJ	VB_AJ
52	II_AJ	VB_AJ
53	AI_AJ	VB_AJ
54	(I)mtrak_AJ	AJ_AJ
55	kar_AJ	AJ_AJ
56	CA_AJ	AJ_AJ
57	(ş)Ar_AJ	AJ_AJ
58	IAk_AJ	AJ_AJ
59	(I)ncI_AJ	AJ_AJ
60	Acık_AJ	AJ_AJ
61	rAk_AJ	AJ_AJ
62	IArcA_AJ	AJ_AJ
63	(I)msI_AJ	AJ_AJ
64	AI_AJ	NN_AJ
65	sII_AJ	NN_AJ
66	II_AJ	NN_AJ
67	sIz_AJ	NN_AJ
68	sAI_AJ	NN_AJ
69	(I)msI_AJ	NN_AJ
70	cAI_AJ	NN_AJ
71	CII_AJ	NN_AJ
72	sI_AJ	NN_AJ
73	DAş_AJ	NN_AJ
74	IAş_VB	NN_VB
75	IAn_VB	NN_VB
76	IA_VB	NN_VB
77	sA_VB	AJ_VB
78	IAş_VB	AJ_VB
79	(A)I_VB	AJ_VB

80	t_VB	VB_VB
81	(I)r_VB	VB_VB
82	DIr_VB	VB_VB
83	(I)I_VB	VB_VB
84	(I)n_VB	VB_VB
85	(I)ş_VB	VB_VB
86	AlA_VB	VB_VB

APPENDIX B**Affix tagset (inflectional / nominal paradigm)**

1	lAr	number/person
2	I	buffer phoneme
3	n	buffer phoneme
4	(y)	buffer phoneme
5	(s)	buffer phoneme
6	I	case
7	In[GEN]	case
8	A[DAT]	case
9	DA[LOC]	case
10	DAn[ABL]	case
11	ile	case
12	Im[1Psn]	person_copula
13	Iz[1Ppl]	person_copula
14	sIn[2Psn]	person_copula
15	sInIz[2Ppl]	person_copula
16	m[Poss]	possessive
17	mIz[Poss]	possessive
18	n	possessive
19	nIz[Poss]	possessive
20	I	possessive

35	mAk_NN	nominal
36	AcAk_NN	nominal
37	mA_NN	nominal
38	sI_NN	nominal
39	DIk_NN	nominal
40	An_AJ	adjectival
41	ki_AJ	adjectival
42	ki_PN	pronominal
43	cA_AV	adverbial
44	cAsInA_AV	adverbial
45	ken_AV	adverbial
46	sA_AV	adverbial

21	i	verb
22	DIr	copula
23	DI[Past]	copula
24	mIş[Perf]	copula
25	m[1Psn]	person
26	n[2Psn]	person
27	k[1Ppl]	person
28	nIz[2Ppl]	person
29	[3Psn]	person
30	lAr[3Ppl]	person
31	sInIz[2Ppl]	person
32	sIn[2Psn]	person
33	Iz[1Ppl]	person
34	Im[1Psn]	person

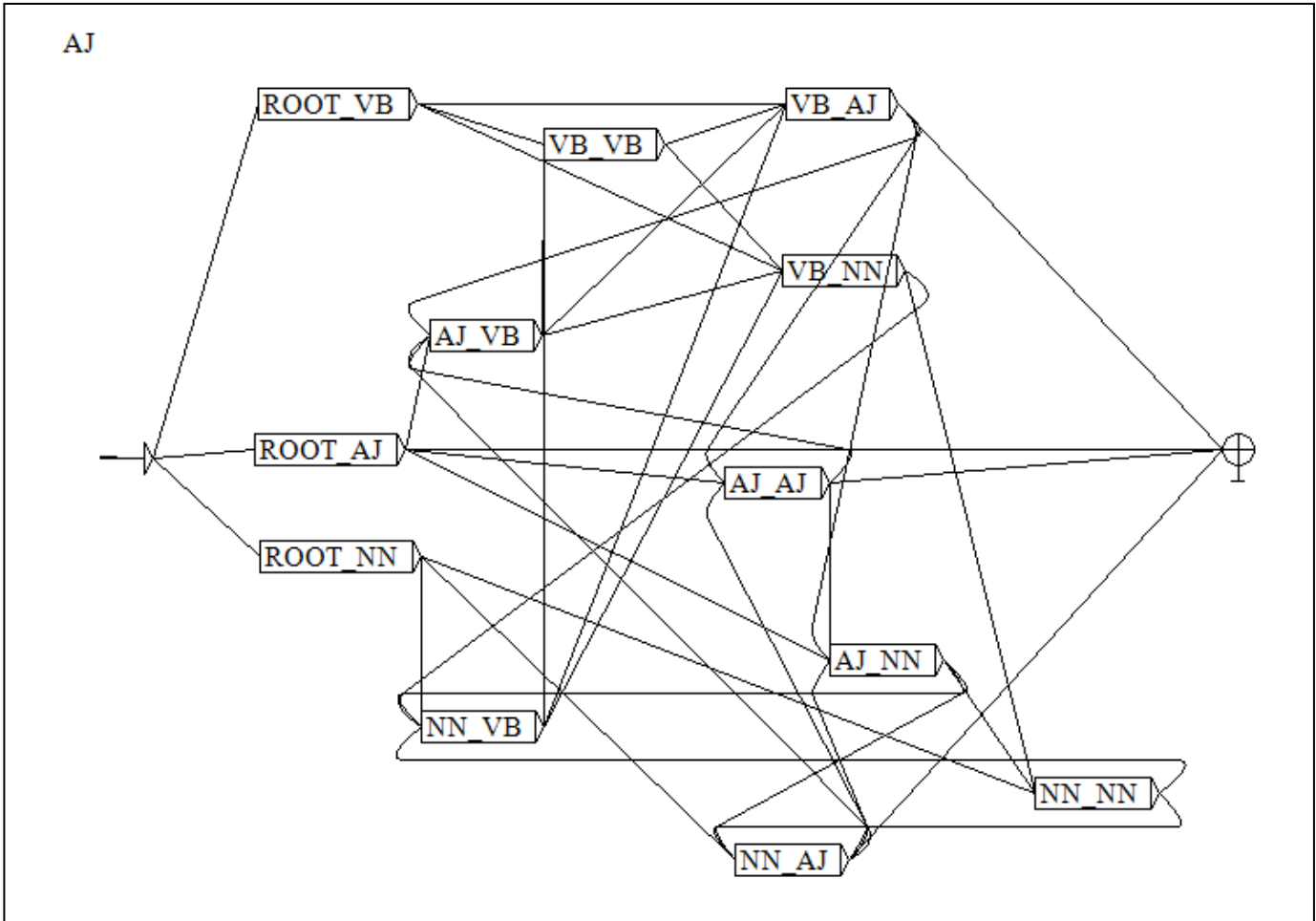
APPENDIX C

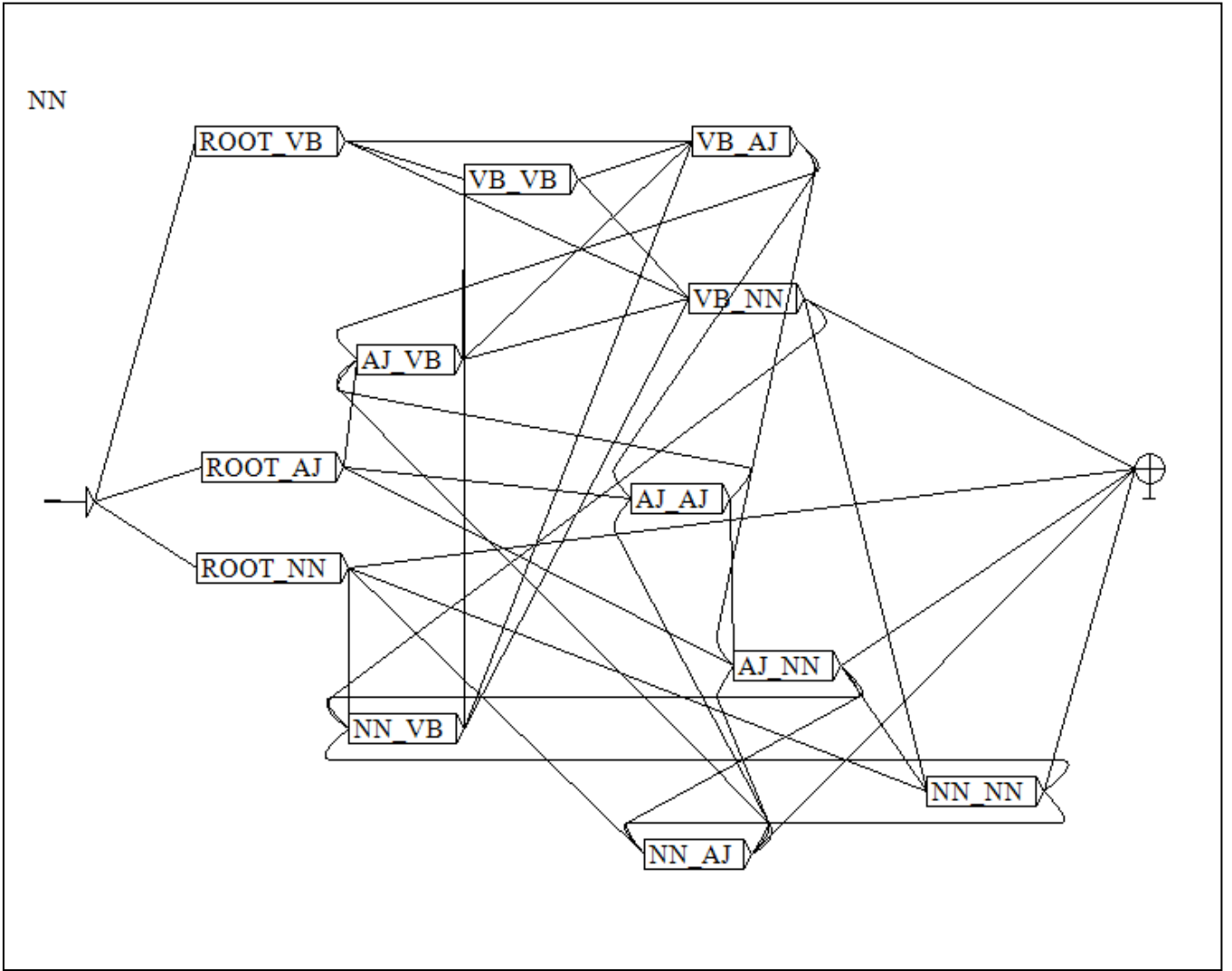
Affix tagset (inflectional / verbal paradigm)

1	(y)	buffer phoneme
2	(I)	buffer phoneme
3	A	buffer phoneme
4	yor	imperfective
5	bil	ability
6	dur	auxiliary verb
7	gel	auxiliary verb
8	gör	auxiliary verb
9	yaz	auxiliary verb
10	kal	auxiliary verb
11	ver	auxiliary verb
12	koy	auxiliary verb
13	AyIm[IMP]	imperative
14	sIn[IMP]	imperative
15	Allm[IMP]	imperative
16	In(Iz)[IMP]	imperative
17	sInIAr[IMP]	imperative
18	mA	negative
19	ik[1Ppl]	person
20	k[1Ppl]	person
21	(I)z[1Ppl]	person
22	(I)m[1Psn]	person
23	nIz[2Ppl]	person
24	sInIz[2Ppl]	person
25	sIn[2Psn]	person
26	n[2Psn]	person
27	lAr[3Ppl]	person
28	r[Aor]	aorist
29	z[Aor]	aorist
30	mAktA[Cont]	imperfective
31	AcAk[Futr]	future
32	mAll[Necc]	necessity
33	DI[Pas]	past / perfective
34	mIş[Per]	referential / perfective
35	i	verb
36	DIr(P)	possibility

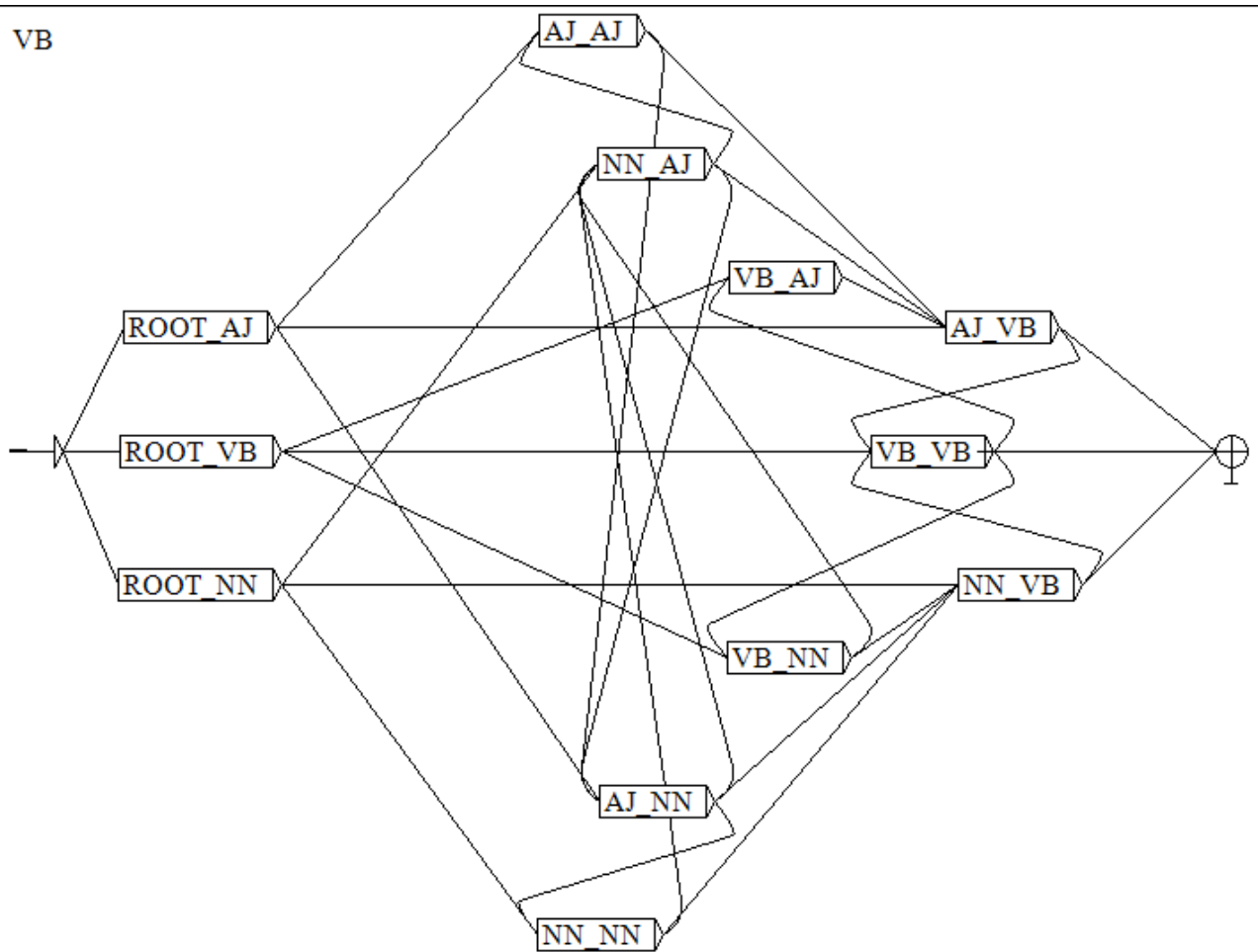
37	All_AV	adverbial
38	ArAk_AV	adverbial
39	ArAktAn_AV	adverbial
40	AsIyA_AV	adverbial
41	DIkçA_AV	adverbial
42	IncA_AV	adverbial
43	Ip_AV	adverbial
44	mAdAn_AV	adverbial
45	mAksIzIn_AV	adverbial
46	ken_AV	adverbial
47	sA_AV	adverbial
48	cAsInA_AV	adverbial

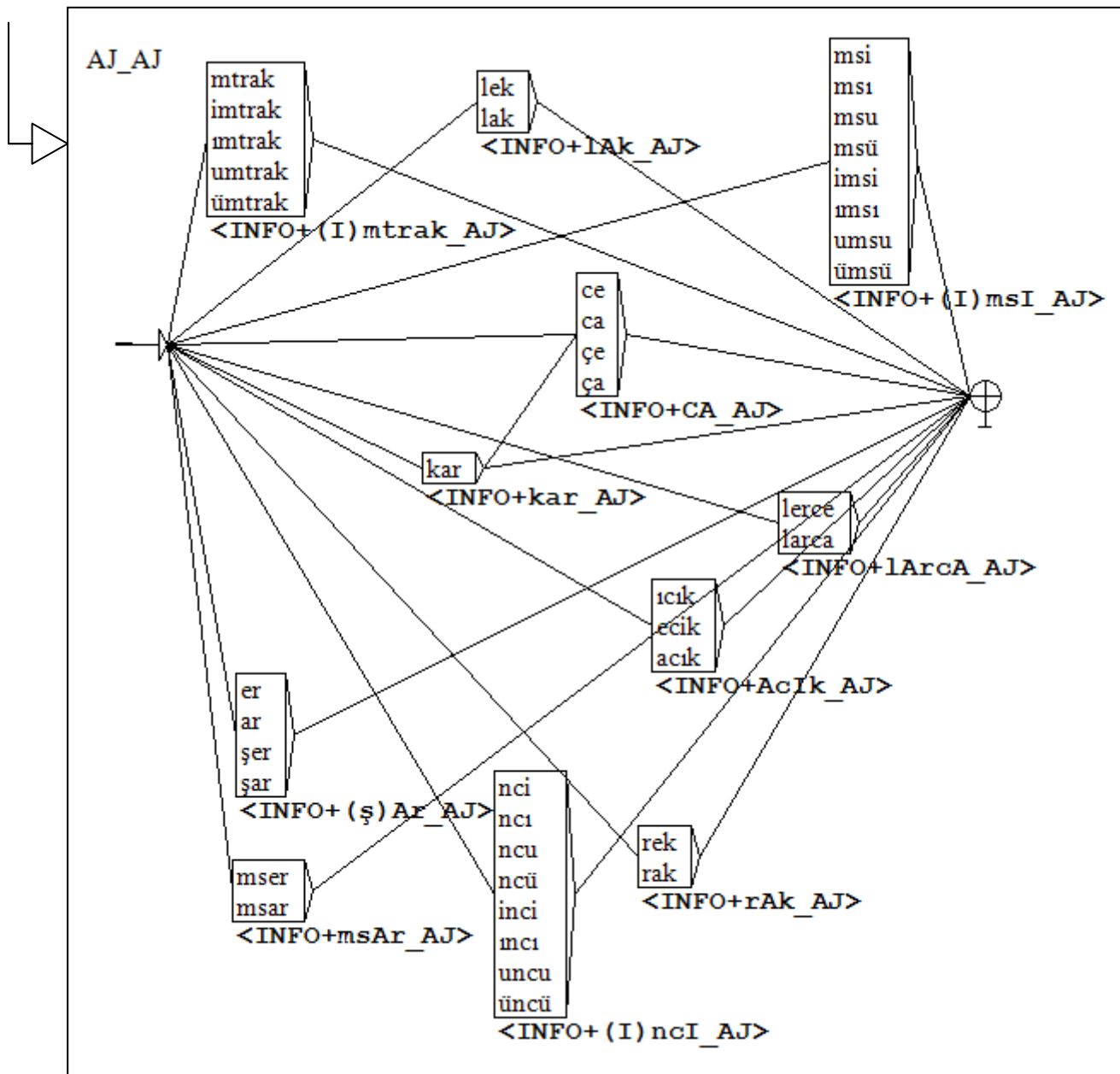
APPENDIX D
Turkish Derivational Affixation.

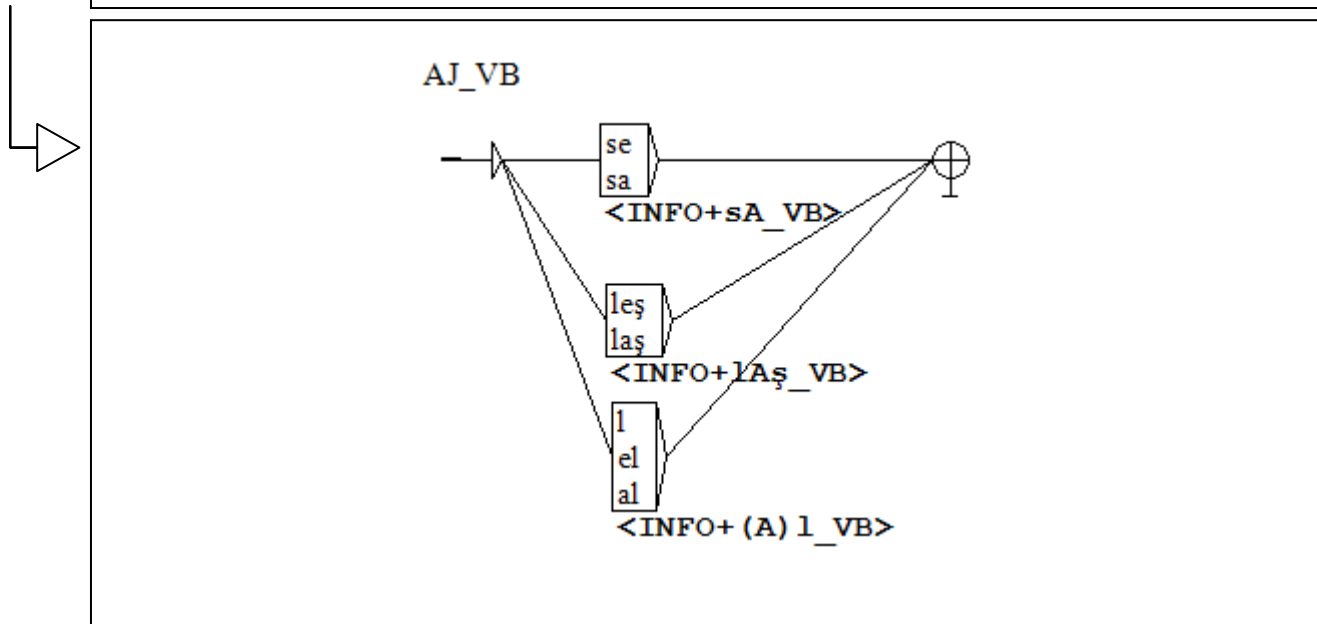
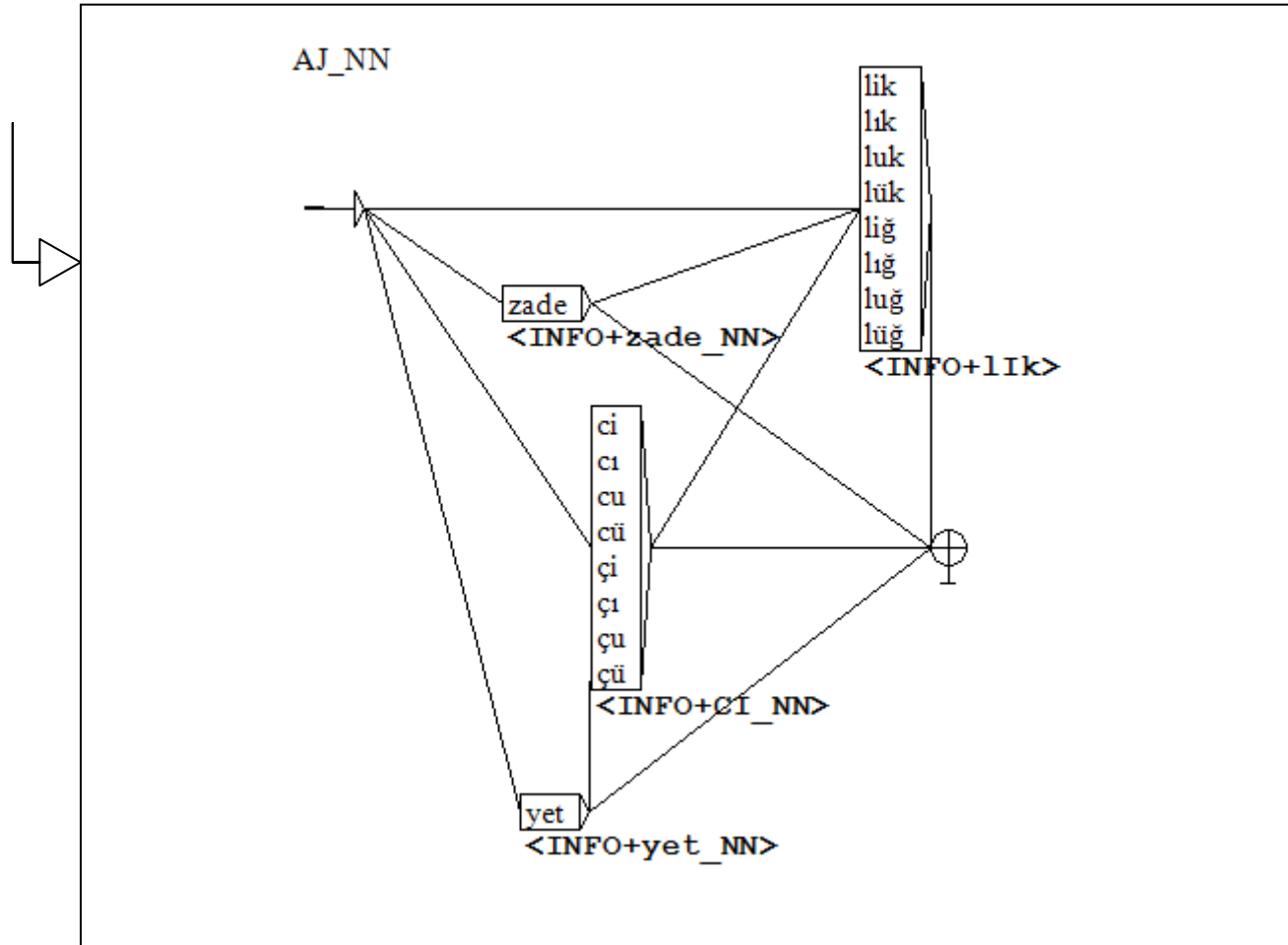


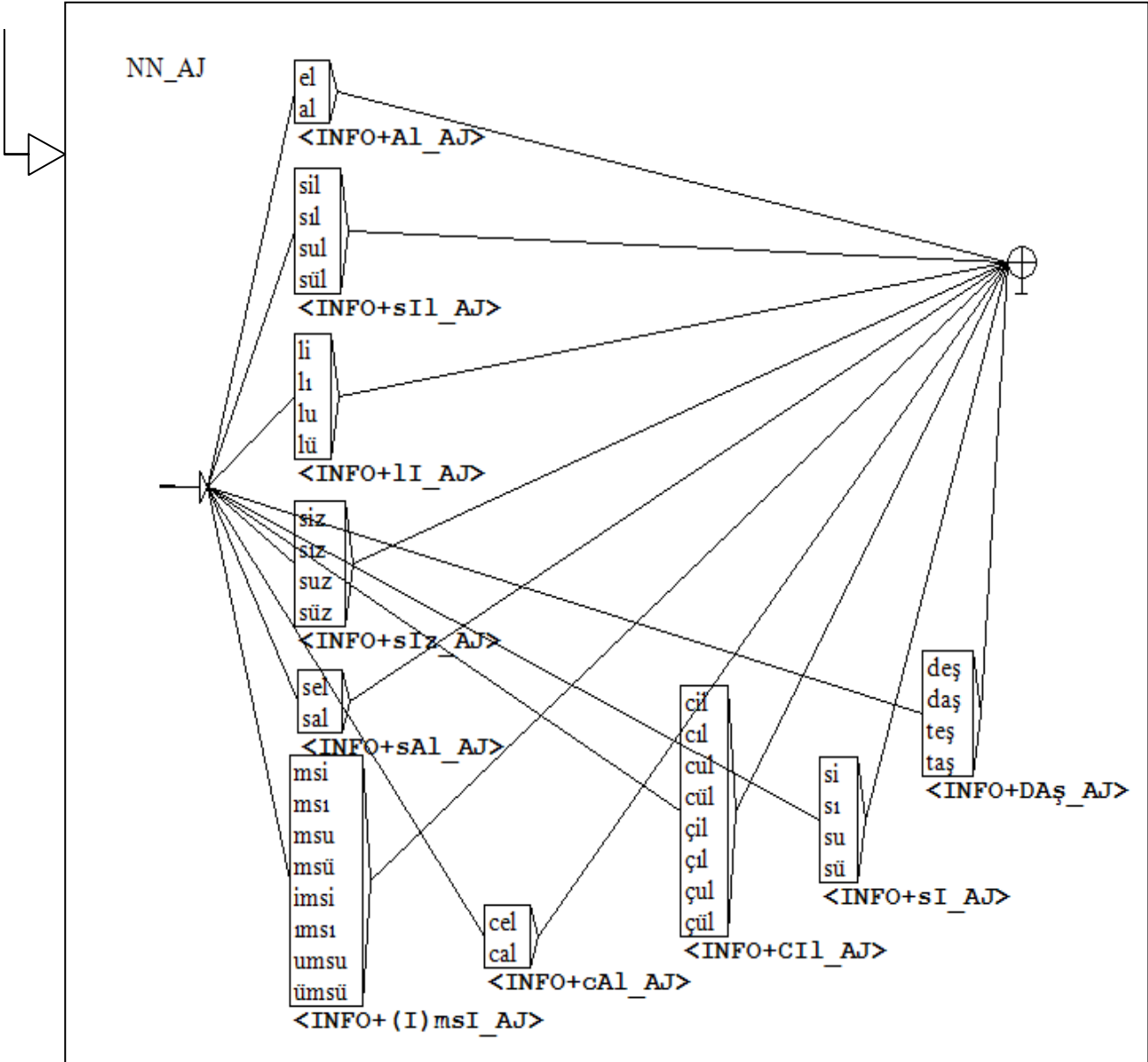


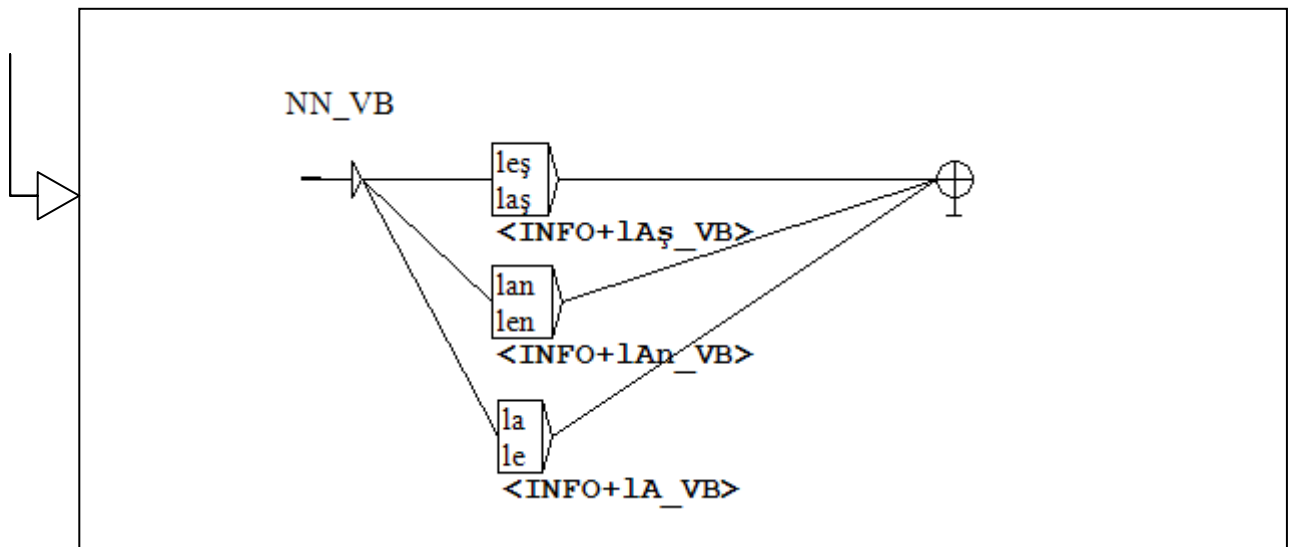
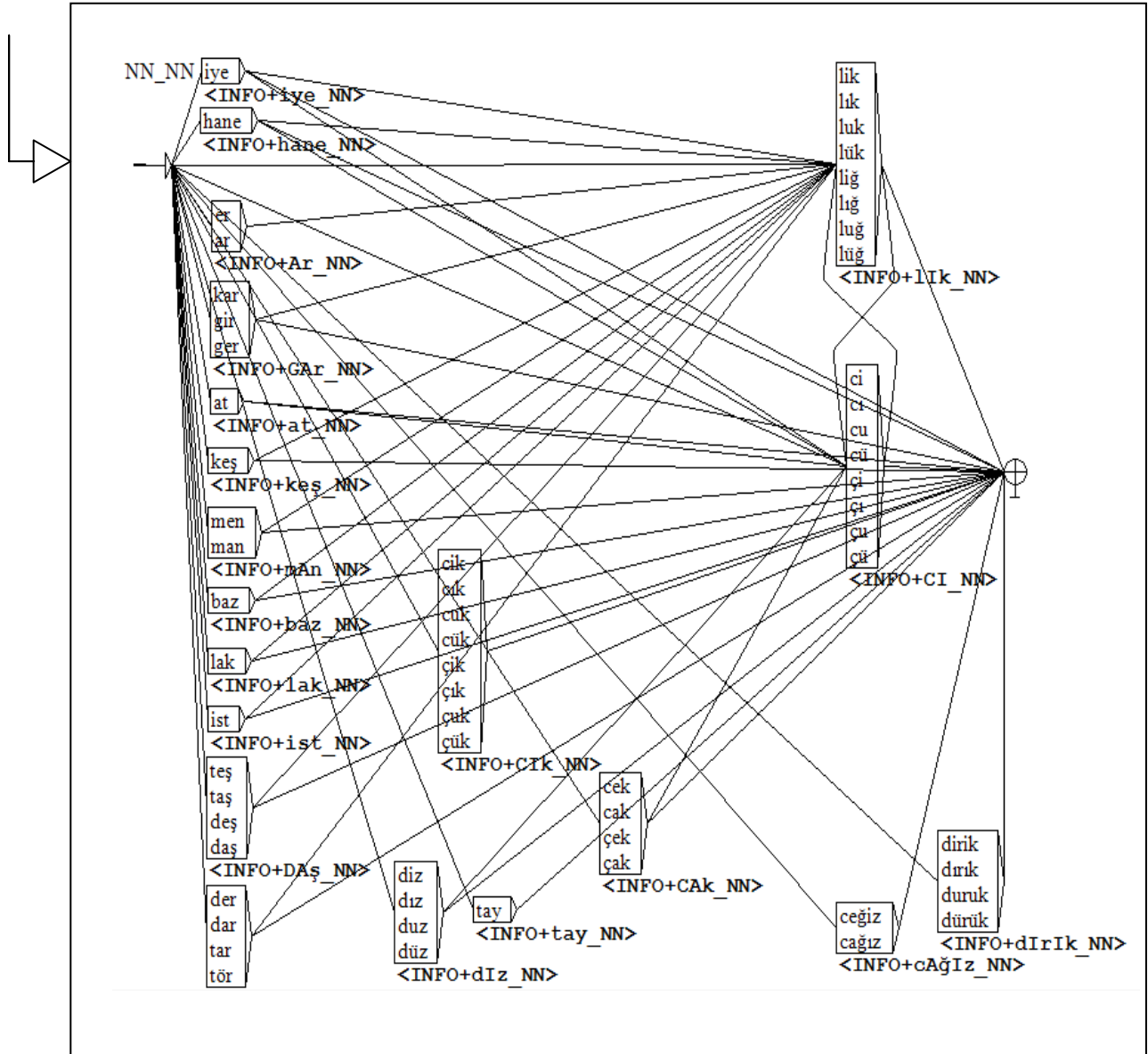
VB

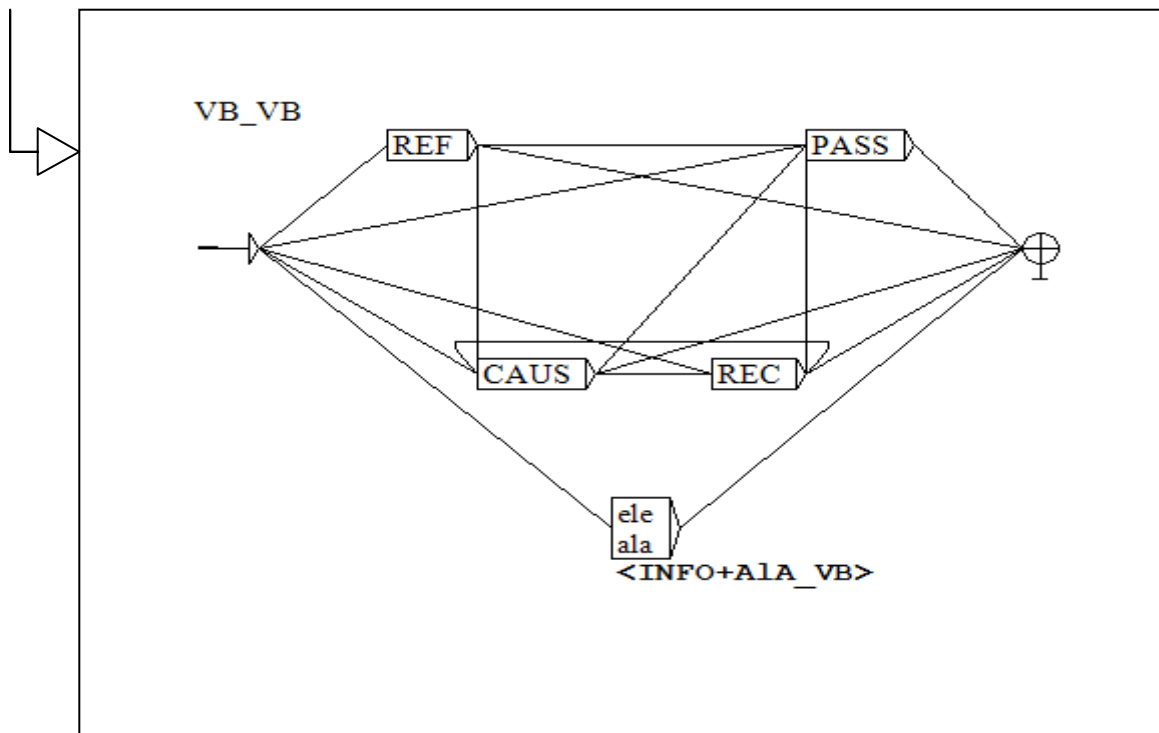
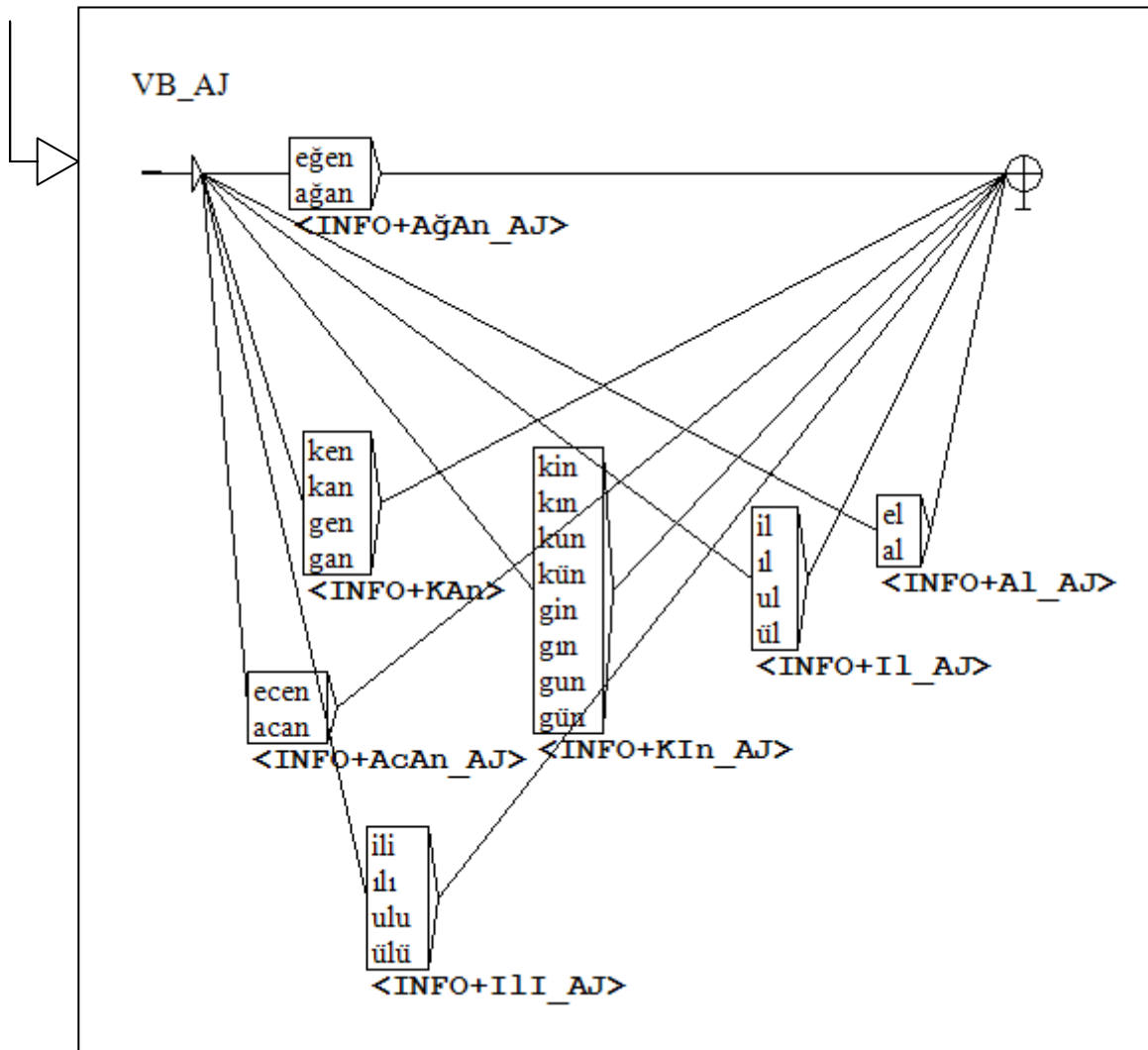


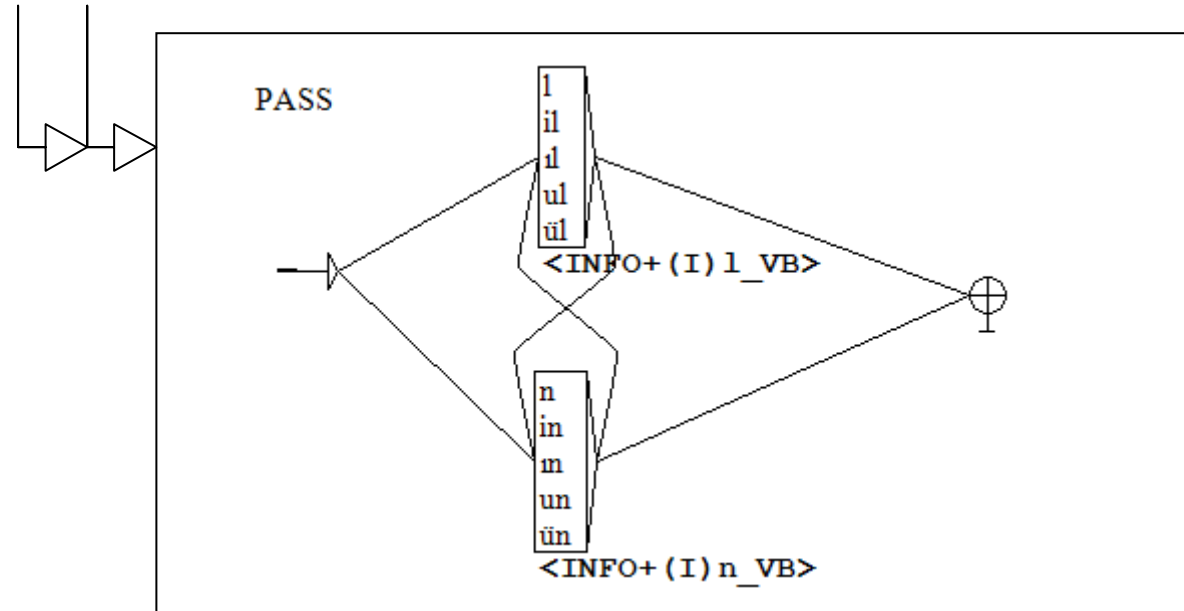
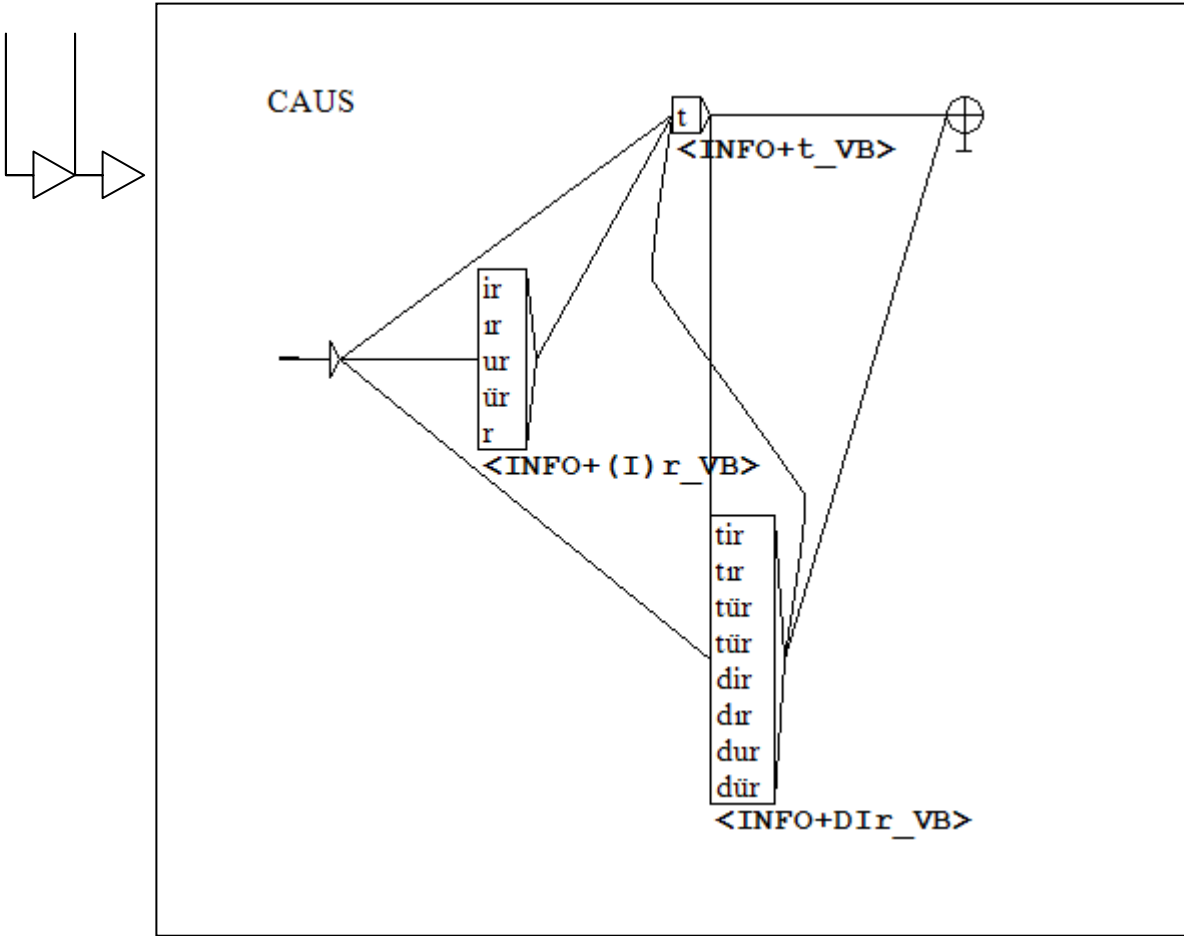


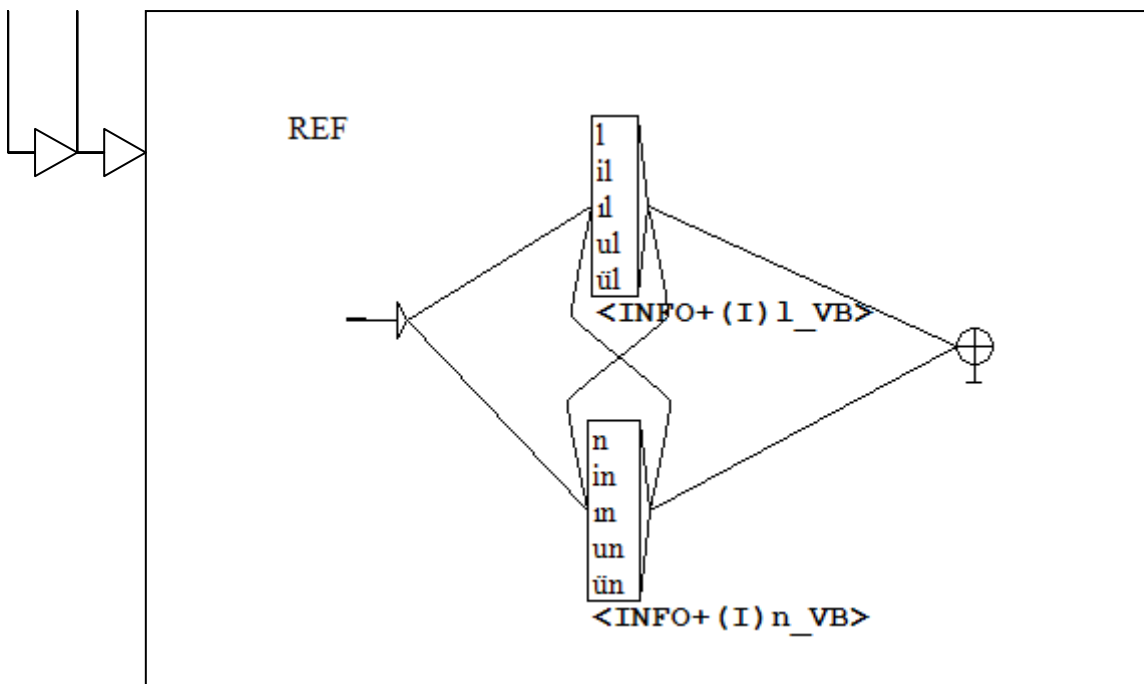
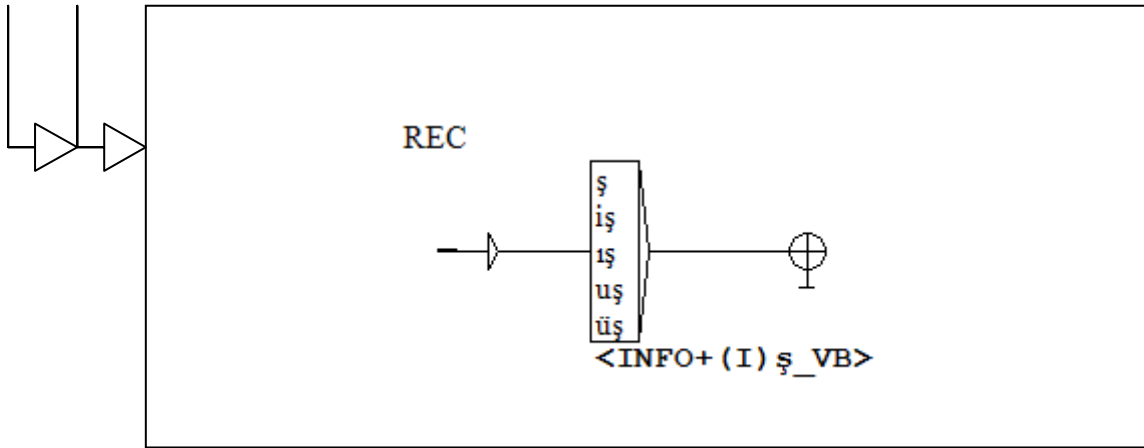


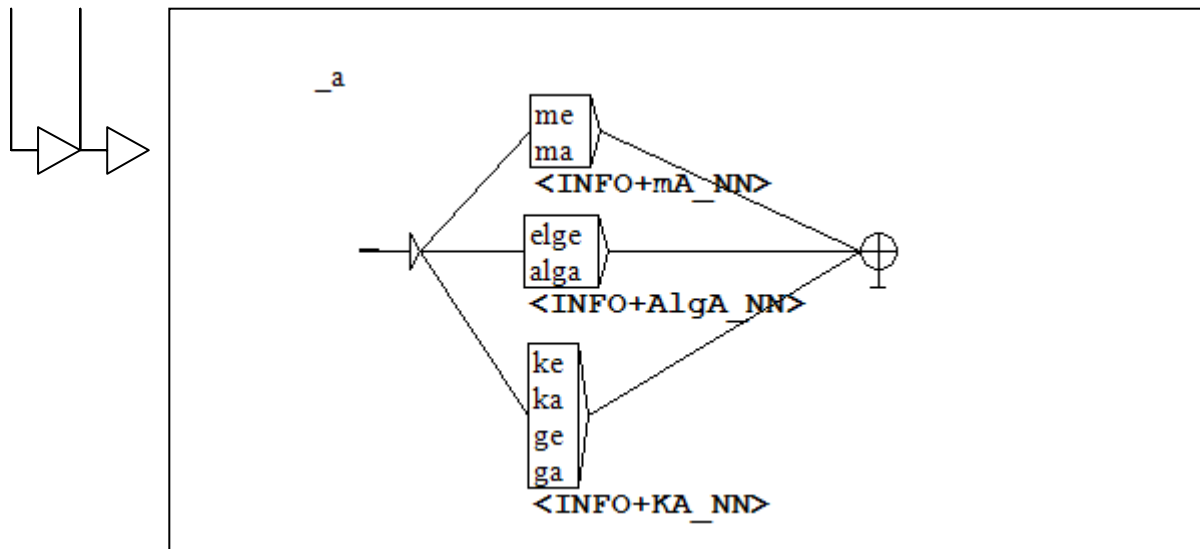
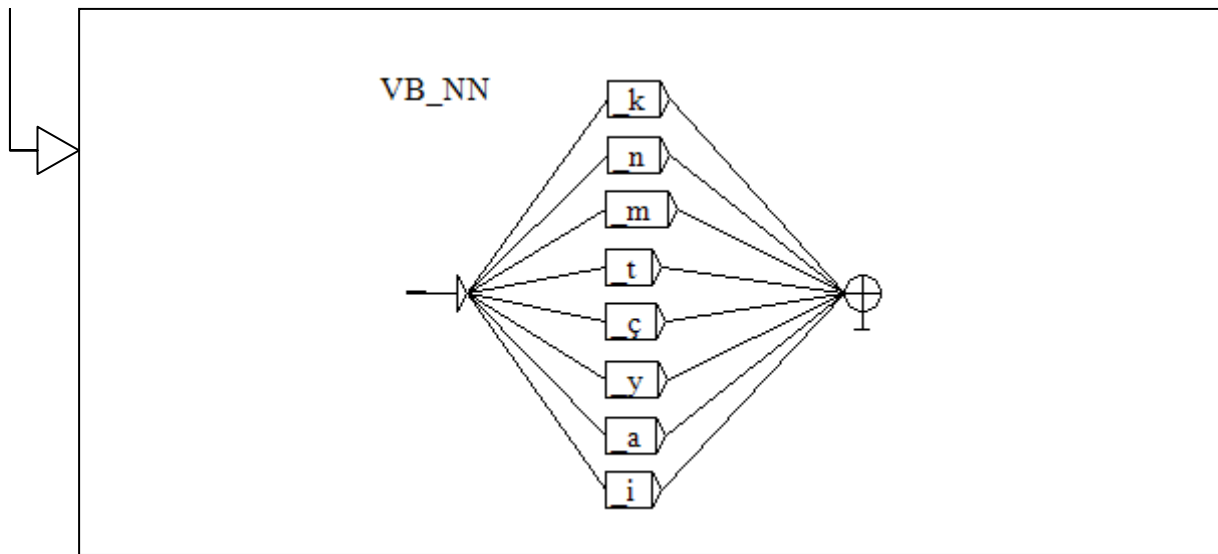


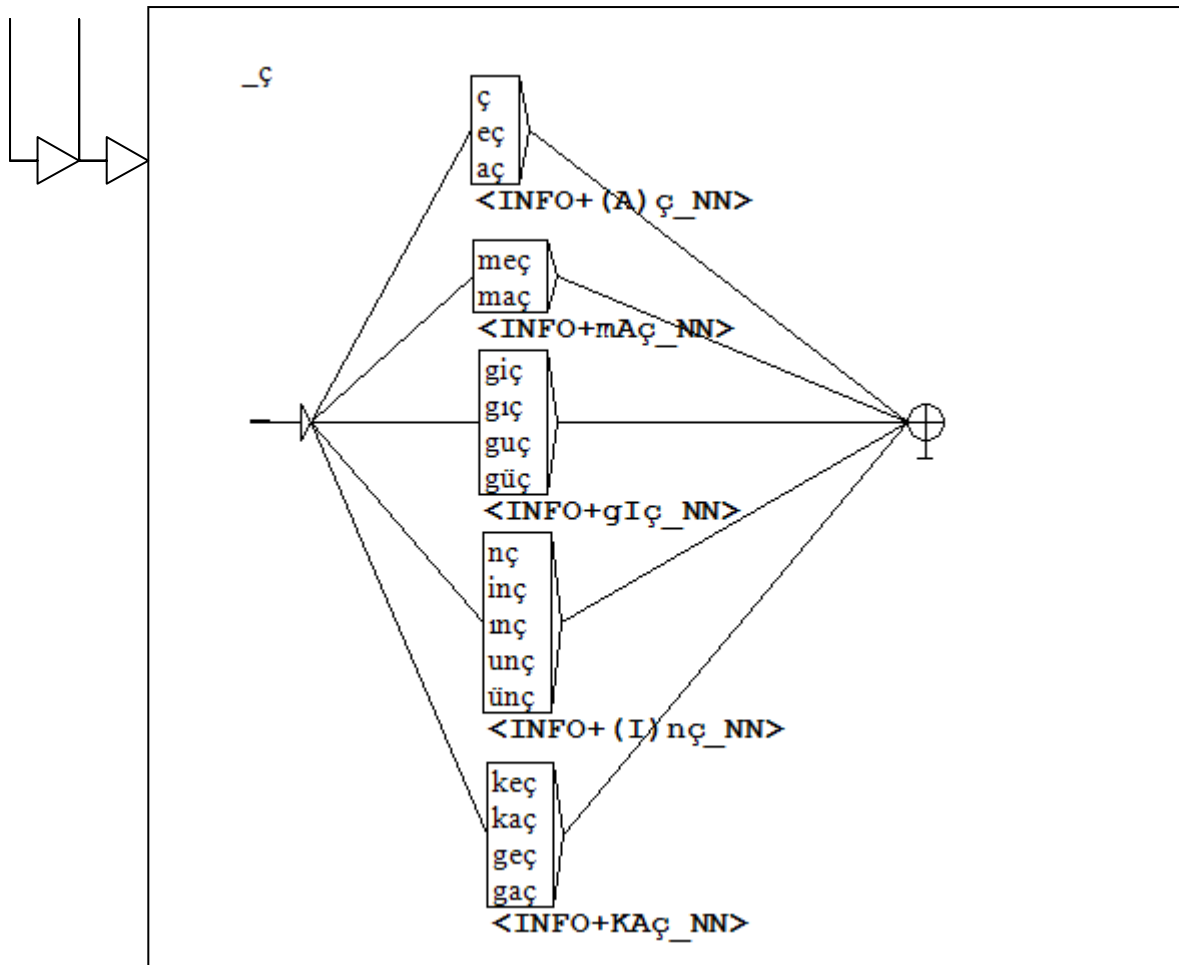


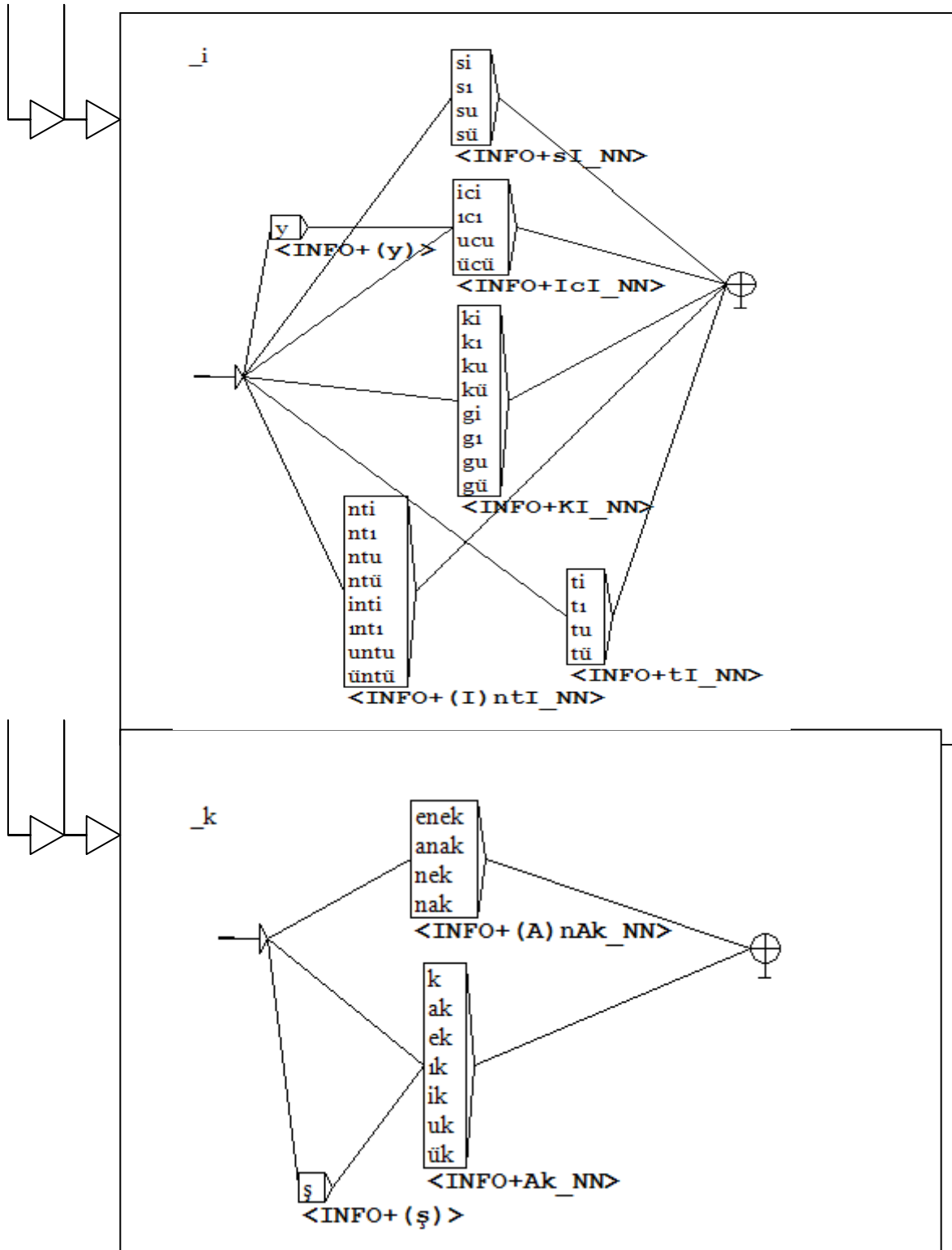


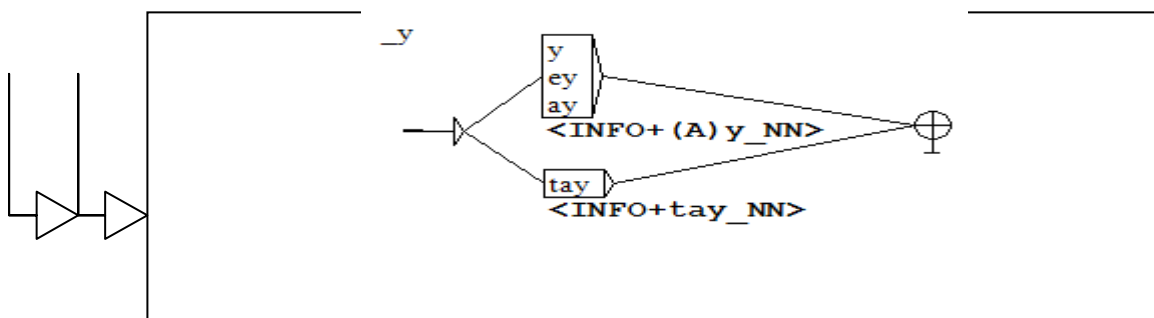
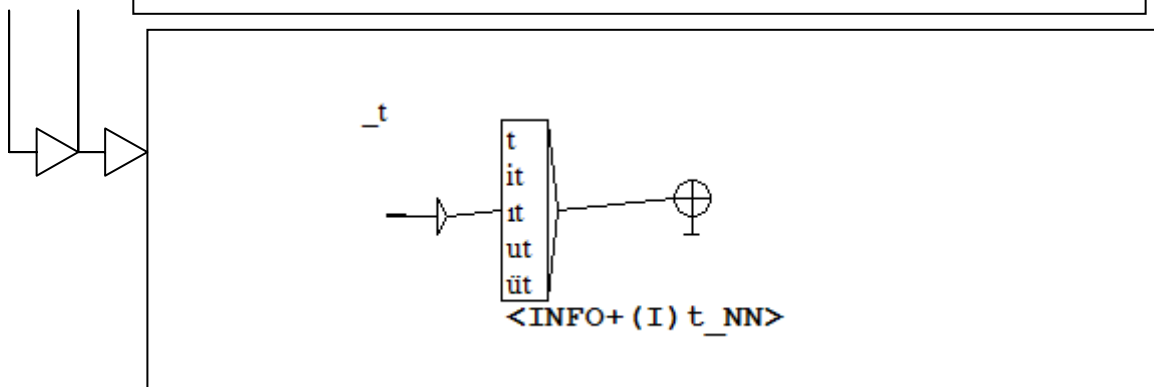
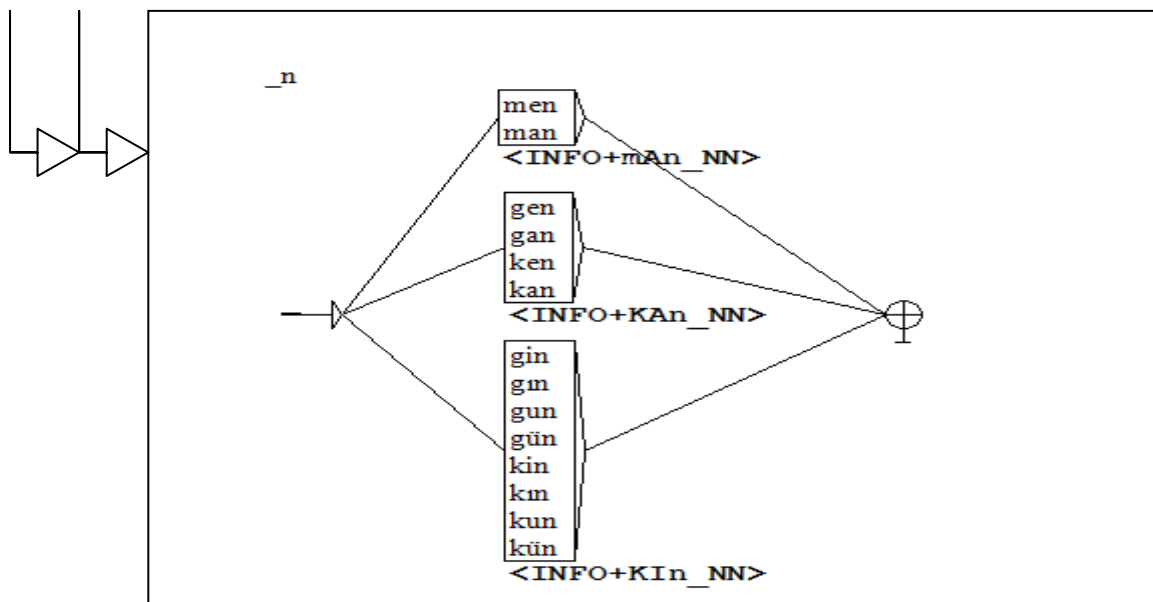
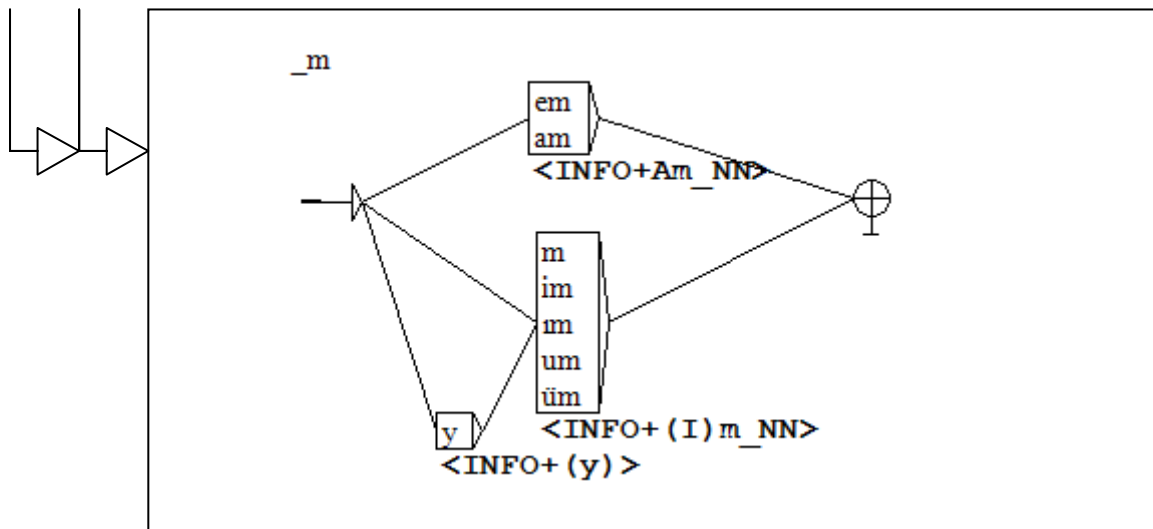




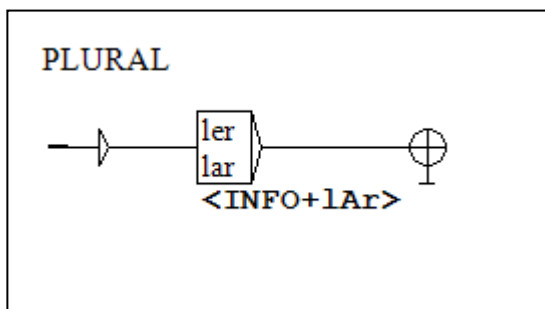
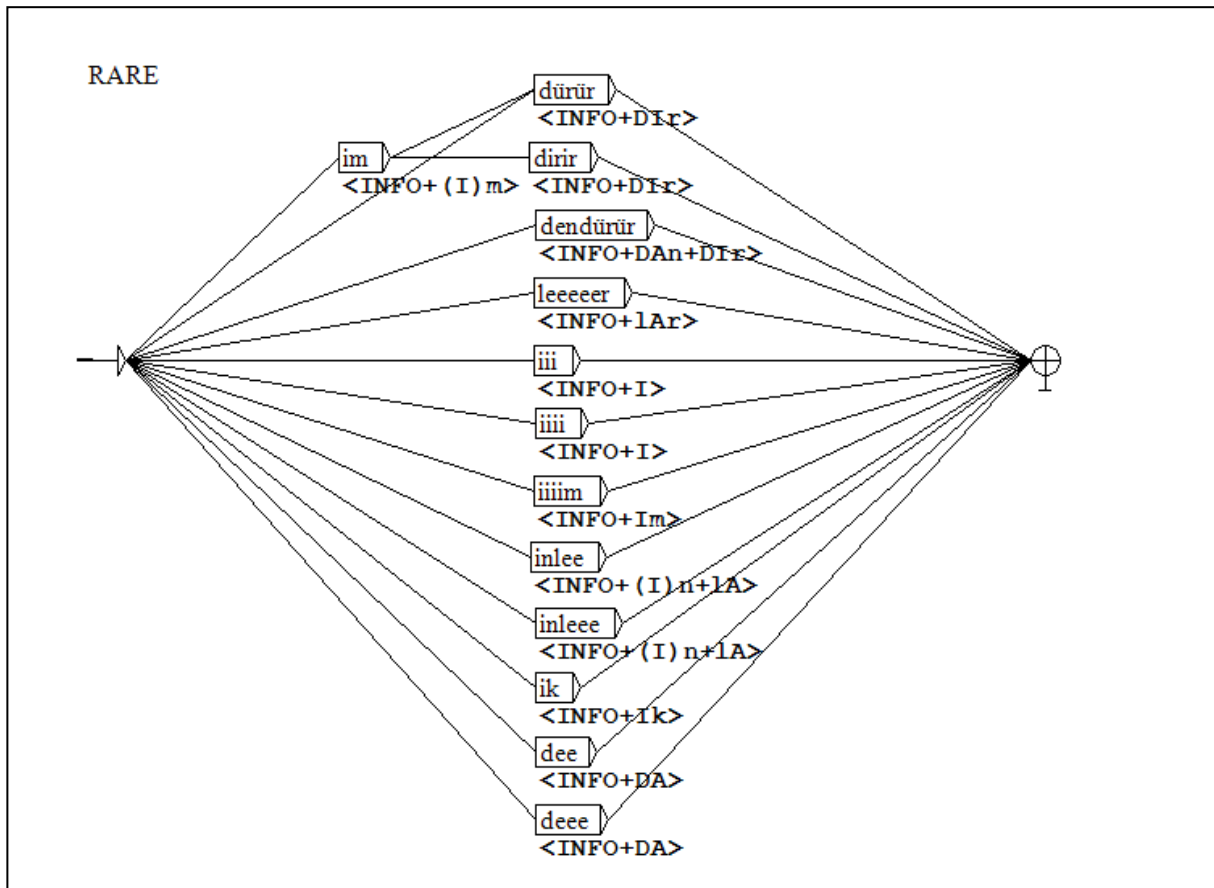
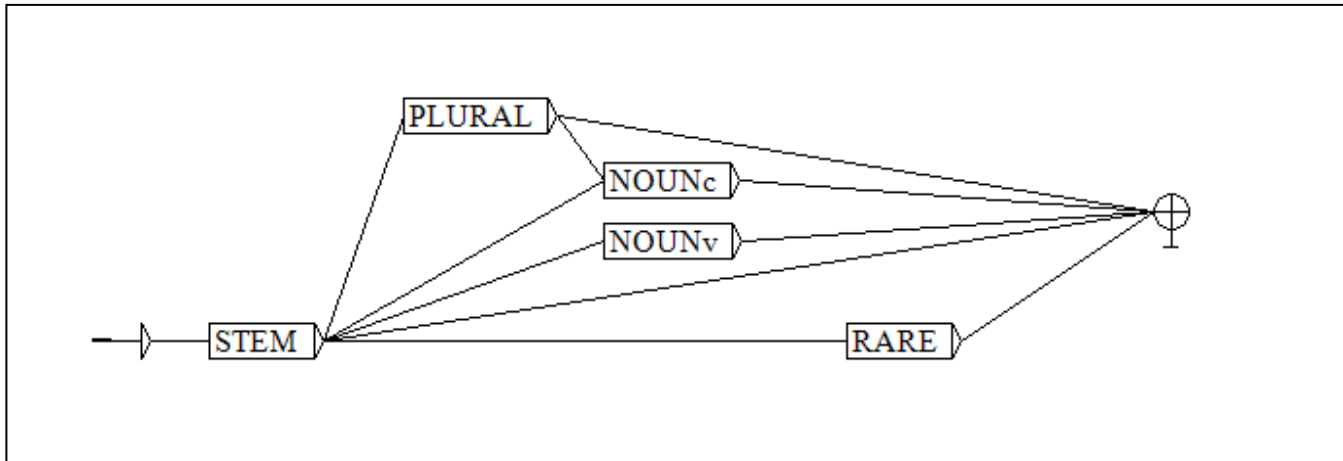


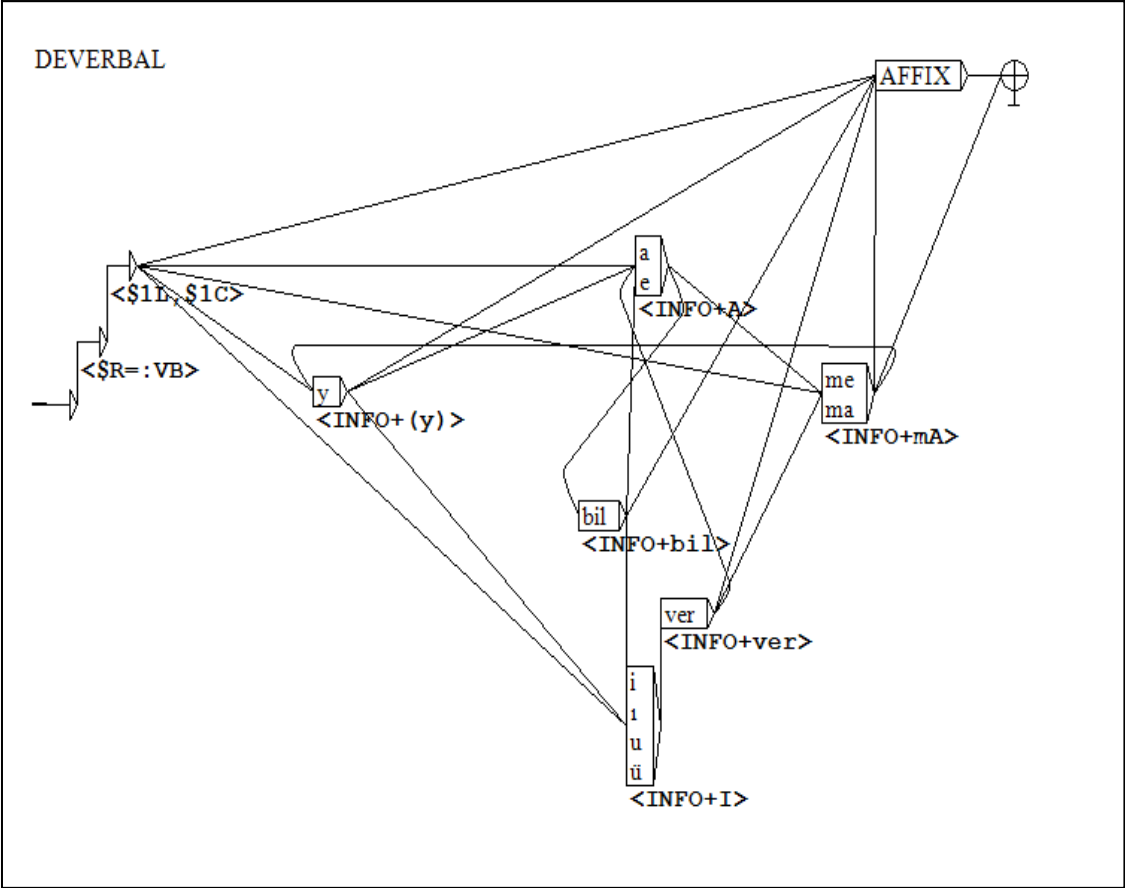
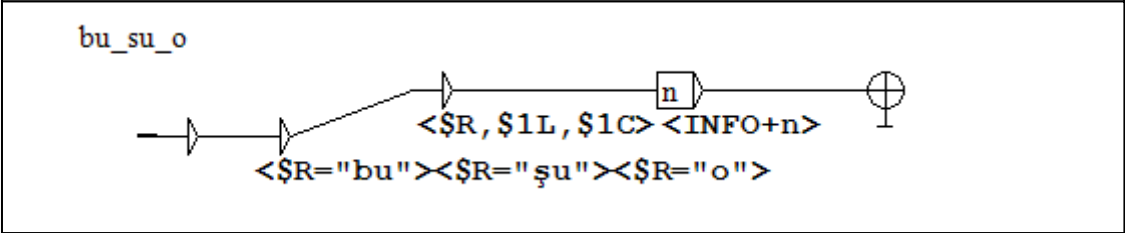
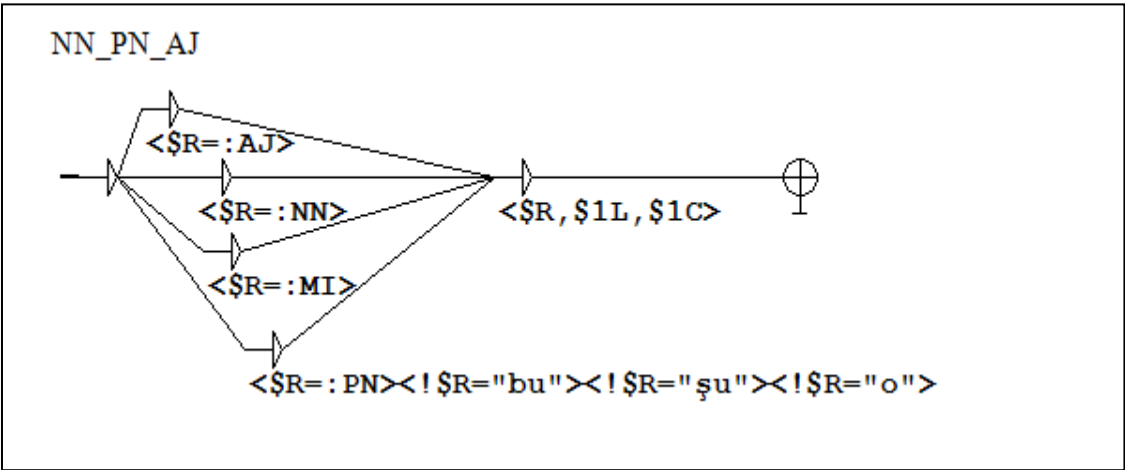
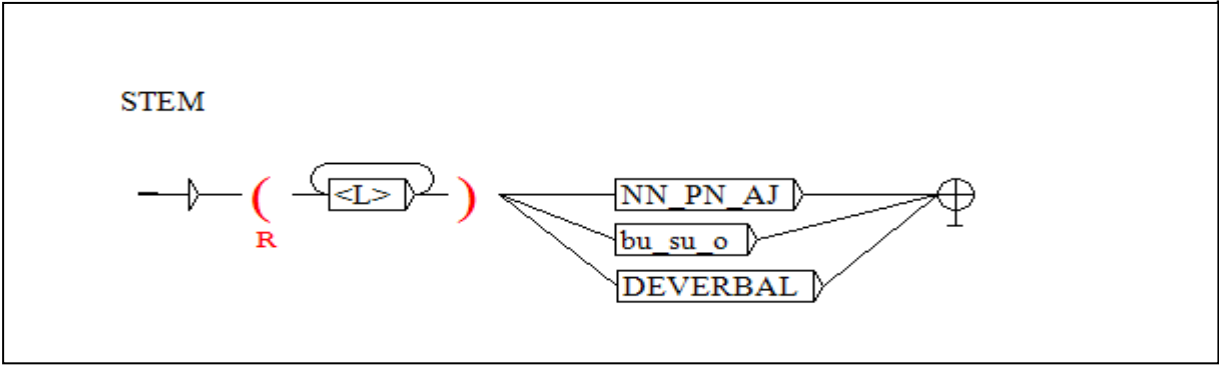


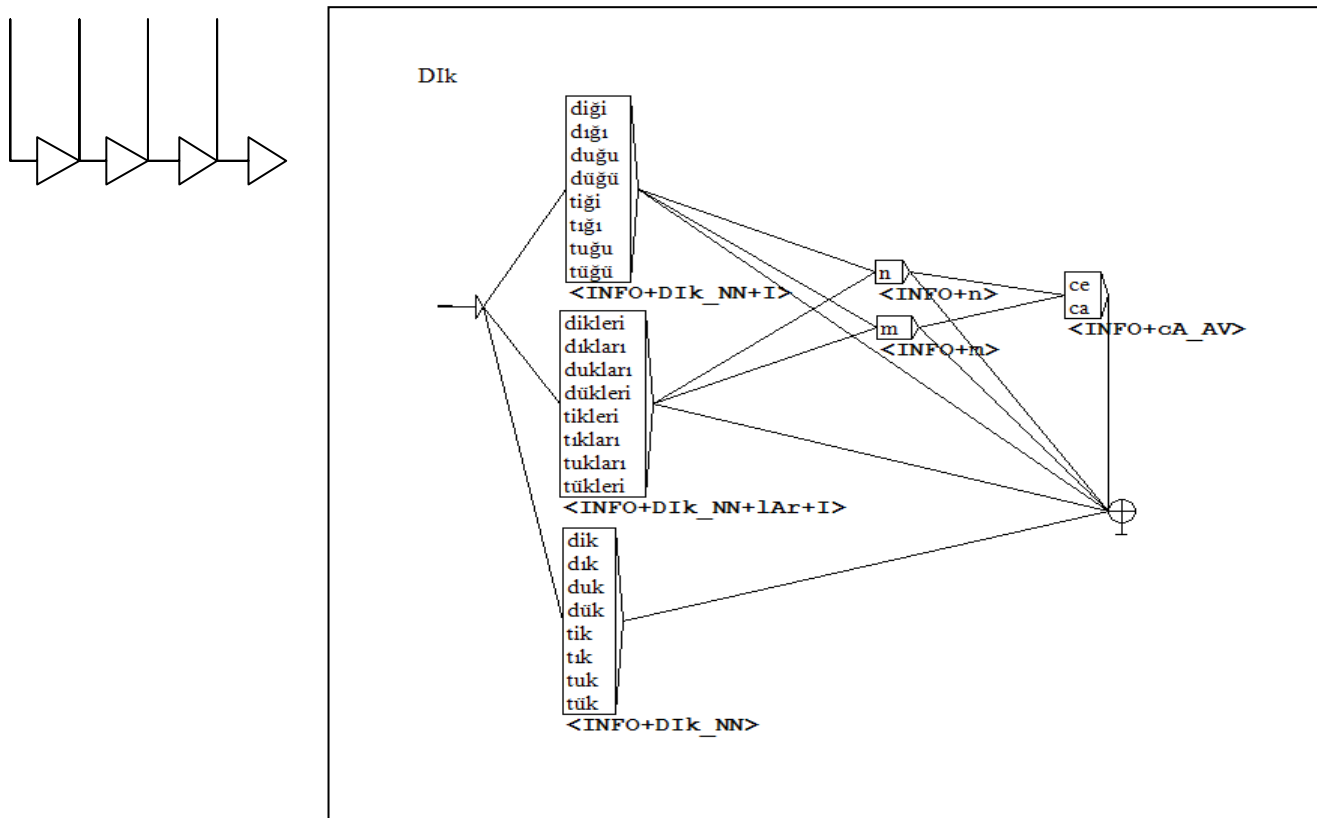
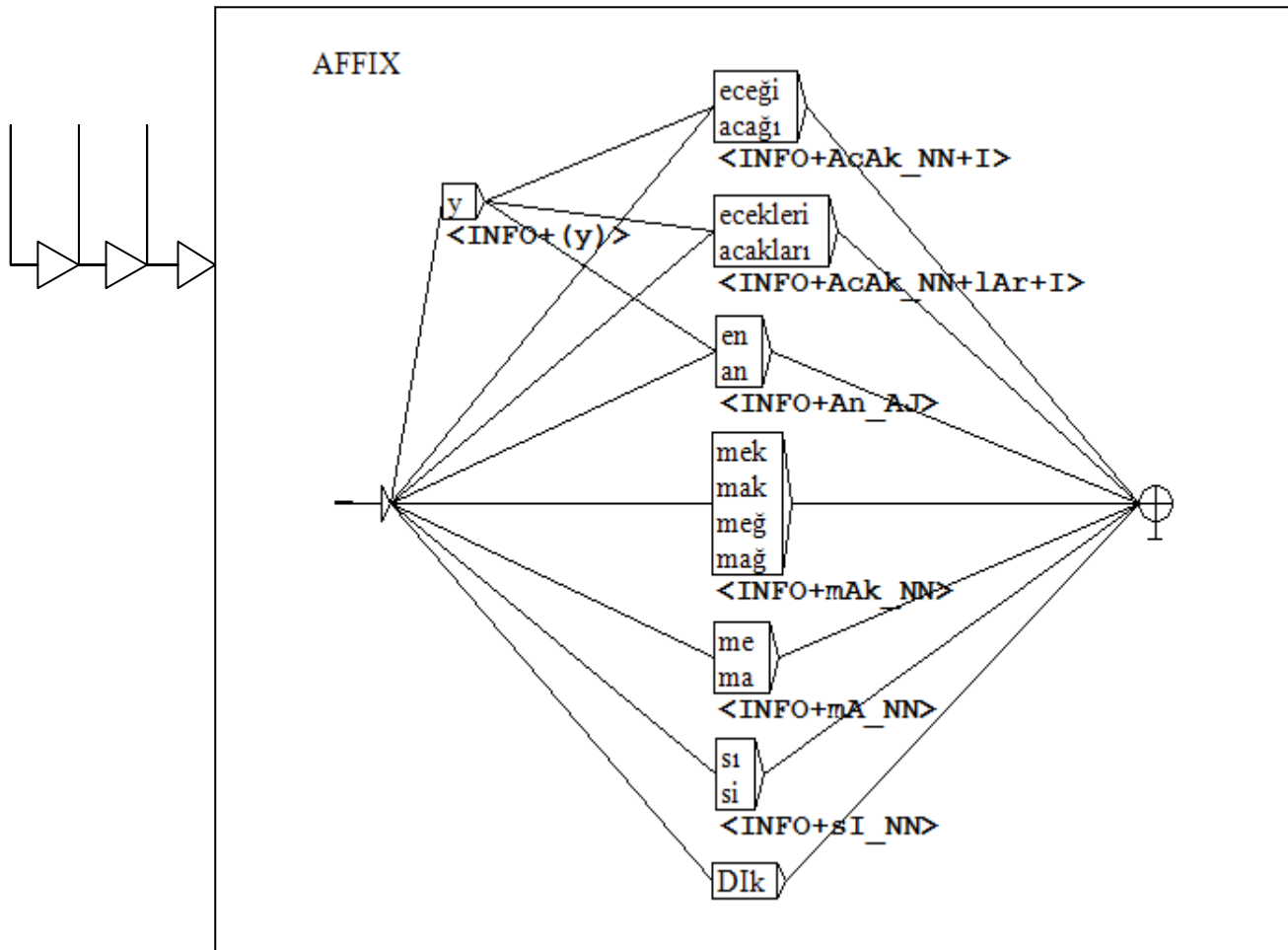


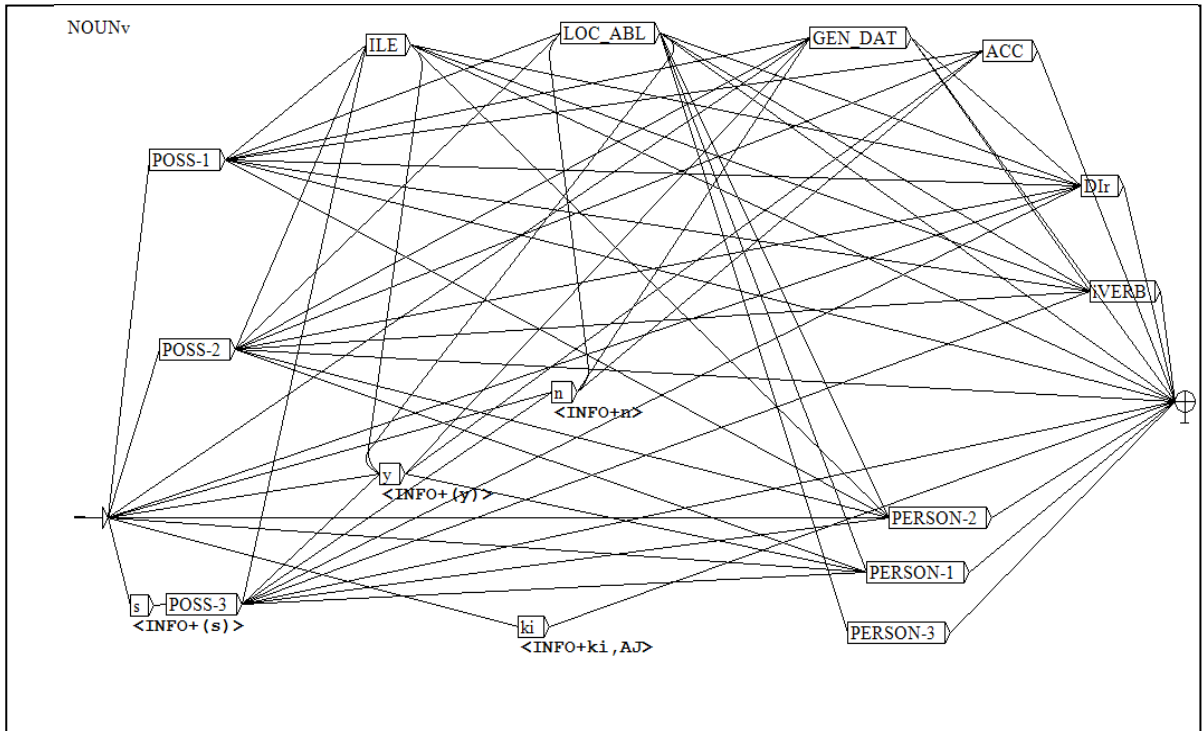
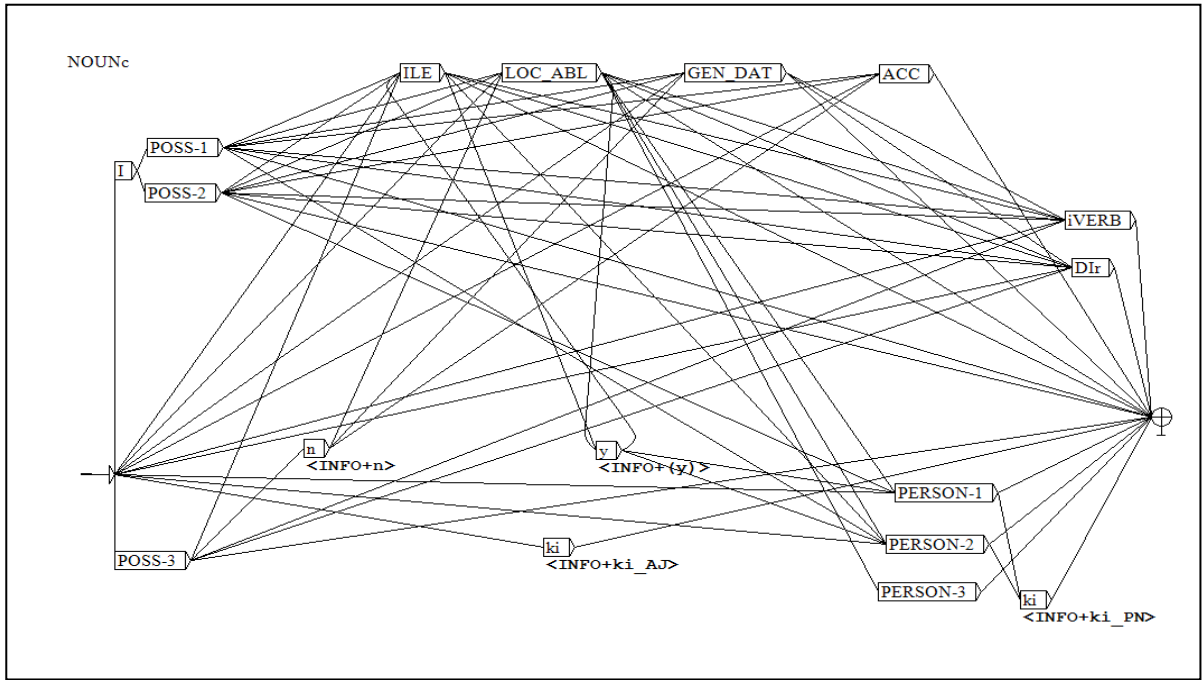


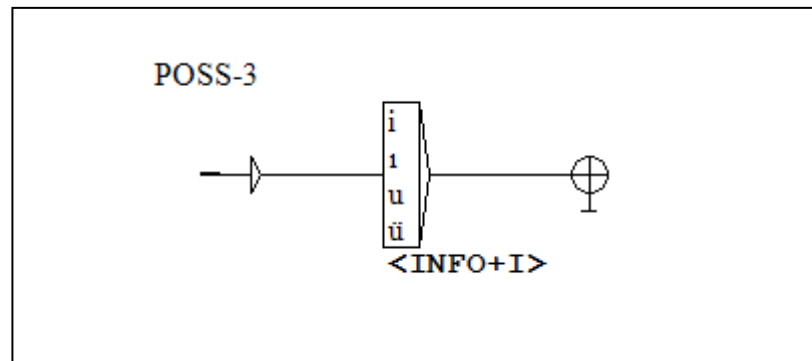
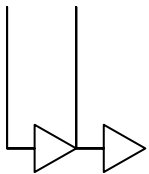
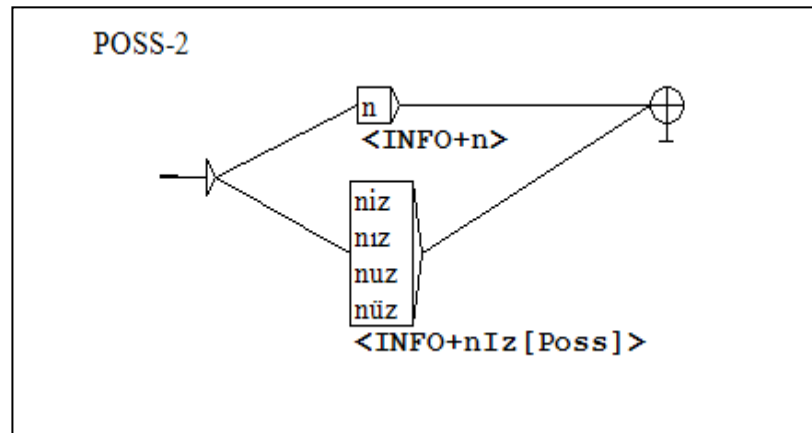
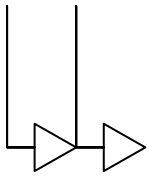
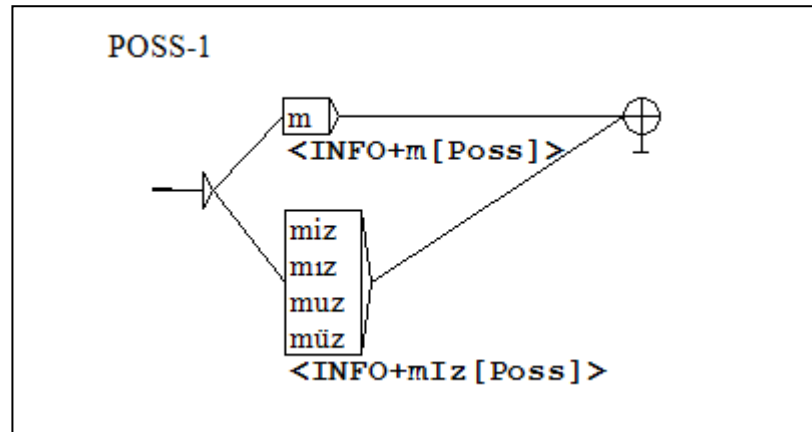
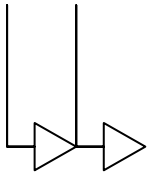
APPENDIX E
Turkish Inflectional Affixation. Nominal Paradigm.

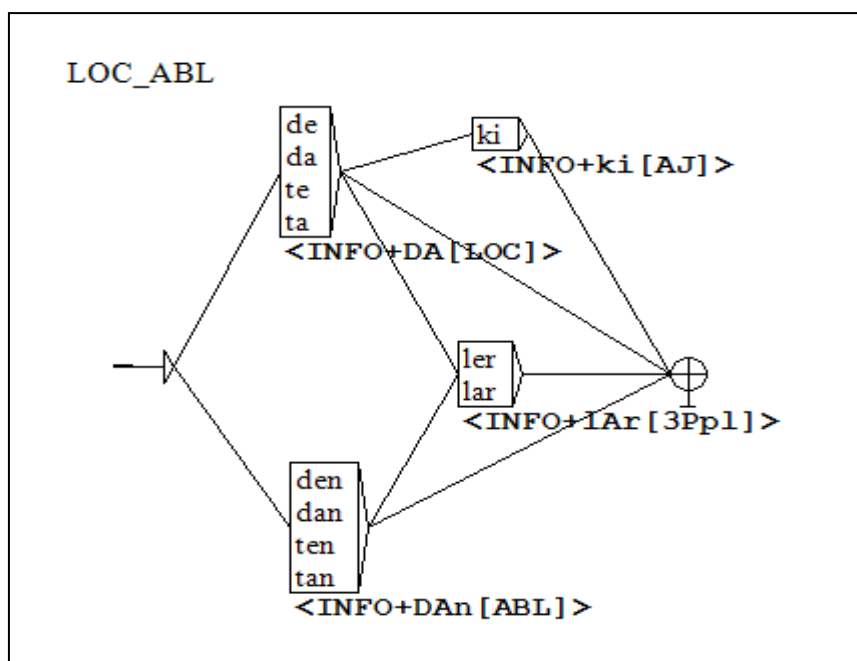
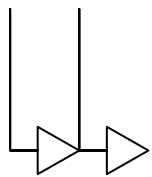
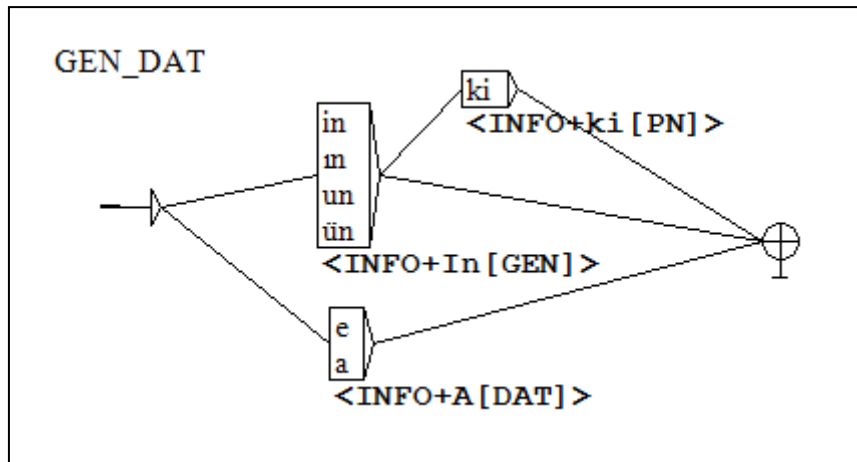
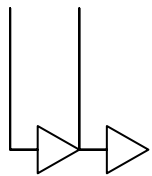
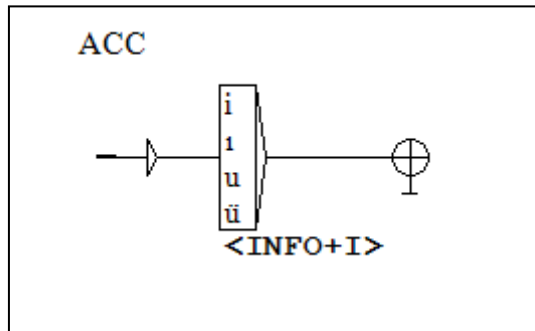
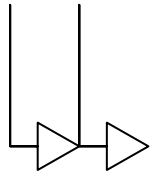
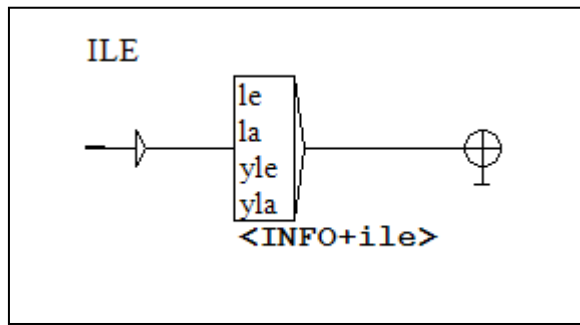
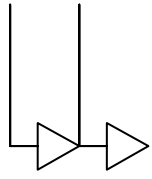


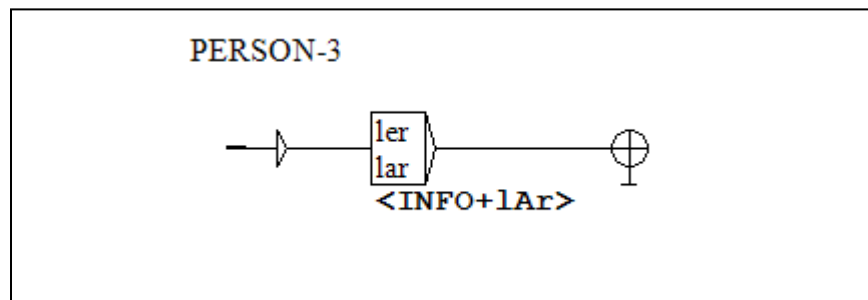
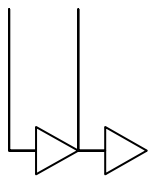
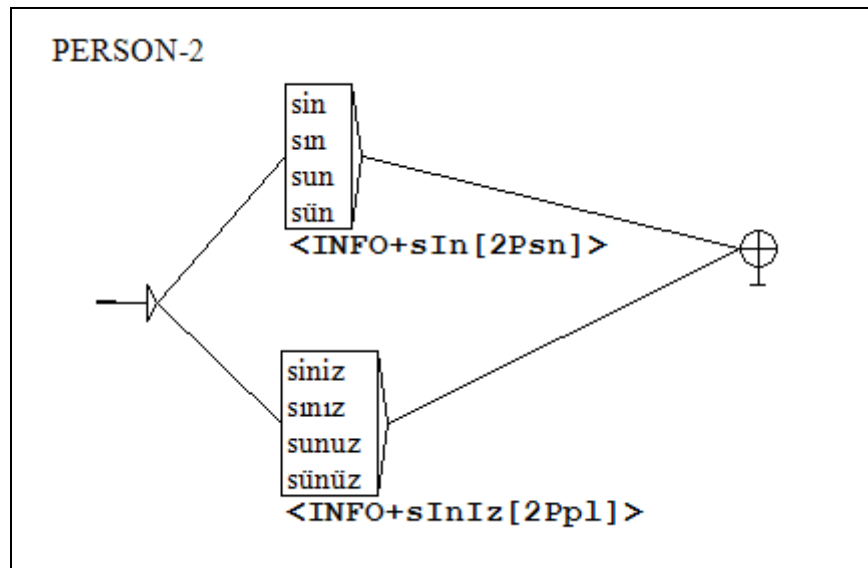
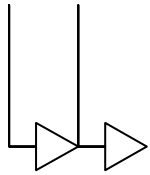
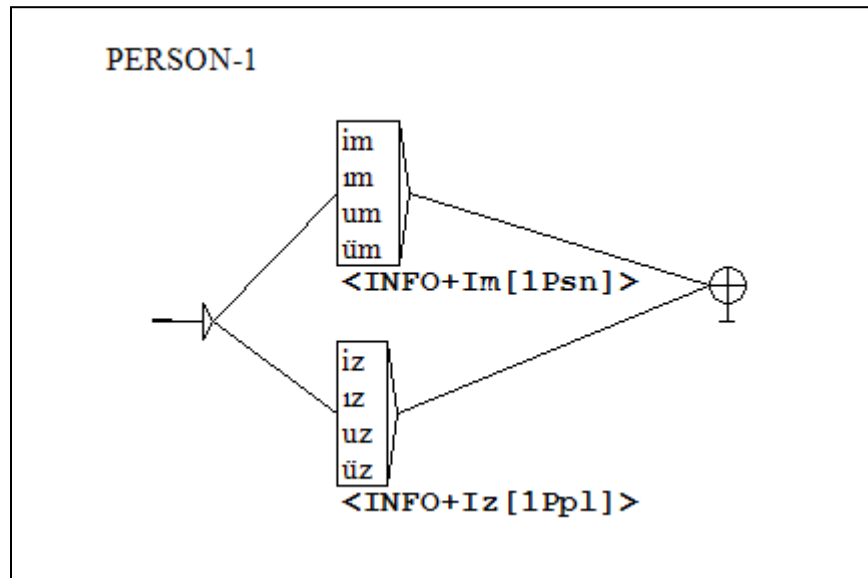
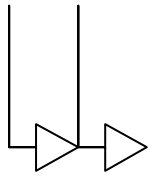


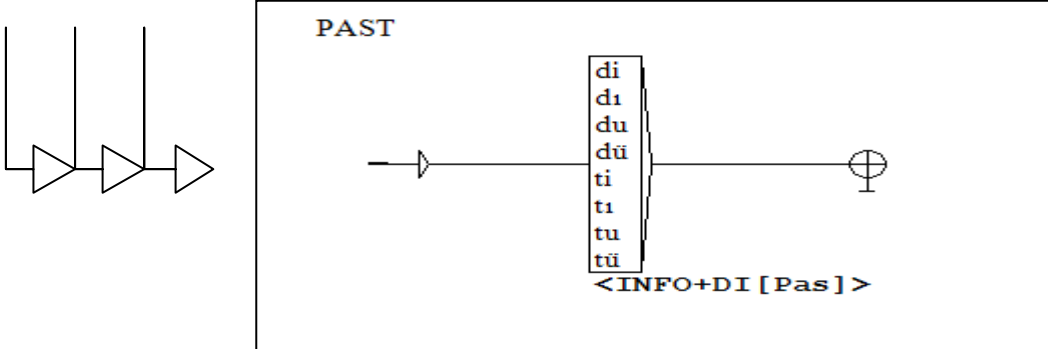
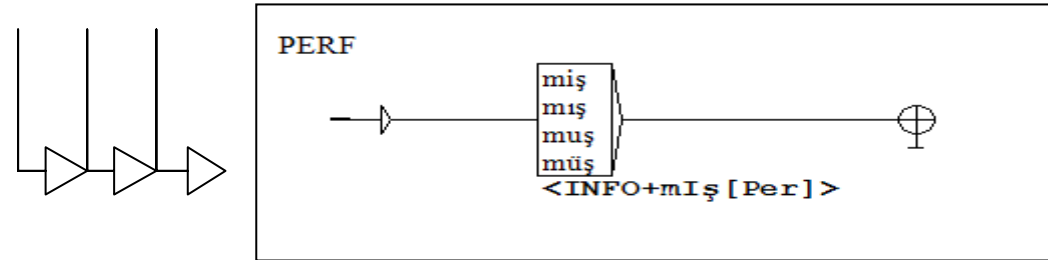
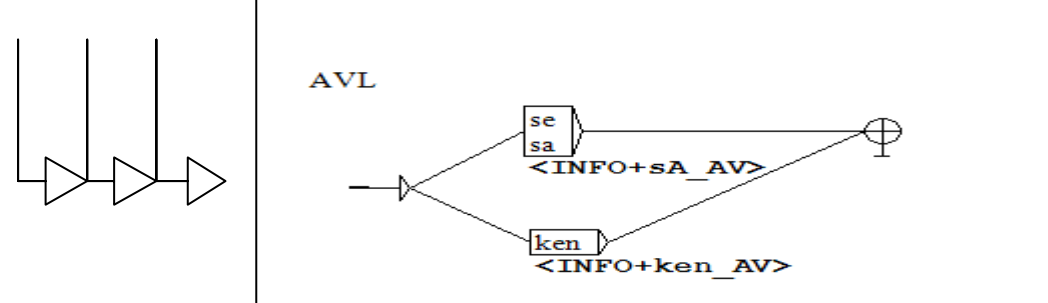
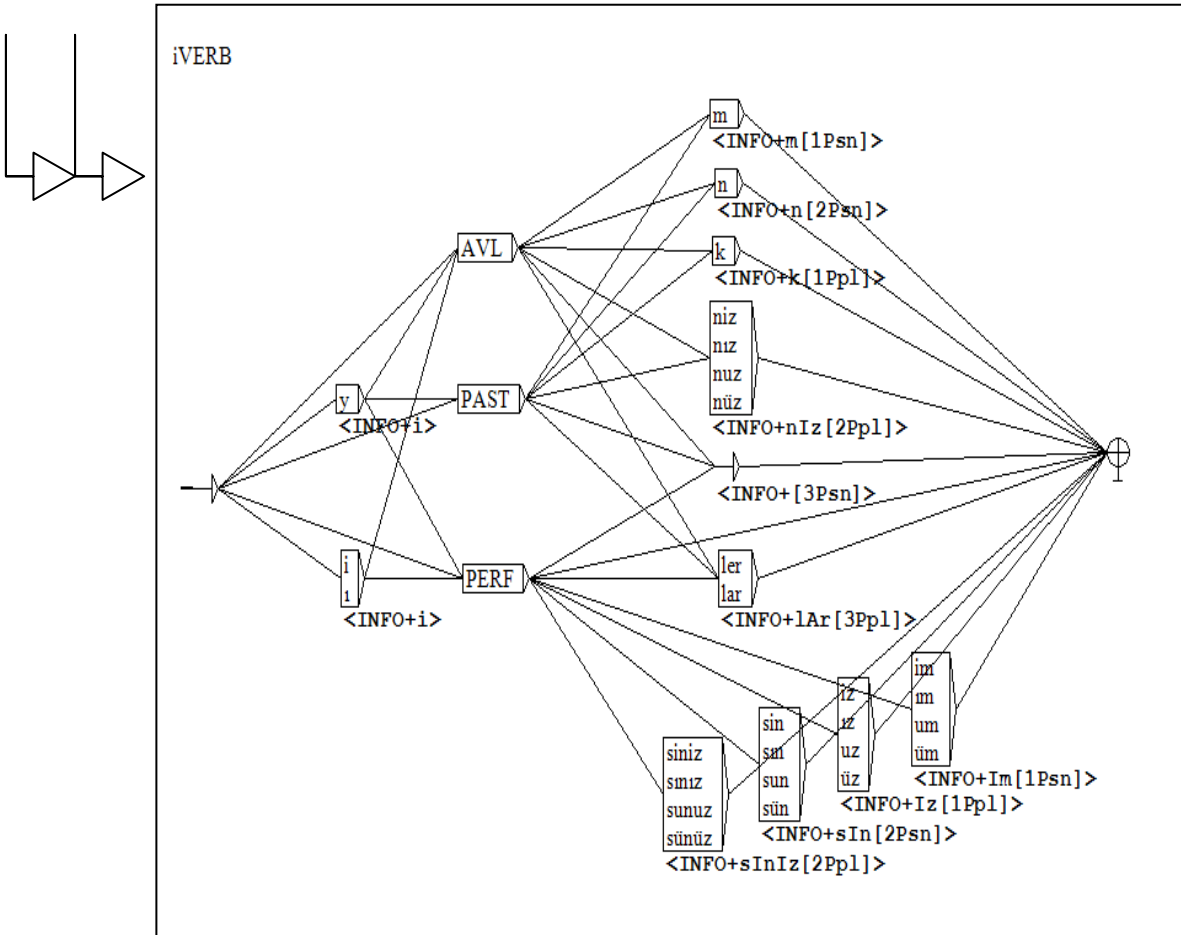












APPENDIX F
Turkish Inflectional Affixation – Verbal Paradigm

