





**ITUMORPH - TÜRKÇE İÇİN DAHA GENİŞ KAPSAMLI  
VE BAŞARILI BİR BİÇİMBİLİMSEL ÇÖZÜMLEYİCİ**

**YÜKSEK LİSANS TEZİ**

**Muhammet ŞAHİN**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Bilgisayar Mühendisliği Programı**

**ARALIK 2013**



**ITUMORPH - TÜRKÇE İÇİN DAHA GENİŞ KAPSAMLI  
VE BAŞARILI BİR BİÇİMBİLİMSEL ÇÖZÜMLEYİCİ**

**YÜKSEK LİSANS TEZİ**

**Muhammet ŞAHİN  
(504101542)**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Bilgisayar Mühendisliği Programı**

**Tez Danışmanı: Yard.Doç.Dr.Gülşen Eryiğit**

**ARALIK 2013**



İTÜ, Fen Bilimleri Enstitüsü'nün 504101542 numaralı Yüksek Lisans Öğrencisi **Muhammet ŞAHİN**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "**İTUMORPH - TÜRKÇE İÇİN DAHA GENİŞ KAPSAMLI VE BAŞARILI BİR BİÇİMBİLİMSEL ÇÖZÜMLEYİCİ**" başlıklı tezini aşağıdaki imzaları olan jüri önünde başarı ile sunmuştur.

**Tez Danışmanı :**      **Yard.Doç.Dr.Gülşen Eryiğit** .....  
İstanbul Teknik Üniversitesi

**Jüri Üyeleri :**        **Doç.Dr Şule Öğüdücü** .....  
İstanbul Teknik Üniversitesi

**Yard.Doç.Dr Arzucan Özgür** .....  
Boğaziçi Üniversitesi

.....

**Teslim Tarihi :**      **16 Aralık 2013**

**Savunma Tarihi :**    **22 Ocak 2014**





*Aileme,*



## ÖNSÖZ

Tez çalışması süresince desteğini esirgemeyen hocam Yard.Doç.Dr. Gülşen Eryiğit'e ve çalışma boyunca bizimle proje konusunda fikirlerini paylaşan Yard.Doç.Dr. Ahmet Cüneyt Tantuğ'a teşekkürü borç bilirim. Hayatımın her alanında olduğu gibi bu tez çalışmasında da sonsuz destekleri ile sürekli yanımda hissettiğim aileme; üniversite yıllarından itibaren motivasyonumu yitirdiğim anlarda bile desteğini ve yaratıcı fikirlerini benden esirgemeyen eşim Rukiye Şahin'e çok teşekkür ederim.

ARALIK 2013

Muhammet ŞAHİN



## İÇİNDEKİLER

	<u>Sayfa</u>
<b>ÖNSÖZ</b> .....	vii
<b>İÇİNDEKİLER</b> .....	ix
<b>KISALTMALAR</b> .....	xi
<b>ÇİZELGE LİSTESİ</b> .....	xiii
<b>ŞEKİL LİSTESİ</b> .....	xv
<b>ÖZET</b> .....	xvii
<b>SUMMARY</b> .....	xix
<b>1. GİRİŞ</b> .....	1
1.1 Tezin Amacı Nedir?.....	3
1.2 Literatür Araştırması .....	3
1.3 İçerik.....	5
<b>2. TÜRKÇE'NİN YAPISI</b> .....	7
2.1 Türkçenin Temel Özellikleri.....	7
2.2 Ses Bilgisi.....	7
2.2.1 Ünlülerin Özellikleri.....	7
2.2.2 Büyük Ünlü Uyumu .....	8
2.2.3 Küçük Ünlü Uyumu .....	8
2.2.4 Ünlü Düşmesi .....	8
2.2.5 Ünlü Daralması.....	8
2.2.6 Ünsüz Harfler Özellikleri ve Ses Olayları .....	9
2.2.7 Ünsüz Benzeşmesi.....	9
2.2.8 Ünsüz Düşmesi.....	9
2.2.9 Ünsüz Türemesi.....	10
2.2.10 Ünsüz Yumuşaması .....	10
2.3 Kök .....	10
2.4 Ekler .....	11
2.4.1 Yapım Ekleri.....	11
2.4.2 Addan Ad Yapan Ekler.....	11
2.4.3 Eylemden Ad Yapan Ekler .....	12
2.4.4 Eylemden Eylem yapan ekler.....	12
2.4.5 Addan Eylem Yapan Ekler .....	12
2.4.6 Çekim Ekleri.....	13
2.4.7 Ad Soylu Sözcüklere Gelen Çekim Ekleri .....	13
2.4.8 İyelik Ekleri .....	13
2.4.9 Hal Ekleri.....	13
2.4.10 İlgi Eki .....	14
2.4.11 Soru Eki.....	14

2.4.12	Çoğul Eki.....	15
2.4.13	Eylemlere Gelen Çekim Ekleri.....	15
2.4.14	Zaman Eki .....	15
2.4.15	Dilek Koşul Kipi.....	15
2.4.16	Gereklilik Eki .....	16
2.4.17	Emir Kipi .....	16
2.4.18	Çoğul Eki.....	16
2.4.19	Soru Eki .....	16
<b>3.</b>	<b>ITUMORF MİMARİSİ.....</b>	<b>17</b>
3.1	Twolc Kuralları .....	23
3.2	Sözlükteki Ekler ve Sözcük Kökleri.....	31
3.2.1	Sözlük Bölütme, İyileştirme ve Genişletme .....	33
3.3	Boyut İndirgeme .....	34
<b>4.</b>	<b>SONLU DURUMLU MAKİNELER.....</b>	<b>39</b>
4.1	Sonlu Durumlu Makineler .....	39
4.2	İsim Soylu Sözcükler İçin Tasarlanan Sonlu Durumlu Makineler .....	39
4.3	Fiil Soylu Sözcükler İçin Tasarlanan Sonlu Durumlu Makineler.....	40
4.4	Bayrak İşaretleri .....	41
4.4.1	Birleştirme Testi .....	41
4.4.2	Olumsuzluk Testi.....	43
4.4.3	Gereksinim Testi .....	43
4.4.4	İzin vermeme testi .....	43
4.5	Sözlükte Olmayan Kelimelerin Analizi.....	44
4.6	Tam Sayıların Analizi .....	47
4.7	Noktalama İşaretlerinin Analizi .....	47
<b>5.</b>	<b>DEĞERLENDİRME .....</b>	<b>51</b>
5.1	Kapsam Testi .....	51
5.2	Boyut İndirgeme Testi .....	52
<b>6.</b>	<b>SONUÇ VE ÖNERİLER .....</b>	<b>53</b>
	<b>KAYNAKLAR.....</b>	<b>55</b>
	<b>ÖZGEÇMİŞ .....</b>	<b>57</b>

## **KISALTMALAR**

<b>TDK</b>	: Türk Dil Kurumu
<b>FST</b>	: Sonlu Durumlu Dönüştürücü (Finite State Transducer)
<b>XFST</b>	: Xerox'un FST aracı
<b>HFST</b>	: Helsinki FST Aracı
<b>LEXC</b>	: Sözlük Derleyicisi (Lexicon Compiler)
<b>TWOLC</b>	: İki Seviyeli Kural Derleyicisi (Two Level Rule Compiler)
<b>BKA</b>	: Bilinmeyen Kelime Analizcisi





## ÇİZELGE LİSTESİ

### Sayfa

<b>Çizelge 1</b>	: An example Turkish predicate nominative corresponding to an English sentence.....	xx
<b>Çizelge 2</b>	: Unification flag examples. ....	xx
<b>Çizelge 3</b>	: Examples of some verbs which allow the reciprocal suffix.....	xxi
<b>Çizelge 4</b>	: Blocking and allowing the reciprocal suffix via unification flags. ....	xxi
<b>Çizelge 5</b>	: Negative setting flag example. ....	xxi
<b>Çizelge 6</b>	: Require flag examples. ....	xxii
<b>Çizelge 7</b>	: Disallow flag example. ....	xxii
<b>Çizelge 8</b>	: Analyses demonstrating the affixation order of derivational and inflectional suffixes. ....	xxii
<b>Çizelge 9</b>	: Examples of various derivations of a single word resulting in shifts in part of speech. ....	xxiii
<b>Çizelge 10</b>	: Examples of grammatically plural stems without an overt plural suffix. ....	xxiii
<b>Çizelge 11</b>	: Examples of grammatically dative stems without an overt dative suffix. ....	xxiii
<b>Çizelge 12</b>	: Possessive suffix ambiguity caused by an idiomatic usage. ....	xxiv
<b>Çizelge 13</b>	: Default agreement, possession and case attributes in Turkish noun stems. ....	xxiv
<b>Çizelge 14</b>	: Apostrophe usage in the affixation of proper nouns.....	xxiv
<b>Çizelge 1.1</b>	: Yılmaz [1]'ın Karşılaştırma Tablosu .....	5
<b>Çizelge 2.1</b>	: Sesli Harfler .....	8
<b>Çizelge 2.2</b>	: Sessiz Harfler.....	9
<b>Çizelge 2.3</b>	: Addan Ad Yapan Yapım Ekleri .....	12
<b>Çizelge 2.4</b>	: Eylemden Ad Yapan Yapım Ekleri.....	12
<b>Çizelge 2.5</b>	: Eylemden Eylem Yapan Yapım Ekleri .....	13
<b>Çizelge 2.6</b>	: Addan Eylem Yapan Yapım Ekleri.....	13
<b>Çizelge 2.7</b>	: İyelik Ekleri .....	14
<b>Çizelge 2.8</b>	: Hal Ekleri.....	14
<b>Çizelge 2.9</b>	: Zaman Ekleri .....	15
<b>Çizelge 3.1</b>	: Sözlükteki Etiketler .....	21
<b>Çizelge 3.2</b>	: Sözlükteki Etiketler .....	22
<b>Çizelge 3.3</b>	: Fazla Analiz (Overgeneration) ve Yanlış Analiz Örneği .....	22
<b>Çizelge 3.4</b>	: Sözcük Sınıfları.....	31
<b>Çizelge 3.5</b>	: Sözlük Bölütleme Örneği .....	33
<b>Çizelge 3.6</b>	: Boyut İndirgeme Yapılan Ekler .....	34
<b>Çizelge 3.7</b>	: +cHK Eki Örnekleri .....	35

<b>Çizelge 3.8</b> :+lHk Eki Örnekleri .....	35
<b>Çizelge 3.9</b> :+lH Eki Örneği .....	35
<b>Çizelge 3.10</b> :+cH Eki Örneği.....	36
<b>Çizelge 3.11</b> :+Hl, +Hn Ekleri ve Örnekleri .....	36
<b>Çizelge 3.12</b> :+Hr,+Ht, +DHr Ekleri ve Örnekleri .....	36
<b>Çizelge 3.13</b> :Fiilden İsim Yapan mA ve mAz Eki Örnekleri .....	38
<b>Çizelge 3.14</b> :lAş ve lAn Örnekleri.....	38
<b>Çizelge 4.1</b> :Fazla Üretim (Overgeneration) Örneği.....	40
<b>Çizelge 4.2</b> :İstisna Durumun Bayrak İşareti ile Düzeltilmesi .....	40
<b>Çizelge 4.3</b> :Birleştirme Testi Örnekleri .....	42
<b>Çizelge 4.4</b> :Analiz Sırasında Birleştirme Testi ve Çıktıları .....	42
<b>Çizelge 4.5</b> :Gereksinim Testi Örnekleri .....	44
<b>Çizelge 4.6</b> :Bilinmeyen Sözcüklerin Analizi.....	46
<b>Çizelge 4.7</b> :Tam Sayıların Analizi .....	47
<b>Çizelge 5.1</b> :Bilinmeyen Kelimeler Analizcisi Kullanılmadan Yapılan Kapsam Testi Sonuçları .....	51
<b>Çizelge 5.2</b> :Üretim Sınırı ve Analiz Sayısı.....	52
<b>Çizelge 5.3</b> :Üretim Sınırı ve Analiz Oranı .....	52

## ŞEKİL LİSTESİ

	<u>Sayfa</u>
<b>Şekil 3.1</b> : Twolc Kural Örnekleri [2].....	24
<b>Şekil 4.1</b> : İsimler İçin Sonlu Durumlu Makine. ....	48
<b>Şekil 4.2</b> : Fiiller İçin Sonlu Durumlu Makine.....	49
<b>Şekil 4.3</b> : Fiiller İçin Sonlu Durumlu Makine.....	50
<b>Şekil 4.4</b> : Sıfatlar İçin Sonlu Durumlu Makine.....	50



## ITUMORPH - TÜRKÇE İÇİN DAHA GENİŞ KAPSAMLI VE BAŞARILI BİR BİÇİMBİLİMSEL ÇÖZÜMLEYİCİ

### ÖZET

İnsan dilinin bilgisayarlar tarafından işlenmesi, doğal dil işlemenin temel konusu olup, yapay zekanın bir alt dalıdır. Amacı insan-insan iletişimini ve insan-bilgisayar iletişimini artırmaktır. Doğal dil işlemenin birçok uygulamasında, bir biçimbilimsel çözümleme bileşenine ihtiyaç duyulur. Biçimbilimsel çözümleme, bir sözcüğün kökünü ve eklerini doğru bir şekilde ayırıp, bunlara ait gerekli sınıf etiketlerini üretmektir. Biçimbilimsel çözümlemenin çıktıları, birçok doğal dil işleme uygulamasının ana girdisini oluşturmaktadır. Türkçe için halihazırda birçok biçimbilimsel çözümleyici mevcuttur. Ancak bunlar, bazı durumlarda çözümlemesi istenen sözcükler için geçerli sonuç üretememekte, bazı durumlarda ise gereğinden fazla sonuç üreterek analiz çıktılarını kullanan üst düzey sistemlerin çalışmasını olumsuz yönde etkilemektedirler. Bu tez çalışmasında, önceki çalışmaların eksik ve sorunlu görülen yönleri düzeltilerek Türkçe için daha geniş kapsamlı ve başarılı bir biçimbilimsel çözümleyici geliştirilmiştir. Geliştirilen biçimbilimsel çözümleyicide uluslararası kabul gören bir etiket kümesi [3] kullanılmıştır. Çözümleyicinin geliştirilmesinde sonlu durumlu dönüştürücüler teknolojisi (FST - Finite State Transducer) kullanılmıştır. Tasarlanan biçimbilimsel çözümleyici farklı FST teknolojileri (XFST - Xerox Finite State Transducer [2], HFST - Helsinki Finite State Transducer [4]) üzerinde çalışabilir hale getirilmiştir. Buna ek olarak akademik çalışmalara hizmet etmesi amacıyla [tools.nlp.itu.edu.tr](http://tools.nlp.itu.edu.tr) adresinden bir web servisi olarak hizmete açılmıştır. Bununla birlikte Türkçe olmadığı halde Türkçe cümlelerin içinde kullanılan ve gövde halleri Türkçe sözlükte bulunmayan sözcüklerin (örn:serverlar, opsiyonel, vb) analizini yapabilen ITUMORPH ile isteğe bağlı birlikte çalışan ek bir biçimbilimsel çözümleyici (BKA - bilinmeyen kelime analizcisi) de geliştirilmiştir. Yapılan testler sonucu, tez kapsamında geliştirilen biçimbilimsel çözümleyicinin üst DDI araçlarının başarımına olan etkisi gösterilmiştir: ITUMORPH'un kullanılmasının, farklı test kümelerinde biçimbilimsel belirsizlik giderme başarımlarını ortalama %2,83 oranında artırdığı gözlemlenmiştir. Benzer şekilde aynı test kümelerinde yapılan kapsam testlerinde, çözümlemesi yapılabilen sözcük oranının ITUMORPH ile ortalama %10,12 ITUMORPH+BKA ile ortalama %12 arttığı görülmüştür.



## ITUMORPH - A HIGH COVERAGE AND HIGH PERFORMANCE MORPHOLOGICAL ANALYZER FOR TURKISH

### SUMMARY

The processing of human language by computers is the main objective of Natural Language Processing (NLP), which is a sub-field of artificial intelligence. NLP aims to enhance human-human communication and facilitate human-computer communication. In many such applications of NLP, a morphological analyzer has to be used. The task of morphological analysis can be defined as trying to deduce the stem of a word and parse the affixes constituting a word, marking their potential usages. The output of morphological analysis provides the essential input for high-level NLP applications. Morphological analyzers are generally rule-based systems implemented using finite state transducers. The development of a morphological analyzer requires time and effort, and its performance depends on resources such as the lexicon and the phonetic modelling. Morphological analyzers are useful in alleviating lexicon word deficiencies and data sparseness problems often encountered in various natural language processing systems. The two level description of the morphology of Turkish has been first described by Oflazer [5]. Since then, Oflazer [5]'s analyzer has evolved and improved with many extensions in the light of comments and critiques from the community. Although there have been many other studies of morphological analysis in the literature such as Hankamer [6], Eryiğit [7], Sak et al. [8], Zemberek [9], and more recently Çöltekin [10], Oflazer [5]'s work is still considered the state-of-the-art with its high coverage on most Turkish surface word forms. Furthermore, Oflazer [5]'s framework is compatible with the universal PoS-tag scheme [3] and has a significant history of corpora annotated using its tagset unlike the other frameworks. Regardless, with the growing interest in alternative corpus annotation projects and the high amount of web data to be parsed, we see that the analyzer still needs to be extended in order to successfully handle raw data. The FST supports feature setting unification that is able to reduce and speed up transducers, constrain the co-occurrence of morpheme pairs or the occurrence of single morphemes with certain words. Flag diacritics are also useful for marking lemmata for idiosyncratic morphological behavior that is feature-based rather than phonological. Feature setting and unification functions of flag diacritics are executed at runtime and used to determine whether a path is possible within a network. A network containing flag diacritics block many illegal paths that would otherwise cause over-generation, especially for a derivationally productive language such as Turkish.

Since Turkish is an agglutinative language, even single words can be often suffixed to correspond to whole sentences in English, as seen in Çizelge 1. In the given figure, the '+' character denotes morpheme boundaries. There is a large variety of such inflectional suffixes, as well as derivational suffixes that can convert noun stems to verb or adjective stems, or verb stems to noun or adjective stems. The main objective of using flag diacritics is to add a bit of memory to the finite state machine during the

**Çizelge 1:** An example Turkish predicate nominative corresponding to an English sentence.

Lexical Form	Surface Form	English Meaning
<i>gel+mA+Hyor+lAr</i>	<i>gelmiyorlar</i>	"They are not coming."

**Çizelge 2:** Unification flag examples.

@U.CASE.ABL@
@U.CASE.DAT@
@U.GEN.SIN@
@U.GEN.PL@
@U.num.sing@
@U.num.plural@

generation and analysis steps at runtime. Without this, a state transition on the FST would depend only on the current state and the input symbol, and there would be no constraint on which transition would be made next. Flag diacritics have also been used in other languages, such as Indonesian [11], Arabic [12] and Persian [13]. Our lexicon consists of 49321 word lemmata arranged under 12 parts of speech, namely Noun, Verb, Pronoun, Adjective, Technical, Duplication, Postposition, Question, Determiner, Pronoun, Number and Connectives. Our non-trivial finite state automata for nouns, verbs and adjectives differ from the predecessor models in many respects, and are illustrated with drawings in this thesis. (şekil 4.1, şekil 4.2, şekil 4.3, şekil 4.4)

### Flag Diacritics

The concatenation of certain affixes may not make sense according to morphotactic idiosyncrasies of a language, which may be impossible or impractical to attain by blocking state transitions in a finite state machine. The most important point of using flag diacritics is to disallow such illegal paths, in addition to scaling down surface form generation and speeding up the analysis. Since there are idiosyncratic exceptions to most morphotactic rules in Turkish, flag diacritics are very convenient.

Flag diacritics are multi-character symbols which are appended to lexical analyses of words, and the co-occurrence of some flags mark the lexical analysis as being invalid. The types of flag diacritics are Unification, Positive Setting, Negative Setting, Require Test, Disallow Test, and Clear Feature.

**Unification Test:** The most commonly used and simplest flag diacritic is the Unification Test, with the template @U.feature.value@, as in the examples in çizelge 2. In the template, U stands for Unification and the arguments are replaced by arbitrary values.

The most straightforward way of denoting incompatible morphemes for U-type flag diacritics is to add flags to both morphemes with the same feature, but with different values. For instance, U-type flag diacritics are used especially for verb morphotactics, because most verb roots do not take reflexive or reciprocal suffixes, with some exceptions such as the reciprocal stems given in çizelge 3. To block the transition that would be made with the reciprocal suffix, the flag @U.Hş.yok@ is appended to the verb root as shown in çizelge 4, which conflicts with the flag @U.Hş.var@ in the reciprocal affix +Hş. Verbs without the @U.Hş.yok@ flag are allowed to take the reciprocal suffix by default.



**Çizelge 3:** Examples of some verbs which allow the reciprocal suffix.

Root Form	English Meaning	Reciprocal Form	English Meaning
<i>döv</i>	"to beat"	<i>döv+Hş</i>	"to fight"
<i>gül</i>	"to laugh"	<i>gül+Hş</i>	"to laugh together"
<i>böl</i>	"to split"	<i>böl+Hş</i>	"to share"

**Çizelge 4:** Blocking and allowing the reciprocal suffix via unification flags.

Legal Paths	Illegal Paths
<i>döv+Hş@U.Hş.var@</i>	<i>yol+la@U.Hş.yok@+Hş@U.Hş.var@</i>
<i>gül+Hş@U.Hş.var@</i>	<i>sal+la@U.Hş.yok@+Hş@U.Hş.var@</i>
<i>it+Hş@U.Hş.var@</i>	<i>hava+lan@U.Hş.yok@+Hş@U.Hş.var@</i>

**Negative Setting:** The Negative Setting is the direct complement of the Unification Test. While the Unification Test restricts subsequent features to take a certain value in order to be valid, the Negative Setting requires them to take other values. A flag diacritic like @N.feature.value@ functions by setting the value of the feature to the complement of the given parameter.

In the example given in çizelge 5, the verb stem *gel* takes the @N.Caus.present@ flag, which means that the stem *gel* can take all verb morphotactics except for the suffix +*DHr*. Therefore, in a lexicon where all flags exclusively take Boolean values, the Unification Test and the Negative Setting are the dual of each other, providing the same functionality.

**Require Test:** The Require Test is a requirement condition for a specific feature, optionally with a specific value. If the flag is provided without a value specification, it only requires the feature to be present among the preceding unification flags, ignoring their values. If a value is also provided, the flag also requires the feature to be set to the specified value, i.e. the test performed by the flag @R.feature.value@ succeeds if and only if there exists a previous feature @U.feature.value@, and differs from the Unification Test in that it is not compatible with the cases where the feature does not previously occur.

For example, the suffix +*cAsHnA* ("as though") is considered grammatical only if it follows one of the tense suffixes +*mHş*, +*Hr* or +*Ar*, and this behavior is modeled with the Require Test as shown in çizelge 6. As can be seen from the example, the Require Test is used not only for stems, but also for suffix sequences.

**Disallow Test:** Like the Require Test, the Disallow Test is a requirement condition for a specific feature, but with a value different from the optionally given parameter. Just as the Negative Setting is the complement of the Unification Test, the Disallow Test is the complement of the Require Test. As such, when the value parameter is omitted, the Disallow Test is only compatible with preceding flags with the given feature and a neutral value.

**Çizelge 5:** Negative setting flag example.

<i>gel@N.Caus.present@</i>	+ <i>DHr@U.Caus.present@</i>
"come"	+Causative

**Çizelge 6:** Require flag examples.

<i>+Hr@U.Case.var@</i>
<i>+Ar@U.Case.var@</i>
<i>+mHş@U.Case.var@</i>
<i>+cAsHnA@R.Case@</i>

**Çizelge 7:** Disallow flag example.

<i>0@U.Adjective.X@:0@U.Adjective.X@</i>
<i>+lAn@D.Adjective.X@</i>

The test is used specifically for adjective morphotactics, since the adjective FST is frequently connected to the noun FST with zero input, and yet, some noun affixes are not convenient for nouns derived from adjective roots and must be disallowed. To handle this, the zero input symbol gets the @U.Adjective.X@ unification flag, whereas the affixes to be disallowed within the noun FST get the disallow flag @D.Adjective.X@, as shown in çizelge 7.

**Adjective Morphotactics**

Our finite state machine for adjective word stems are based on nominal morphotactics. However, some suffixes are exclusively added to adjectives.

The finite state automaton that models the affixation of adjective stems is shown in şekil 4.3. As can be seen in the figure, certain derivational suffixes or copula markers cause a part-of-speech shift in the adjective stem and convert it to a verb or noun stem. Although some adjective stems can be connected to nominal roots by zero input and used as noun stems, they are disallowed from taking certain nominal suffixes as a rule. This behavior is also modeled using the Disallow flag.

**Verb Morphotactics**

Verb stems follow an affixation pattern in which they optionally take reflexive, reciprocal, causative and/or passive derivational suffixes, followed by the optional polarity suffix and the mandatory tense and person suffixes as exemplified in çizelge 8. The tense is denoted by at least one aorist, progressive, perfect, future or narrative tense suffix or an imperative, necessitative, optative or conditional mood suffix, optionally followed by a second suffix for compound tenses. Due to this ordinal hierarchy, verb morphotactics are significantly more complicated than those for the other parts of speech. Flag diacritics are also often used for verb stems in addition to adjusting state transitions to implement some of these ordinal constraints.

**Çizelge 8:** Analyses demonstrating the affixation order of derivational and inflectional suffixes.

<i>Döv</i>	<i>+Hn</i>	<i>+mA</i>	<i>+yAcAk</i>	<i>+yDH</i>	<i>+Hm</i>
"beat"	+Reflexive	+Negative	+Future	+Past	+P1sg
<i>Patla</i>	<i>+t</i>	<i>+Hl</i>	<i>+yAcAk</i>	<i>+lAr</i>	
"explode"	+Causative	+Passive	+Future	+P3pl	

**Çizelge 9:** Examples of various derivations of a single word resulting in shifts in part of speech.

Analysis	English Meaning
<i>Oda+DA</i>	"in the room" ( <i>Noun</i> )
<i>Oda+DA+yken</i>	"while [ <i>he is</i> ] in the room" ( <i>Adverb</i> )
<i>Oda+DA+ymHş+CAshNa</i>	"as though [ <i>he were</i> ] in the room" ( <i>Adverb</i> )
<i>Oda+DA+ysA</i>	"if [ <i>he is</i> ] in the room" ( <i>Adverb</i> )
<i>Oda+DA+ymHş</i>	"[ <i>he</i> ] was in the room" ( <i>Verb</i> )
<i>Oda+DA+yHm</i>	"[ <i>I</i> ] am in the room" ( <i>Verb</i> )

**Çizelge 10:** Examples of grammatically plural stems without an overt plural suffix.

<i>ahali</i> , "crowd"	Noun+A3pl+Pnon+Nom
<i>enkaz</i> , "debris"	Noun+A3pl+Pnon+Nom

Furthermore, there are many irregularities among verb stems that dictate which derivational suffixes may be appended to which stems, and in which combinations. These irregularities may be handled by grouping similarly behaving verb stems together and partitioning the lexicon so that verb stems of certain groups are only allowed to take certain combinations of suffixes. However, the lexicon partitioning method is too costly for modeling less common idiosyncrasies, at which point flag diacritics become a more practical approach again.

### Nominal Morphotactics

There are two main groups of suffixes that nominal structures may take, both of which are optional. The first part consists of the possessive, plural and case suffixes, whereas the second part is for derivational suffixes that possibly convert the noun stem to an adverb or verb stem, as seen in the example in çizelge 9.

Lexicon partitioning is used for noun stems as well, in order to formalize three exceptional groups of nouns that take on respectively dative, plural and possessive meanings as stems, without the need for the dative, plural and possessive suffixes. Samples from the first two groups are shown in çizelge 10 and çizelge 11.

The third group of noun stems is covered by a specific compound noun formation, which is made up of two nouns in a possessive relation fused together. Such compound nouns orthographically occur as a single word, but retain their possessive meanings, and are subject to a different possessive affixation paradigm. Samples from this group are shown in çizelge 12.

The fourth group of noun stems is the default group, with 3rd person plural agreement and in the nominative case by default. Samples from this group are shown in çizelge 13.

### Proper Noun Morphotactics

**Çizelge 11:** Examples of grammatically dative stems without an overt dative suffix.

<i>içeri</i> , "inward"	Noun+A3sg+Pnon+Dat
<i>dışarı</i> , "outward"	Noun+A3sg+Pnon+Dat
<i>aşağı</i> , "downward"	Noun+A3sg+Pnon+Dat
<i>yukarı</i> , "upward"	Noun+A3sg+Pnon+Dat

**Çizelge 12:** Possessive suffix ambiguity caused by an idiomatic usage.

<i>buzdolabı</i> +Noun+A3sg+Pnon+Nom	<i>buzdolabı</i> “refrigerator”
<i>buzdolabı</i> +Noun+A3sg+P3sg+Nom	<i>buzdolabı</i> “his refrigerator”

**Çizelge 13:** Default agreement, possession and case attributes in Turkish noun stems.

<i>masa</i>	Noun+A3sg+Pnon+Nom
<i>sandalye</i>	Noun+A3sg+Pnon+Nom
<i>cam</i>	Noun+A3sg+Pnon+Nom

Proper noun morphotactics are basically the same as noun morphotactics. However, Turkish requires inflectional suffixes added to proper noun stems (except for the plural suffix *-lar*) to be separated from the stem by an apostrophe character. Derivational affixes (as well as the plural suffix) are not subject to this rule. To model this orthographical phenomenon, we include duplicates of the relevant inflectional suffixes with apostrophes in the surface form in our lexicon, as shown in çizelge 14.

**Çizelge 14:** Apostrophe usage in the affixation of proper nouns.

İstanbul+'DA	İstanbul'da
Ankara+'yH	Ankara'yı
Ali+'DAn	Ali'den

### Pronoun Morphotactics

Pronouns, like adjectives and proper nouns, are a part of the nominal category. However, they have some differences from the noun-nominal root in affixation, and their analyses may return different tags. Furthermore, such differences also exist between the subdivisions of pronouns, such as personal (e.g. *ben*, “I”), reflexive (e.g. *kendi* “himself”), reciprocal (e.g. *birbiri*, “each other”), demonstrative (e.g. *ora*, “there”) and interrogative pronouns (e.g. *kim*, “who”). The subtle differences between the different kinds of pronouns could not be defined via flag diacritics, therefore they have been modeled on separate FSTs for all five kinds of pro-nouns.

In this thesis work, the morphological analyzer developed by Oflazer [5], which has been the state-of-the-art analyzer so far, has been taken as the baseline. Eventually, most deficient affix sequences have been allowed, and not only over-generation has been greatly mitigated, but also coverage has been increased by the finite-state transducer we developed using the flag diacritics method. Example comparison outputs seen in çizelge 3. The developed finite-state transducer is enabled to work both on XFST [2] and HFST [4], and made commonly available through a public web interface. ([tools.nlp.itu.edu.tr](http://tools.nlp.itu.edu.tr)) For some NLP systems previously using the output of Oflazer [5]’s morphological analyzer, we demonstrate the increases in extrinsic performances using the output of ITUMORPH instead. Additionally, an analyzer (BKA) with the ability to come up with analyses for unknown words is developed using our ITUMORPH. The unknown word analyzer is essentially an extension of the main analyzer, which makes use of wildcard entries that are able to morph into any phonologically valid Turkish stem, example outputs as seen in çizelge 2. As such, the analyzer derives the input word from a lexicon stem if possible, or backs off to the unknown stem expression if the lexicon did not contain a valid stem. During our evaluations, we showed the impact of our morphological analyzer on high level

NLP tools performances. (compared to using Ofizer's [5] analyzer) We observed that using ITUMORPH as the subtask of morphological disambiguation (MD) improves the average MD performance by %2,83. Similarly, during tthe coverage tests on the same data sets, it is observed that ITUMORPH improved the coverage by %10,12 and ITUMORPH used together with BKA improved the coverage by %12 in average.



## 1. GİRİŞ

Türkçe sondan eklemeli bir dil olup Altay dil ailesine ait bir dildir. Türkçe bir sözcüğün birden fazla ek alabilmesinden dolayı sözlükteki (lexicon) sözcük sayısı Türkçe’de türetilen kelime sayısından çok daha az olmaktadır. Bunun yanısıra, sondan eklemeli bir dil olması ve dil bilgisi kurallarının diğer dillere göre karmaşık bir yapıda olması Türkçe’nin morfolojik çözümlemesini de karmaşık hale getirmektedir. İnsanlar tarafından kullanılan dillerin anlaşılmasını ve analiz edilmesini amaçlayan doğal dil işleme yapay zekanın bir alt dalıdır. Doğal dil işlemenin günümüzdeki en popüler kullanım alanları arasında diller arasında çeviri, konuşma analizi, metinden sese, sestene metine dönüştürme ve insan-makine etkileşimi gibi konular yer almaktadır. Doğal dil işleme uygulamalarının en temel işlemlerinden biri biçimbilimsel çözümlemedir. Bu işlem, bir sözcüğün yapısının bilgisayar tarafından otomatik olarak çözümlenmesi işlemidir.

Tez çalışması kapsamında iki seviyeli bir biçimbilimsel çözümleyici geliştirilmiştir. İki seviyeli biçimbilimsel çözümleyici sayesinde hem geliştirilen tek bir sistem ile bir sözcüğün çözümlemesi yapılabilmekte hem de verilen bir çözümlemeden çıktı olarak ilgili sözcüğü sentezleyebilmektedir. İki seviyeli çözümleyiciler sonlu durumlu dönüştürücüler (FST) kullanılarak gerçekleştirilmektedir. Literatürde birçok FST yazılımı mevcuttur. Bunlardan bazıları Xerox’un XFST [2], Helsinki üniversitesinin HFST [4] Stanford üniversitesinin SFST yazılımlarıdır. Bu tez çalışmasında tasarlanan biçimbilimsel çözümleyici farklı FST teknolojileri (XFST ve HFST) üzerinde çalışabilir halde geliştirilmiştir.

İki seviyeli biçimbilimsel çözümleyici tasarımında, öncelikle dildeki ses olaylarını yani dilbilgisi kurallarını tanımlayan iki seviyeli kurallar ve dildeki sözcüklerin ve eklerin bulunduğu bir sözlük tasarlanması gerekir. Ses olayları bağlam duyarlı yeniden yazım kuralları (context sensitive rewrite rules) ile tanımlanmaktadır. Bu kurallar, hesaplamalı dil biliminde, düzenli ifadeler (regular expressions) ve bağlamdan bağımsız yeniden yazım kurallarına (Context sensitive rewrite rules) kıyasla daha

güçlü bir tanımlama biçimini mümkün kılar. Bu kurallar hazırlandıktan sonra iki seviyeli bir kural derleyicisi (Twolc - two level compiler) üzerinde derlenir. Sözcük gövdelerinin ve eklerin tutulduğu sözlük ise Lexc veya Xfst adı verilen derleyiciler ile derlenir. Xfst, Lexc'e göre daha yeni bir teknoloji olmasına ve oluşan sözlük üzerinde çeşitli düzenli ifadeler ile değiştirmeler yapılabilmesine rağmen derleme aşamasında Lexc'e göre daha yavaş çalışabilmektedir. Twolc kuralları ve sözlük tasarlandıktan sonra bu iki çalışmanın birleştirilmesi gerekir. Bu birleştirme işlemi de yine Lexc veya Xfst üzerinde yapılabilmektedir.

Xfst'nin bir özelliği olan bayrak işaretleri (flag diacritics) yöntemi ilk kez finite state morphology [2] isimli kitap ile 2003 yılında tanıtılmıştır. Bayrak işaretleri (flag diacritics) aslında sözcüklerin sonuna eklenen ve bazı ekleri alıp alamayacağını belirten işaretlerdir. Bu tez çalışmasında önceki çalışmalardan farklı olarak, sözlükler (lexicon) oluşturulurken bayrak işaretleri yöntemi işaretleri yöntemi yoğun şekilde kullanılmış ve bu sayede fazla üretimlerin önüne geçilmeye çalışılmıştır. Bu aşamada istisnai durumlar için elle eklemeler yapmanın yanı sıra otomatik bayrak işareti üretme ve sözlüğe ekleme yöntemleri de geliştirilmiş ve uygulanmıştır.

Biçimbilimsel çözümlemenin en ciddi sorunu bir sözcük için yapılan fazla analizlerdir. Örneğin "patlıcan" sözcüğünün çözümlemesinde "pat" kökünün sanki "lı" ve "ca" eklerini almış gibi bir çözümleme üretilmesi kuralsal olarak mümkün dahi olsa, Türkçe için geçersizdir. Bunun engellenmesi gerekir. Bahsedilen bayrak işaretleri yöntemiyle hem bir sözcük için çözümleme sonucundaki gereksiz fazla üretilen sonuçlar engellenmiş, hem de Türkçe için istisna olan durumların çözümü kolaylaşmıştır. Böylelikle biçimbilimsel çözümleyicinin çıktısındaki karmaşıklık önemli ölçüde azaltılmıştır.

Bu tez çalışması kapsamında Türkçe için bayrak işaretleri yöntemi kullanılarak sözcüklerin, sayıların ve noktalama işaretlerinin analizini yapabilen ITUMORPH isminde bir biçimbilimsel çözümleyici geliştirilmiştir. ITUMORPH ile birlikte çalışan, sözlükte olmayan bir sözcüğün analizinin yapılabilmesi için, sonlu durumlu makinedeki ek kümesine uygun bir şekilde eklerin sondan atılarak, olabilecek kelime kombinasyonlarına göre analiz üretebilen bir tanınmayan sözcük çözümleyicisi de tasarlanmıştır. Sözlükte bulunmayan bu tarz kelimelerin analizini yapabilen bu sistem bir biçimbilimsel çözümleyici için çok önemlidir. Dilin yaşayan bir varlık olması



nedeni ile, her gün yeni sözcüklerin dile girmesi söz konusudur. Bu neden ile derlenen bir sözlükte dildeki bütün sözcüklerin bulunması gerçekte neredeyse imkansızdır. Öte yandan, Türkçe gibi bir dilde kullanılan ek kümesi büyük olmasına rağmen sınırlı sayıdadır ve gelişmeye açık değildir. Dile yeni giren sözcükler sözlükte yer almasalar dahi, Türkçe ekler ile çekimlenerek kullanıma geçebilirler (örneğin “serverlar”, “opsiyonsuz”). Bu nedenle, çözümlemesi istenen sözcük sözlükte yoksa, en azından aldığı ek dizilimine bakılarak bir çözümlemenin üretilmesi gerekmektedir. Bilinmeyen kelime analizcisi (BKA) ismini verdiğimiz bu sistem girilen sözcük sözlükte yoksa bile bir analizin elde edilmesini sağlar. Bu sayede biçimbilimsel çözümleyicinin özellikle sosyal medya gibi bu tür durumların çok görüldüğü ortamlarda kapsamının artması sağlanır. Bunun yanı sıra BKA'nın bir avantajı da, sözlükte bulunması gereken ancak henüz eklenmemiş olan sözcüklerin tespitini kolaylaştırması ve biçimbilimsel çözümleyicinin zaman içerisinde iyileştirilmesini kolaylaştırmasıdır. Bu şekilde sözlüğün hızlı bir şekilde genişletilmesine de yardımcı olur. Tasarlanan biçimbilimsel çözümleyici farklı FST teknolojileri (XFST - Xerox Finite State Transducer [2], HFST - Helsinki Finite State Transducer [4]) üzerinde çalışabilir hale getirilmiştir. Buna ek olarak akademik çalışmalara hizmet etmesi amacıyla tools.nlp.itu.edu.tr adresinden bir web servisi olarak hizmete açılmıştır.

### **1.1 Tezin Amacı Nedir?**

Yüksek lisans tezi kapsamında, Türkçe için ITUMORPH isminde bir biçimbilimsel çözümleyici geliştirilmiştir. Bu alanda daha önce geliştirilen biçimbilimsel çözümleyicilerden farkı, bir sözcük için üretilen analizlerin arasında fazla gereksiz analizlerin engellenmesi, sözlükte yapım ekleri üzerinde yapılan geliştirme, sayılar ve noktalama işaretleri için analiz üretme ve sözlükte bulunmayan sözcüklerin de analizini yapabilmesidir. Buna ek olarak akademik çalışmalara hizmet etmesi amacıyla tools.nlp.itu.edu.tr adresinden bir web servisi olarak hizmete açılmıştır. analiz yapılması sağlanmıştır.

### **1.2 Literatür Araştırması**

Öncelikle Türkçe için varolan biçimbilimsel çözümleyiciler incelenerek, bunların yetersiz ve sorunlu görülen yönleri belirlenmiştir. Bunlardan bazıları Hankamer

ve Jorge [6], Oflazer [5], Eryiğit ve Adalı [7], Akın ve diğ. [9], Say ve diğ. [14], Çoltekin [10], Sak ve diğ. [15]'nin çalışmalarıdır. Bu çalışmaların birtakım özellikler bakımından karşılaştırması Yılmaz [1]'in çalışmasında yapılmıştır. Çizelge 1.1'de bu karşılaştırma incelenebilir. Eryiğit ve Adalı [7]'nin hazırladığı sözlüksüz köke ulaşma yöntemi, sonlu durumlu makineleri ters çevirip Türkçe dilbilgisi kurallarına göre dilimizde geçerli olabilecek kökleri (root) bulmakta ve sözcüğü biçimbilimsel olarak ayrıştırmaktadır. Oflazer [5]'in çalışması Xfst'de hazırlanmış olup iki seviyeli biçimbilimsel (two level morphology) çözümlmeye dayanır. Burada Xerox'un Xfst aracının kodları açık kaynak kodlu olmadığından çözümleyicinin kodlarına erişilemez. Rakam içeren sayıların analizinin yapılamaması kapsam başarımını etkilemektedir. Bununla birlikte, yapım eki almış olan sözcükler için yaptığı fazla analizler (overgeneration) çözümleyicinin başarımını etkilemektedir. Türkçe için hazırlanan bir diğer biçimbilimsel çözümleyici de Zemberek [9] aracıdır. Zemberek Türkçe için hazırlanmış platformdan bağımsız, genel amaçlı bir doğal dil işleme kütüphanesi ve araç kümesidir. Ürettiği analizlerdeki etiketler uluslararası standartlara uymamaktadır. Ayrıca türemiş sözcüklerin aldığı sınırlar (Derivational Boundary) analizde belirtilmemiştir. Sak ve diğ. [15]'nin yaptığı çalışma herhangi bir dış programa bağlı olmayacak şekilde tasarlanmış olup TDK'nın sözlüğünden yararlanılmıştır. Analizlerindeki etiketler Petrov ve diğ [3]'nin etiket kümesine çok benzerlik göstermektedir. Yılmaz [1]'in çalışmasında TDK'nın sözlüğünden yararlanılarak uluslararası etiket kümesi kullanılarak biçimbilimsel çözümleyici tasarlanmıştır. Ancak java'da hazırlanmasından dolayı yavaş çalışmaktadır. Bu tez çalışmasında, Yılmaz [1], Oflazer [5], Sak ve diğ. [15] ve Zemberek [9] sistemleri üzerinde görülen aksaklıklar ayrıntılı olarak incelenmiş ve çözüm önerileri getirilmiştir. Bu çalışmalar arasında, Türkçe için en çok ses getiren ve bugüne kadar en çok kullanılan biçimbilimsel çözümleyici Oflazer [5]'in 1994'te PC-KIMMO üzerinde yaptığı daha sonra Xerox sonlu durumlu araçları kullanarak geliştirdiği biçimbilimsel çözümleyicisidir. Oflazer [5]'de de ITUMORPH'da olduğu gibi ses kuralları Koskeniemi [16]'nin iki seviyeli kuralları kullanılarak, morfotaktik kurallar (eklerin yapı ve sıralanış kuralları) ise sonlu durumlu makineler kullanılarak tanımlanmıştır. Oflazer [5]'in kullandığı sözlüğün de bir sonlu durumlu dönüştürücü şeklinde tasarlanmış olması ve kullanılan derleyicinin özellikleri sayesinde oldukça hızlı bir uygulama olduğu söylenebilir. Oflazer [5]'de bazı sözcükler için çözümleyicinin

fazladan gereksiz analiz üretmesi, analizini yapabildiği sözcük sayısının (covarage) az olması ve kaynak kodlarının Xerox Xfst'den dolayı açık kaynak kodlu olmayışı bizi ITUMORPH'u yapmaya yönlendirmiştir. ITUMORPH'da uluslararası bir etiket kümesi olan Petrov ve diğ. [3]'nin hazırladığı etiket kümesi kullanılmıştır. ITUMORPH'u diğerlerinden ayıran en önemli özelliği, bayrak işaretleri kullanımı, yapım ekleri üzerinde yapılan fazla analizleri engelleyen çalışma ve sözlükte bulunmayan sözcüklerin analizini yapabilen bir sistemle her sözcüğe aldığı eklere göre bir analiz üretmesidir. İTU ağaç yapılı derlemi [17], OdtüSabancı ağaçyapılı derlemi [18], Dilek ve diğ. [19]'nin kullandığı veri kümesi üzerinde karşılaştırmalı testler yapılmış olup sonuçlar son bölümde verilmiştir.

**Çizelge 1.1:** Yılmaz [1]'in Karşılaştırma Tablosu

Özellik	Kemal Oflazer	Zemberek	Haşim Sak	ITUMORPH
Hangi ekin eklendiği gösterme	-	-	-	-
Türetim sınırlarını gösterme	+	+	+	+
Türkçe etiket üretme	-	+	-	-
Türkçe harf içeren sözcükleri çözümlenme	+	+	+	+
Sözlükte yer almayan sözcükler için sonuç üretme	-	-	-	+
Sayılar için sonuç üretme	-	-	+	+
Noktalama işaretleri için sonuç üretme	-	-	-	+
Özel isimlere sonuç üretme	+	+	+	+
Platformdan bağımsız	+	+	-	+

### 1.3 İçerik

Tezin başlangıcında ve giriş kısmında doğal dil işleme ile ilgili temel bilgiler verilmiş ve literatür araştırması sunulmuştur. İkinci kısımda, Türkçe'nin dilbilgisi kurallarından, istisna durumlarından, ses ve yapım özelliklerinden bahsedilmiştir. Üçüncü kısımda Türkçe için hazırlanan Twolc kurallarından, yapım ekleri üzerinde yapılan çalışmadan, dördüncü kısımda bayrak işaretlerinden sözlükte bulunmayan kelimelerin, sayıların ve noktalama işaretlerinin analizinden, beşinci kısımda yaptığımız testlerin sonuçlarından, altıncı ve son kısımda sonuç ve önerilerden bahsedilmiştir.



## 2. TÜRKÇE'NİN YAPISI

### 2.1 Türkçenin Temel Özellikleri

Ural-Altay dil grubuna giren Türkçe kural tabanlıdır ve sondan eklemeli bir dildir. Özellikle Orta Asya bölgesinde konuşulan ve çok geniş bir coğrafyaya hitap eder. Türkçe kendi öz Türkçe sözcüklerden oluşmasının yanında özellikle Arapça ve Farsçadan gelen sözcüklerle kendi sözlüğünü oluşturmuştur. Türkçe'deki cümleler özne yüklem ve tümleş gibi sözcük öbeklerinin birleşmesiyle oluşur. Bu bölümde Türkçe'nin dilbilgisi ve ses olayları tezin kapsamına girdiği kadarıyla anlatılmıştır.

### 2.2 Ses Bilgisi

Türkçe'deki harfler sesli ve sessiz olmak üzere iki ana gruba ayrılır ve toplamda 29 harften oluşmaktadır. Sesli harfler kendi arasında düz-yuvarlak, dar-geniş, kalın ince olmak üzere üç gruba ayrılmıştır. Sessiz harfler ise sert-yumuşak, sürekli-süreksiz gibi gruplara ayrılmaktadır. Bu harfler sözcüklerin oluşumunda birtakım kurallara göre birlikte bulunmaktadır. Örnek olarak

çocuk+da -> çocukta, kuş+da -> kuşta gibi durumlar verilebilir. Bu bölümde bu kuralları inceleyeceğiz.

#### 2.2.1 Ünlülerin Özellikleri

Ünlüler dudak, çene ve dilin aldıkları duruma göre üç gruba ayrılır. Çizelge 2.1'de ünlü harfler gösterilmiştir.

- Düz Yuvarlak Ünlüler
- Geniş Dar Ünlüler
- Kalın İnce Ünlüler

**Çizelge 2.1: Sesli Harfler**

	<b>Düz</b>		<b>Yuvarlak</b>	
	<b>Geniş</b>	<b>Dar</b>	<b>Geniş</b>	<b>Dar</b>
<b>Kalın</b>	a	ı	o	u
<b>İnce</b>	e	i	ö	ü

### **2.2.2 Büyük Ünlü Uyumu**

Büyük ünlü uyumu Türkçe'ye özgü bir özelliktir ve ince ünlülerden sonra ince, kalın ünlülerden sonra kalın bir ünlünün gelmesi kuralıdır. Örnekeleyecek olursak masa, bardak, telsiz, kablo gibi kelimeler büyük ünlü uyumuna uyar ancak kalem, ekran, fare gibi kelimeler ise bu kurala uymazlar. Ayrıca bileşik sözcükler de bu kural aranmaz. Türkçe'deki eklerden bazıları bu kurala uyum sağlarken bazı ekler ise bu kurala uymaz.

### **2.2.3 Küçük Ünlü Uyumu**

Türkçe'de yuvarlak ünlülerden sonra geniş düz ya da dar yuvarlak, düz ünlülerden sonra düz ünlülerin gelmesine küçük ünlü uyumu denir. Örnek verecek olursak anlam, elemek, ısınmak, öğretmen gibi kelimeler küçük ünlü uyumuna uyar.

### **2.2.4 Ünlü Düşmesi**

Türkçe'deki bir sözcüğe herhangi bir ekin gelmesi sırasında ya da bazı bileşik isimlerin oluşumu sırasında sözcüğün yapısındaki bir ünlü düşer, buna ünlü düşmesi (hece düşmesi) denir. Örnek olarak ağız -> ağızı, alın -> alını, göğüs -> göğsü, cuma+ertesi -> cumartesi, hapis+etmek -> hapsetmek gibi sözcükler verilebilir. Türkçe'de sadece sözcüğün kendi yapısındaki bir ünlünün düşmesi yanında gelen ekteki ünlünün düşmesi de sıkça gerçekleşir. Örneğin anne+im -> annem, kedi+imiz -> kedimiz.

### **2.2.5 Ünlü Daralması**

Türkçe'de son harfi "a", "e" olan bazı sözcükler "+yor" eki aldığıında bu sözcükteki "a", "e" harfi daralarak "ı", "i", "u", "ü" olur, bu kurala ünlü daralması denir. Örneğin anla+yor -> anlıyor, söyle+yor -> söylüyor.

## 2.2.6 Ünsüz Harfler Özellikleri ve Ses Olayları

Türkçe alfabede 21 tane ünsüz harf bulunmaktadır. Bu harfler "b", "c", "ç", "d", "f", "g", "ğ", "h", "j", "k", "l", "m", "n", "p", "r", "s", "ş", "t", "v", "y", "z" dir. Çizelge 2.2'de sessiz harflerin grupları gösterilmiştir.

**Çizelge 2.2:** Sessiz Harfler

	Süreksiz	Sürekli
Sert	p,ç,t,k	f,h,s,ş
Yumuşak	b,c,d,g	ğ,j,l,m,n,r,v,y,z

## 2.2.7 Ünsüz Benzeşmesi

Türkçe'de sert ünsüzlerden sonra sert, yumuşak ünsüzlerden sonra yumuşak ünsüz gelmesi kuralına ünsüz benzeşmesi denir. Türkçe'de ekler bu kurala genellikle uyar. Bu kural genellikle sözcük içinde ve sözcük sonunda uygulanır ancak dört yerde görülür.

- Sözcük içinde
- Sözcük sonunda
- Çekim eklerinde
- Ad durum eklerinde

Ancak tez kapsamında sözcük içinde gerçekleşen ünsüz benzeşmesi ele alınmıştır. Örnek olarak sınıf+da -> sınıfta, çocuk+da -> çocukta, kuş+da -> kuşta gibi sözcükler verilebilir.

## 2.2.8 Ünsüz Düşmesi

Türkçe sözcüklerde bazı ünsüzlerin kaybolmasına ünsüz düşmesi denir. Bu durum genellikle bir sözcük ek aldığı anda görülür.

- sıcak+cık -> sıcacık
- küçük+cük -> küçücük,

- çabuk+cak -> çabucak
- büyük+cek -> büyücek

### 2.2.9 Ünsüz Türemesi

Türkçe'de bazı sözcükler sesli ile başlayan ek aldığıında sondaki ünsüz harf iki kez yazılır buna ünsüz türemesi denir. Ancak Türkçe'de ünsüz türemesi pek yaygın değildir.

- af + ı -> affi
- hak + ı -> hakkı

### 2.2.10 Ünsüz Yumuşaması

Türkçe sözcüklerin sonunda bulunan "p", "ç", "t", "k" ünsüzlerinden sonra ünlü ile başlayan bir ek geldiğinde p->b, ç->c, t->d, k->ğ' ye dönüşür bu duruma ünsüz yumuşaması denir.

- kitap + ı -> kitabı
- çocuk + u -> çocuğu
- yurt +u -> yurdu

## 2.3 Kök

Sondan eklemeli bir dil olan Türkçe'deki sözcükler kök ve eklerden oluşur. Sözcükler genellikle bir ya da birkaç heceden oluşurlar ve sonuna eklenen ekler çıkarıldığında kök bulunur. Türkçe'deki sözcük türleri ad, sıfat, adıl, belirteç, ilgeç, bağlaç, ünlem ve eylem olmak üzere 8 türe ayrılır. Bu türlerin ilk yedisi ad kökündendir. Türkçe'deki sözcük türleri arasındaki en önemli olanları ad ve eylem kökleridir. Soyut, somut varlıkları ve onların nitelik nicelik durum ve ilişkilerini gösteren kökler ad soylu köklerdir. "Masa", "kalem", "fare", "fil" bu köklere örnek olarak verilebilir. Herhangi bir olayı hareketi anlatan sözcükler ise eylem kökündendir. Tanımları bu şekilde olsa da bu türleri birbirinden ayırmak çok da kolay olmayan bir durumdur. Çünkü normalde ad kökü olan bir sözcük bir yapım eki olarak eyleme ya da eylem olan bir sözcük bir ek



olarak ad haline dönüşebilir. Bu yüzden sözcüklerin ancak cümle içinde kullanıldıkları yere anlamına ve işlevine bakarak karar vermek daha kolaylaşır. Bizim çalışmamızdan önce yapılan çalışmalarda sıfatlar ad grubundan kabul edildiğinden sonlu durumlu makineler ad grubundaki sözcüklerin aldıkları ekler göre düzenlenmiştir. Ancak bu tez çalışmasında sıfatlar cümlelerin içinde kullanıldığı duruma göre değişebileceğinden dolayı herhangi bir eki aldığında o ekin eklendiği sözcüğü ad mı yoksa eylem mi yaptığına bakarak sıfatlara özgü sonlu durumlu makine tasarlanmıştır.

## **2.4 Ekler**

Ekler yapı bakımından sözcüklere önek, içek ve sonek olmak üzere üç şekilde eklenmektedir. Türkçe ise sondan eklemeli dil olduğundan sözcüklerin köklerine çoğu zaman sonek alırlar. Eklerin kendi başına bir anlamı olmadığı gibi sadece sözcüğün köküne eklendiğinde kökün anlamını değiştirerek anlam kazanır. Ekler Türkçe’de temelde yapım ekleri ve çekim ekleri olmak üzere iki ana gruba ayrılır. Bir sözcüğe birden fazla ek gelebilir bunun yanısıra bir sözcüğe hem yapım hem de çekim eki gelecektse yapım ekleri çekim eklerinden daha önce gelir. Fiil kökü olan sözcüklere gelen eklerle isim köküne gelen ekler birbirinden hem yapım ekleri hem de çekim ekleri olarak farklıdır. Türkçe kök bakımından oldukça zengin bir dil olduğu gibi ek bakımından da oldukça zengin bir dildir. Bu yüzden bir sözcüğün köküne gelebilecek ek kombinasyonlarıyla yüzlerce farklı kelime türetilmektedir.

### **2.4.1 Yapım Ekleri**

Yapım ekleri isimden isim, fiilden fiil, fiilden isim, isimden fiil yapan ekler olmak üzere dört temel grupta incelenir. Yapım ekleri eklendikleri sözcüğün yada kökün anlamını değiştirerek farklı bir sözcük haline getiren ek grubudur. Türkçe yapım ekleri bakımından çok zengin bir dildir. Ad ve eylemlere gelen eklerle binlerce yeni sözcük türetmek mümkündür.

### **2.4.2 Addan Ad Yapan Ekler**

Ad soylu bir sözcüğün yada kökün sonuna gelerek başka bir ad soylu sözcük yapan ekler addan ad yapan yapım eki denir. Genellikle adlardan sıfat, belirteç ve ad yaparlar. Bizim çalışmamızda sonlu durumlu makineleri oluştururken eğer ad soylu bir

sözcük yapım eki aldığı halde de sözlükte bulunuyorsa bu ekten türetilen analiz engellenmiştir. Çünkü zaten sözcüğün yapım eki almış hali Türkçe’de yaygın halde kullanıldığından bu şekilde doğabilecek fazla analizlerin önüne geçilmiştir. Addan ad yapan eklere örnekler çizelge 2.3’te verilmiştir.

**Çizelge 2.3:** Addan Ad Yapan Yapım Ekleri

<b>+cH</b>	<b>Boya+cH -&gt; Boyacı</b>
<b>+IHk</b>	<b>Kömür+IHk -&gt; Kömürlük</b>
<b>+çHI</b>	<b>Balık+çHI -&gt; Balıkçıl</b>
<b>+tAş</b>	<b>Meslek+tAş -&gt; Meslektaş</b>
<b>+IH</b>	<b>Akıl+IH -&gt; Akıllı</b>
<b>+cHk</b>	<b>Kitap+cHk -&gt; Kitapçık</b>

Çizelge 2.3’teki boya sözcüğü "cı" ekini alarak farklı bir sözcük olan boyacı sözcüğü haline gelmiştir. Benzer şekilde kömürden kömürlük, balıktan balıkçıl, meslekten meslektaş gibi sözcükler türetilmiştir.

#### 2.4.3 Eylemden Ad Yapan Ekler

**Çizelge 2.4:** Eylemden Ad Yapan Yapım Ekleri

<b>+AcAk</b>	<b>İç+AcAk -&gt; İçecek</b>
<b>+An</b>	<b>Bak+An -&gt; Bakan</b>
<b>+mA</b>	<b>As+mA -&gt; Asma</b>
<b>+mAz</b>	<b>Çık+mAz -&gt; Çıkmaz</b>
<b>+Hr</b>	<b>Düşün+Hr -&gt; Düşünür</b>

Eylem soylu bir sözcük ya da kökten ad yapan eklere eylemden ad yapan ek denir. Çizelge 2.4’te örnekleri verilmiştir. Mesela buradaki iç kökü "ecek" ekini alarak başka bir sözcük haline gelmiştir.

#### 2.4.4 Eylemden Eylem yapan ekler

Eylem soylu bir sözcük ya da köke gelerek yine eylem soylu bir sözcük türeten eklere eylemden eylem türeten ek denir. Çizelge 2.5 te örnekler gösterilmiştir.

#### 2.4.5 Addan Eylem Yapan Ekler

Ad soylu bir sözcüğün ya da kökün sonuna gelerek bu sözcüğü eylem soylu bir sözcük haline getiren eklere addan eylem yapan ek denir. Çizelge 2.6’da örnekler verilmiştir.

**Çizelge 2.5:** Eylemden Eylem Yapan Yapım Ekleri

<b>+t</b>	<b>Ağla+t -&gt; Ağlat</b>
<b>+Hl</b>	<b>Yaz+Hl -&gt; Yazıl</b>
<b>+Hn</b>	<b>Öv+Hn -&gt; Övün</b>
<b>+Hr</b>	<b>Bit+Hr -&gt; Bitir</b>
<b>+dHr</b>	<b>Aç+dHr -&gt; Açtır</b>
<b>+Hş</b>	<b>Döv+Hş -&gt; Dövüş</b>

**Çizelge 2.6:** Addan Eylem Yapan Yapım Ekleri

<b>+lAn</b>	<b>Umut+lAn -&gt; Umutlan</b>
<b>+lA</b>	<b>Av+lA -&gt; Avla</b>
<b>+lAş</b>	<b>Taş+lAş -&gt; Taşlaş</b>

#### **2.4.6 Çekim Ekleri**

Ad ya da eylem soylu sözcüklere, köklere gelerek onlara soru, zaman, çoğul, kişi, olumluluk, olumsuzluk gibi anlamlar katan ekler çekim eki denir. Yapım ekinden farkı eklendiği sözcüğün yapısını, anlamını değiştirmemesidir; yani ad soylu bir sözcüğe ekleniyorsa sözcük bu eki aldıktan sonra yine ad soylu bir sözcük olarak kalır. Ad ve eylem soylu sözcüklere gelen çekim ekleri olmak üzere iki ana grupta incelenirler.

#### **2.4.7 Ad Soylu Sözcüklere Gelen Çekim Ekleri**

Ad soylu sözcüklere gelen çekim ekleri, iyelik, hal, ilgi, soru ve çoğul eki olmak üzere 5 bölümde incelenir.

#### **2.4.8 İyelik Ekleri**

İyelik ekleri eklendiği sözcüğe kişi kavramını katar ve ünlü uyumuna uyar. Çizelge 2.7'de sözcüklere gelen iyelik ekleri incelenebilir.

#### **2.4.9 Hal Ekleri**

Ad soylu sözcüklerin anlamını, kime, nereye ait olduğunu ve yer, yön bakımından tamamlar. Çizelge 2.8'de örnekleri verilmiştir.

**Çizelge 2.7: İyelik Ekleri**

<b>Benim</b>	<b>Masa-m</b>	<b>Araba-m</b>
<b>senin</b>	<b>Masa-n</b>	<b>Araba-n</b>
<b>onun</b>	<b>Masa-sı</b>	<b>Araba-sı</b>
<b>bizim</b>	<b>Masa-mız</b>	<b>Araba-mız</b>
<b>sizin</b>	<b>Masa-nız</b>	<b>Araba-nız</b>
<b>onların</b>	<b>Masa-ları</b>	<b>Araba-ları</b>

**Çizelge 2.8: Hal Ekleri**

<b>Belirtme durumu -ı-i-u-ü</b>	<b>Araba-yı</b>	<b>Okul-u</b>
<b>Yönelme durumu -a -e</b>	<b>Araba-ya</b>	<b>Okul-a</b>
<b>Kalma durumu -da -de -ta -te</b>	<b>Araba-da</b>	<b>Okul-da</b>
<b>Çıkma durumu -dan-den -tan -ten</b>	<b>Araba-dan</b>	<b>Okul-dan</b>
<b>Tamlayan durumu m-in -un -ün</b>	<b>Araba-n</b>	<b>Okul-un</b>

#### **2.4.10 İlgî Eki**

İlgî eki eklendiği sözcüğe aidiyet kavramı kazandıran ve ad soylu sözcüklere gelen bir ektir.

- Sokakta-ki -> Sokaktaki
- Evde-ki -> Evdeki
- Yukarıda-ki -> Yukarıdaki

#### **2.4.11 Soru Eki**

Ad ve eylem soylu sözcüklere gelen "mı", "mi", "mu", "mü" ekidir. Eklendiği sözcüğe soru anlamı katar ve birleşik yazılmaz ayrı yazılır.

- Kızınız mı?
- Arabanız mı?
- Geldiniz mi?

#### 2.4.12 Çoğul Eki

Eklendiği sözcüğe çoğul anlamı katar ve ses uyumu kuralına uygun bir formda eklenir.

- Kitap-lar -> Kitaplar
- Araba-lar -> Arabalar
- Kalem-ler -> Kalemler

#### 2.4.13 Eylemlere Gelen Çekim Ekleri

Eylem soylu sözcüklere gelen ekler zaman, dilek koşul, gereklilik, çoğul ,soru ve emir kipi olmak üzere 6 grupta incelenir.

#### 2.4.14 Zaman Eki

Zaman eki olarak şimdiki, geçmiş, gelecek, geniş zaman gibi formları vardır. Çizelge 2.9'de örnekleri verilmiştir.

Çizelge 2.9: Zaman Ekleri

Kişi	Geçmiş zaman	Şimdiki zaman	Gelecek zaman	Geniş zaman
Ben	Gel-di-m	Gel-iyor-um	Gel-ecek-im	Gel-ir-im
Sen	Gel-di-n	Gel-iyor-sun	Gel-ecek-sin	Gel-ir-sin
O	Gel-di	Gel-iyor	Gel-ecek	Gel-ir
Biz	Gel-di-k	Gel-iyor-uz	Gel-ecek-iz	Gel-ir-iz
Siz	Gel-di-niz	Gel-iyor-sunuz	Gel-ecek-siniz	Gel-ir-siniz
Onlar	Gel-di-ler	Gel-iyor-lar	Gel-ecek-ler	Gel-ir-ler

#### 2.4.15 Dilek Koşul Kipi

Dilek şart eki eylem soylu sözcüğe bir istek yada şart anlamı kazandırır.

- bil-se-m ->bilsem
- bil-se-n ->bilsen
- gel-se ->gelse

#### **2.4.16 Gereklilik Eki**

Eylem soylu sözcüklere eklenen "meli", "malı" eki gereklilik ekidir.

- Bil-meli-yim -> Bilmeliyim
- Gel-meli-sin -> Gelmelisin
- Konuş-malı -> Konuşmalı

#### **2.4.17 Emir Kipi**

Eylem soylu sözcüğe emir anlamı katan ektir ancak ikinci tekil şahıs için herhangi bir ek almasına gerek yoktur. Yani eylemin yalın hali emir kipindedir.

- Bil-sin-> Bilsin
- Gel-sin-ler-> Gelsinler
- Konuş

#### **2.4.18 Çoğul Eki**

Eylem soylu sözcüklere eklenerek bu sözcüklerin çoğul anlam taşımalarını sağlayan ektir.

- Gel-miş-ler-di -> Gelmişlerdi
- Gel-ecek-ler -> Gelecekler
- Başla-mış-lar -> Başlamışlar
- Al-mış-lar -> Almışlar

#### **2.4.19 Soru Eki**

Eylem soylu sözcüğe eklenerek sözcüğe soru anlamı katan ve ayrı yazılan ektir.

- Görmüş müydü?
- Vermeyecek misiniz?

### 3. ITUMORF MİMARİSİ

Giriş bölümünde anlatıldığı üzere, biçimbilimsel çözümleme, bir girdi sözcük için olası kök ve ek dizilimlerinin oluşturulması aşamasıdır. Örneğin "araba" sözcüğü için geçerli olabilecek arap ve araba kökleri ve bunlara eklenecek ekler aşağıda belirtilmektedir. Birinci analizde arap +Noun kısmı arap sözcüğünün isim kökünden olduğunu +A3sg kısmı tekil durumda olduğunu, +Pnon iyelik eki almadığını ve +Dat kısmı ise yönelme eki olan 'a' ekini aldığını belirtmektedir. İkinci analiz ise yalın alindeki araba sözcüğünün analizini temsil etmektedir.

- “araba arap +Noun+A3sg+Pnon+Dat”
- “araba araba +Noun+A3sg+Pnon+Nom”

Türkçede sözcükler yapım eklerini alarak buldukları sözcük kökünden başka bir sözcük grubuna geçebilirler. Bir diğer örnek aşağıda "Arabadayım" sözcüğü için verilmektedir. Burada araba +Noun kısmı araba sözcüğünün isim kökünden olduğunu +A3sg tekil olduğunu +Pnon iyelik eki almadığını +Loc kısmı bulunma eki olan "da" ekini "DB+Verb+Zero+Pres+A1sg" kısmı ise sözcüğün fiil hale geçtiğini gösterir. Bu aşamadan sonra sözcük fiil çekim ekleri almaya başlamıştır. "+Pres+A1sg" kısmı tekil, geniş zaman durumunda olan sözcüğü temsil eder.

- arabadayım araba +Noun+A3sg+Pnon+Loc+DB+Verb+Zero+Pres+A1sg

Eğer sözcüğe eklenen ek yapım eki ise "DB" etiketi mutlaka analizde bulunur. "DB" den sonraki kısımda sözcüğün aldığı yapım eki, sözcüğü hangi sözcük grubuna dönüştürdüğünü, en sondaki kısım ise ekin gerçek etiketini temsil eder. Çizelge 3.1 ve çizelge 3.2’de bu eklere karşılık gelen etiketler incelenebilir.

- kocaman koca +Verb+Pos+DB+Noun+Inf2+A3sg+P2sg+Nom
- toplumun top +Noun+A3sg+Pnon+Nom+DB+Adj+With+DB+Noun+Zero+A3sg+P1sg+Gen

Biçimbilimsel çözümlene yapan sistemlerin en büyük problemlerlerinden biri, bir girdi için yapılan gereksiz fazla analizlerdir. Çizelge 3.3'deki örnekler fazla analiz örneği olarak verilebilir. Kocaman sözcüğü normalde sıfat olan bir sözcüktür. Bu sözcüğün analizleri arasında koca sözcüğünün önce "ma" ekini ardından ikinci tekil şahıs eki olan "n" ekini almış gibi analiz edilmesi doğru bulunmamıştır. Bu tarz durumlarda bayrak işaretleri kullanarak analiz sayısı indirgenmeye çalışıldı. Analizdeki etiketlere bakılırsa, "koca +Verb+Pos" kısmı sözcüğün kökünün pozitif bir fiil kökü olduğunu, "+DB+Noun+Inf2" kısmı koca sözcüğünün sonuna eklenen "ma" ekini, "+A3sg+P2sg+Nom" kısmı ise yalın halde ikinci tekil şahıs eki olan "n" ekini temsil eder. Toplumun sözcüğündeki analizde ise top sözcüğünün "lu" ekini sonrasında ise "m" ve "un" eklerini almış olması sonlu durumlu makine için olası bir durum olsa da analizde engellenmesi gerekir. Bu şekildeki bir analiz hem yanlış hem de fazla analiz örneğidir.

Fazla analizin bir diğer sebebi aynı kök grubuna (örneğin isim kökü) sahip sözcüklerin sonlu durumlu makinede her zaman aynı ekleri almasıdır. Örnek olarak, "Müslümanlık" sözcüğünün analizi aşağıda verilmektedir.

- "müslüman+Noun+A3sg+Pnon+Nom+DB+Noun+Ness+A3sg+Pnon+Nom"

Burada "müslüman" isim köküdür ve analizde "+Noun (Noun)" kısmı bunu belirtir. "DB+Noun" kısmı sözcüğe eklenen ekin bu sözcüğü isim soylu bir sözcüğe dönüştürdüğünü yani isimden isim yapan bir ek aldığını, "Ness" kısmı ise sözcüğe eklenen "lık" yapım ekini temsil eder. Burada analizde herhangi bir sorun görünmemektedir. Çünkü "müslümanlık" sözcüğünün "müslüman" sözcüğünden türemesi doğrudur.

- ter +Noun+A3sg+Pnon+Nom+DB+Noun+Ness+A3sg+Pnon+Nom
- terlik +Noun+A3sg+Pnon+Nom

Ancak yukarıdaki analizde de görüldüğü gibi isim kökünden gelen "terlik" sözcüğünün analizinin<sup>1</sup> "ter" sözcüğünden türemesi yanlıştır. Burada "ter" isim köküdür ve analizde "+Noun (Noun)" kısmı bunu belirtir. "DB+Noun" kısmı sözcüğe eklenen

<sup>1</sup>örnek analiz <http://open.xerox.com/Services/fst-nlp-tools/Consume/176> adresindeki Oflazer [5] biçimbilimsel çözümleyicisinden alınmıştır.



ekin bu sözcüğü isim soylu bir sözcüğe dönüştürdüğünü yani isimden isim yapan bir ek aldığını, "Ness" kısmı ise sözcüğe eklenen "lık" yapım ekine karşılık gelir. Bu analizde terlik sözcüğünün ter sözcüğünden türemiş olması doğru değildir. Bunun engellenmesi amacıyla "ter" sözcüğünün sonuna "lık" ekini alamamacayak şekilde yerleştirilen işaretlere bayrak işaretleri denir. Xerox [2]'un geliştirdiği bu teknoloji bu şekilde bir çok durumun çözümünde kullanılmıştır. Bu tez çalışması kapsamında bir kelime sonlu durumlu makineye girdi olarak verildiğinde üretilen analizlerin hem doğru olması hem de gereksiz yere üretilen birtakım analizlerin (overgeneration) azaltılması ve bu sayede morfolojik analizin üstünde çalışan sistemlerin başarımının artırılması amaçlanmıştır. Morfolojik analiz yapan bir sistem temelde sözlük (lexicon) ve dil ile ilgili birtakım kurallardan oluşur.

Biçimbilimsel analiz yapan sistemler genellikle kural tabanlı sistemlerdir ve başarımları sözlüğün (lexicon) genişliğine bağlıdır. Kural tabanlı sistemler, üst seviye (surface level) ve alt seviye (lexical level) olmak üzere iki seviyeden oluşan sistemlerdir. Sözcüğün sözlükteki hali alt seviye, biçimbilimsel çözümleyiciye girdi olarak verilen kısmı üst seviyedir. Örnek olarak arabam sözcüğü alt seviyede (Lexical level) araba+Hm şeklinde tutulmaktadır. Burada alt seviyede araba+Hm ifadesi üst seviyede araba00m şekline dönüşür. Yani "+" ve "H" karakterleri '0'a dönüşmüştür kaybolmuştur.

- Alt seviye: masa+DA
- Üst seviye: masa0da
- Alt seviye: ak\$ıl+sH
- Üst seviye: ak00l00ı

Verilen örneklerde + karakterinden sonraki kısım sözcüğün aldığı ek kısmıdır. masa+DA örneğinde + karakteri üst seviyede 0 a dönüşmüş yani kaybolmuştur. "D" karakteri "d" karakterine "A" karakteri "a" karakterine dönüşmüştür. Akıl sözcüğü sesli ile başlayan bir ek aldığında sözcükteki "ı" harfi düşeceğiinden dolayı bu harften önce \$ karakteri koyulmuş ve düşmesi gereken harf belirtilmiştir. "D" karakteri üst seviyede hem "d" hem de "t" karakterini temsil eder. "A" karakteri üst seviyede

hem "a" hem de "e" karakterini temsil eder. "H" karakteri ise ı,i,u,ü karakterlerini temsil eder. Bu dönüşümle ilgili detay aşağıdaki listede verilmiştir. "D","A" ve "H" karakterleri kendilerinden önce gelen harflere göre üst seviyede temsil ettikleri harflere dönüşürler.

- A: a yada e harfi yerine
- D: d yada t yerine
- C: c yada ç yerine
- H: ı,i,u,ü harfleri yerine
- I: ı yada i harfleri yerine
- :0 boş geçiş

Üst seviyeden girdi olarak verilen bir sözcükteki ses olaylarının çözümlenip sözlüğün anlayacağı sözcük haline çevrilmesi, iki seviyeli analizdir. Türkçe için iki seviyeli analiz ilk defa Oflazer [5], tarafından yapılmıştır. Bu alanda çalışma yapan diğer araştırmacılar tarafından bu sistem hem geliştirilmiş hem de üstünde birçok yorumlar yapılmıştır ve hala Türkçe biçimbilimsel çözümleme için en iyi olarak kabul görmektedir.

Tez kapsamında Oflazer [5]'in, çıktılarında yararlanarak hem sonlu durumlu makinelerin son hali hazırlandı hem de yapım ekleri üzerinde üretim sınırını azaltacak geliştirmeler yapıldı. Bunun yanısıra özellikle bir sözcük için üretilen fazla analizler bayrak işaretleri kullanarak azaltıldı, yapım eki alan kelimelerin yapım eki almış hali yine sözlükte varsa o yapım ekini alması engellendi. Böylece alt seviyede biçimbilimsel çözümleyiciyi kullanan biçimbilimsel belirsizlik giderici gibi sistemlerin başarımının artması sağlandı. Tasarlanan biçimbilimsel çözümleyici hem üst seviyedeki araçların başarımını artırmış olup hem de üretilen analizlerde kapsam testlerinde en iyi sonuçları almıştır. Son bölümde bu konuya değinilmiştir.

**Çizelge 3.1: Sözlükteki Etiketler**

Ekin etiketi	İngilizcesi	Açıklaması	Ekin kendisi
Ness	Necessitative	Gereklilik eki	+lHk
Agt	Agt	İlgi eki	+cH
Dim	Dimentional	Küçümseme eki	+cHk
A1sg	A1sg	1. tekil şahıs eki	+Hm
A1pl	A1pl	3. çoğul şahıs eki	+Hz
A2sg	A2sg	2. tekil şahıs eki	+sHn
A2pl	A2pl	2. çoğul şahıs eki	+sHnHz
A3pl	A3pl	3. çoğul şahıs eki	+lAr
A3sg	A3sg	3. tekil şahıs eki	0
Become	Become	Olma eki	+lAş
Acquire	Acquire	Aitlik eki	+lAn
P3pl	Personal	3. çoğul şahıs iyelik eki	+lArH
P1sg	Personal	1. tekil şahıs iyelik eki	+Hm
P2sg	Personal	2. tekil şahıs iyelik eki	+Hn
P1pl	Personal	1. çoğul şahıs iyelik eki	+HmHz
P2pl	Personal	2. çoğul şahıs iyelik eki	+HmHz
P3sg	Personal	3. tekil şahıs iyelik eki	+sH
With	With	Berberlik eki	+lH
Without	Without	Berber olmama eki	+sHz
Dat	Dative	İsmin yönelme (e) hali	+yA
Loc	Locative	İsmin bulunma (de) hali	+DA
Abl	Ablative	İsmin ayrılma (den) hali	+DAn
Gen	Genitive	İsmin ilgi (in) hali	+nHn
Ins	Ins	Birliktelik eki	+(y)lA
Acc	Accusative	İsmin yüklenme (i) hali	+(y)H
Rel	Relation	İlgi eki	+ki
While	While	Zaman zarfı	+yken
Cond	Condition	Koşul eki	+ysA
Adverb+AsIf	Adverb	Fiilden zarf yapma eki	+cAsHnA
Narr	Narrative	Rivayet geçmiş zaman eki	+ymHş
Reflex	Reflexive	Dönüşlülük eki	+Hn
Recip	Reciprocal	İşteşlik eki	+Hş
Caus	Causative	Ettirgen fiil eki	+dHr
Pass	Passive	Pasiflik eki	+Hl
Fut	Future	Gelecek zaman eki	+yAcAk
Prog2	Progressive	Devamlılık eki	+mAktA
Neces	Necessitative	Gereklilik eki	+mAIlH
Prog1	Progressive	Şimdiki zaman eki	+Hyor
Adj+Aor	Adjective	Fiilden sıfat yapma eki	+Ar
Aor	Aorist	Geniş zaman eki	+Hr
Able	Able	Yeterlilik eki	+yAbHl
Hastily	Hastily	Tezlik eki	+yHver
Repeat	Repeat	Süreklilik eki	+yAdHr
Neg	Negative	Olumsuzluk eki	+mA
Caus	Causative	Ettirgenlik eki	+t
Inf1	Infinitive	Fiilden isim yapma eki	+mA
Inf2	Infinitive	Fiilden isim yapma eki	+mAk
Inf3	Infinitive	Fiilden isim yapma eki	+Hş
Past	Past	Geçmiş zaman eki	+yDH
Cop	Cop	Kesinlik eki	+DHr
Narr	Narrative	Rivayet geçmiş zaman eki	+mHş
Cond	Condition	Koşul eki	+ysA
Pos-Aor	Positive aorist	Olumlu geniş zaman eki	+Hr
Pos-Fut	Positive future	Olumlu gelecek zaman eki	+yAcAk
Pos-Narr	Positive narrative	Olumlu rivayet zaman eki	+ymHş
Pos-Opt	Positive optative	Olumlu dilek şart eki	+yA
Pos-Desr	Positive desire	Olumlu istek eki	+sA

Çizelge 3.2: Sözlükteki Etiketler

Ekin etiketi	İngilizcesi	Açıklaması	Ekin kendisi
AsLongAs	As long as	Uzun süren zaman eki	+dHkçA
SinceDoingSo	Since doing so	Süre gelen zamanı bildirir ek	+yAIH
AfterDoingSo	After doing so	Sonraki zamanı bildirir ek	+yHp
ByDoingSo	By Doing So	devam eden zaman eki	+yArAk
When	When	Olduğu zamanı gösteren ek	+yHncA
Feellike	Feel like	Gibi hissetme eki	+yAsH
PastPart	Past participle	Sıfat Fiil eki	+dHk
FutPart	Future participle	Sıfat Fiil eki	+yAcAk
PresPart	Present participle	Sıfat Fiil eki	+yAn
Adj+NarrPart	Narrative participle	Sıfat Fiil eki	+ymHş
NotState	Not State	Olumsuzluk bildiren fiilden isim yapma eki	+mAzLHk
WithoutHavingDoneSo	Without having done so	Yapmaksızın eki	+mAksHzHn
WithoutHavingDoneSo	Without having done so	Yapmadan eki	+mAdAn
Adj+Agt	Adjective Agt	İlgi eki	+yHcH
Adamantly	Adamantly	Kıyasıya eki	+yAsHyA
Aor-A1sg	Aorist A1sg	Geniş zaman 1. tekil şahıs eki	+m
Aor-A2sg	Aorist A2sg	Geniş zaman 2. tekil şahıs eki	+zsHn
Aor-A3sg	Aorist A3sg	Geniş zaman 3. tekil şahıs eki	+z
Aor-A1pl	Aorist A1pl	Geniş zaman 1. çoğul şahıs eki	+yHz
Aor-A2pl	Aorist A2pl	Geniş zaman 2. çoğul şahıs eki	+zsHnHz
Aor-A3pl	Aorist A3pl	Geniş zaman 3. çoğul şahıs eki	+zAr
Start	Start	Başlama eki	+yAkoy
Stay	Stay	Kalma eki	+yAkAl
Equ	Equal	Eşitlik eki	+cA
Narr+A3sg	Narrative A3 singular	Rivayet 3. tekil şahıs eki	+mHş
Past+A2sg	Past A2 singular	Geçmiş zaman 2. tekil şahıs eki	+yDHn
Past+A1sg	Past A1 singular	Geçmiş zaman 1. tekil şahıs eki	+yDHm
Past+A3sg	Past A3 singular	Geçmiş zaman 3. tekil şahıs eki	+yDH
Zero-Cond	Condition	Koşul eki	+sA
Zero-Pres-A2pl	Present A2 plural	Şimdiki zaman 2. çoğul şahıs eki	+sHnHz
Zero-Pres-A1pl	Present A1 plural	Şimdiki zaman 1. çoğul şahıs eki	+yHz
Zero-Pres-A2sg	Present A2 singular	Şimdiki zaman 2. tekil şahıs eki	+sHn
Zero-Pres-A1sg	Present A1 singular	Şimdiki zaman 1. tekil şahıs eki	+yHm
+Pres+Zero+Cop	PresZeroCop	Şimdiki zamanda kesinlik eki	+DHR
Adj+Asif	As if	Gibi eki	+cA
Repeat	Repeat	Süreklilik eki	+yAgör
EverSince	Ever since	Süreklilik eki	+yAgel
Almost	Almost	Yaklaşma eki	+yAyAz
DB	Derivational Boundary	Üretim Sınırı	Yapım ekleri için kullanılır

Çizelge 3.3: Fazla Analiz (Overgeneration) ve Yanlış Analiz Örneği

<b>OFLMORPH [5]</b>	<b>kocaman koca +Verb+Pos+DB+Noun+Inf2+A3sg+P2sg+Nom kocaman kocaman +Adj</b>
<b>ITUMORPH</b>	<b>kocaman kocaman +Adj</b>
<b>OFLMORPH [5]</b>	<b>toplumun top +Noun+A3sg+Pnon+Nom+DB+Adj+With +DB+Noun+Zero+A3sg+P1sg+Gen toplumun toplu +Adj+DB+Noun+Zero+A3sg+P1sg+Gen toplumun toplum +Noun+A3sg+P2sg+Nom toplumun toplum +Noun+A3sg+Pnon+Gen</b>
<b>ITUMORPH</b>	<b>toplumun toplu +Adj+DB+Noun+Zero+A3sg+P1sg+Gen toplumun toplum +Noun+A3sg+P2sg+Nom toplumun toplum +Noun+A3sg+Pnon+Gen</b>

### 3.1 Twolc Kuralları

Türkçe'de karşılaştığımız ses olaylarını iki seviyeli analiz yapabilmek için tez çalışması boyunca Xfst aracını kullanarak Türkçenin dilbilgisi kuralları tanımlandı. Bu kurallar ile hazırlanan sözlüğü, birleştirme yöntemi (compose-intersect) ile yine Xfst kullanarak birleştirip sonlu durumlu makine hazırlandı. Bu kısımda Türkçe'deki ses olaylarını inceleyen kuralların hazırlanışından bahsedilmiştir. Öncelikle kullanılan harfler Türkçe'deki ses olaylarına uygun şekilde gruplandı ve grupların isimleri hazırlanan kurallar için kullanıldı. Hazırlanan kurallarda ':' nın sağ tarafı üst seviye (surface level) yani sözcüğün sisteme girdi olarak verilen hali sol taraf ise alt seviye (lexical level) kısmıdır.

- Alt seviye: bey\$in+sH
- Üst seviye: bey00n00i

Beyin, sözcüğü sesliyle başlayan ek aldığındaki "i" harfi düşer. Bu şekilde olan sözcükler sözlükte \$ karakteri içerir ve \$ karakterinden sonraki ünlü harf, sözcük sesliyle başlayan ek aldığındaki düşer. Alt seviyede bey\$in+sH şeklindeki bir ifade üst seviyede beyni haline gelir. Şekil 3.1 de alt seviyedeki bir sözcüğün üst seviyede başka bir karaktere nasıl dönüştüğü incelenebilir.

Kurallar hazırlanırken sadece grup ismi yazılarak o gruptaki bütün harfleri ve karakterleri temsil ettiğinden alt seviyede sözlük hazırlanırken harfler gruplara ayrılmıştır. Twolc kuralları hazırlarken kullandığımız yer kısıtlama (context restriction) özelliği dört adettir şekil 3.1'de bununla ilgili örnekler verilmiştir. İlk satırdaki kural tipi, soldaki dönüşümün her zaman ve sadece işaretin sağ tarafındaki bağlamda gerçekleşeceğini belirtir. İkinci satırdaki kural tipi dönüşümün her zaman, üçüncü satırdaki kural tipi dönüşümün sadece ilgili bağlamda gerçekleşeceğini belirtir. Dördüncü satırdaki kural tipi dönüşümün asla ilgili bağlamda gerçekleşmeyeceğini belirtir.

- CONS = b c ç d f g ğ h j k l m n p r s ş t v y z D; (Ünsüz Harfler)
- VOWEL = a e ı i o ö u ü H A ; (Ünlü Harfler)

a:b $\Leftrightarrow$ l_r	lar ✓ lbr	lar ✗ lar	lbr ✓ lbr	xay ✗ xby
a:b $\leq$ l_r	lar ✓ lbr	lar ✗ lar	lbr ✓ lbr	xay ✓ xby ✓
a:b $\geq$ l_r	lar ✓ lbr	lar ✓ lar	lbr ✓ lbr	xay ✗ xby
a:b/ $\leq$ l_r	lar ✗ lbr	lar ✓ lar	lbr ✓ lbr	xay ✓ xby ✓

Şekil 3.1: Twolc Kural Örnekleri [2]

- BACKV = a ı u o ; (Kalın Ünlüler)
- FRONTV = e i ö ü; (İnce Ünlüler)
- FRUNRV = i e; (İnce Düz Ünlüler)
- FRROV = ö ü; (İnce Yuvarlak Ünlüler)
- BKROV = u o; (Kalın Yuvarlak Ünlüler)
- BKUNRV = a ı ; (Kalın Düz Ünlüler)
- X = s y n; (Kaynaştırma Ünsüzleri)

Aşağıda, kullanılan iki seviyeli kurallar sırasıyla anlatılmaktadır. Kurallar hazırlanırken kullanılan ifadeler birinci kural üzerinden açıklanacak olursa, kuraldaki [BACKV:l:BACKV] [:CONS]\* kısmı sözcüğün kendisinde mutlaka kalın ünlü (BACKV) ve en az bir tane ünsüz harf (CONS) bulunması gerektiğini gösterir. (+:0) kısmı sözcüğe eklenen eki temsil eder. Sözlükte eklerin başında + karakteri bulunur. (+:0) ifadesi alt seviyede eklerin başında bulunan '+' karakterinin '0' a

dönüştüğünü gösterir. [CONS:!:CONS | :0]\* kısmı ise sözcüğe eklenen ekin içeriğinde en az bir adet sessiz harf (CONS) bulunması gerektiğini gösterir. En sondaki \_ kısmı ise alt seviyedeki "A" karakterinin geleceği yeri gösterir. ":" karakterinin sağ tarafı üst seviyeyi (surface level) sol tarafı alt seviyeyi (Lexical level) temsil eder. Tez kapsamında Oflazer [5]'in çalışmasındaki yöntemlerden yararlanılarak Türkçe dilbilgisi için Twolc kuralları yazılmıştır.

- **Kural 1 - 'A' nın 'a' ya dönüşümü;**

- A:a => [BACKV:!:BACKV] [:CONS]\* (%+:0) [%']\*[CONS:!:CONS | :0]\*\_ ;

Birinci kural büyük ünlü uyumu ile ilgili olan bir kuraldır. Sözcüğün kökünde kalın ünlüler (BACKV) varsa alt seviyede (lexical level) sözcüğün ekinde "A" karakterini içeren morfeplerdeki "A" karakteri üst seviyede "a" karakterine dönüştürülmektedir. araba+lAr -> arabalar masa+lAr -> masalar Alt seviyede sözcüğün aldığı ektaki "A" karakteri kendinden önce gelen ünlüler kalın olduğundan üst seviyede "a" karakterine dönüşmüştür.

- **Kural 2 - 'A' nın 'e' ye dönüşümü;**

- A:e => [^ :o!%#:al:FRONTVIFRONTV:] [:CONS]\*(%+:0)[']\*[CONS:!:CONS | :0]\*\_ ;

İkinci kural da yine büyük ünlü uyumu ile ilgili olan bir kuraldır. Sözcüğün kökünde ince ünlüler varsa bu sözcüğe eklenen morfeplerdeki "A" karakteri "e" karakterine dönüşür. Bununla birlikte "hal" ve "rol" gibi içinde şapkalı kalın ünlü olan sözcükler, içinde "A" karakteri olan bir ek aldıklarında "A" karakteri "e" karakterine dönüşür. Bu şekilde harfler sözlükte # ve ^ karakterleriyle temsil edilmiştir. Örnek olarak,

- rol+lAr ->roller
- hal+lAr+sH -> halleri
- kedi+lAr -> kediler
- ince+lAş -> inceleş

gibi durumlar verilebilir.

- **Kural 3 - 'H' nin 'i'ye dönüşümü;**

- H:i => [:FRUNRV|FRUNRV:] [CONS]\* (%+:0) [%']\* [CONS:|:CONS | :0]\* \_ ;  
 [%#:a][CONS](%+:0) [CONS:| :CONS | :0]\* \_ ;

Üçüncü kuralda sözcük eğer "i" yada "e" karakteri içeriyorsa ve köke eklenen morfemde "H" karakteri varsa bu "H" karakteri "i" ye dönüşür istisnai durum olarak "seyahat" sözcüğünde olduğu gibi eğer sözcük şapkalı "a" harfi içeriyorsa bu sözcüğe eklenen "H" karakteri yine "i" ye dönüşür.

- gel+Hr -> gelir
- seyahat+sH -> seyahati

- **Kural 4 - 'H'nin 'ı'ya dönüşümü;**

- H:ı => [:BKUNRV] [CONS]\* (%+:0) [%']\* [CONS:|:CONS | :0]\* \_ ;

- **Kural 05 - 'H nin 'u'ya dönüşümü;**

- H:u => [BKROV:|:BKROV] [CONS]\* (%+:0) [%']\* [CONS:|:CONS | :0]\* \_

Dördüncü ve beşinci kurallarda ise sözcüğün içindeki sesliler "u" ve "o" ise ekin içindeki "H" karakteri "u" ya dönüşür. Eğer sözcüğün içindeki sesli harfler "a" ve "ı" ise o zaman ekteki "H" karakteri "ı" ya dönüşür. Bazı istisnai durumlar da mevcuttur mesela sözcüğün içindeki sesli harf şapkalı karakter içeriyorsa o zaman morfemdeki "H" karakteri "ü" ye dönüşür.

- araba+sH -> arabası
- masa+yH -> masayı
- kol+sH -> kolu
- petrol+sH -> petrolü

- **Kural 6 - 'H'nin üst seviyede silinmesi;**

- H:0 => [0:VOWEL] %+:0 \_



Altıncı kuralda eğer bir sözcük sesli harfle bitiyorsa ve "H" karakteri ile başlayan bir ek alırsa ekteki "H" karakteri düşer. Örnek olarak

- masa+Hm -> masam
- araba+HmHz -> arabamız
- **Kural 7 - '+Hyor' ekinden önce 'A'nın düşmesi;**
- A:0 <=>\_ %+:0 [H:] y o r;

Yedinci kuralda ise "+Hyor" morfeminden önceki "A" karakteri ile biten bir morfem varsa bu morfemdeki "A" karakteri düşüp "H" karakterinin ise büyük ünlü uyumuna göre gerekli sesli harfe dönüşümü sağlanır.

- sepet+lA+Hyor -> sepetl0iyor -> sepetliyor
- konuş+mA+Hyor -> konuşm0uyor -> konuşmuyor
- **Kural 8 - Kaynaştırma harflerinin silinmesi;**
- Cx:Cy <=> [:CONS|CONS:] %+:0 \_ [:CONS !:VOWEL]; where Cx in (y s n) Cy in (0 0 0) matched;

Sekizinci kuralda sessiz harfle biten bir sözcüğe "y", "s", "n" ile başlayan bir morfem eklendiğinde morfemdeki "y", "s", "n" harfi silinir.

- telefon+nHn -> telefonun
- kalem+sH -> kalemi
- monitör+yH -> monitörü
- **Kural 9 - Ünlü düşmesi;**
- Cx:Cy <=> %\$:0 \_ [ [ CONS %+:0 (X:0) [A:%H:] ] | [ %+:0 [%H:] y o r ] ];where Cx in (ı i o ö u ü) Cy in (0 0 0 0 0 0) matched;

Türkçe’de bazı sözcükler sesliyle başlayan bir ek aldığında sözcüğün son sesli harfi düşer, düşecek olan harfi belirlemek için o harften önce özel bir karakter olan "\$" karakteri sözlükte bu sözcüğe eklendi. Dokuzuncu kuralda bu durm temsil edilmiştir.

- ak\$ıl+Hm -> aklım

- bur\$un+Hm -> burnum

- **Kural 10 - D'nin t'ye dönüşümü;**

- %D:t <=> [h | ç | ş | k | p | t | f | s | H|Ç|Ş|K|T|F|S] %+:0 (:0) \_ [%H:| A:];

Onuncu kural, Türkçe dilbilgisi kurallarından biri olan ünsüz sertleşmesi kuralıdır. Yani "f", "s", "t", "k", "ç", "ş", "h", "p" harflerinden sonra "d" ile başlayan bir ek gelirse bu "d" harfi "t" harfine dönüşür.

- sahaf+DA -> sahafta

- Salih+Dan -> salihten

- fıstık+DA -> fıstıkta

- **Kural 11 - 'c' nin 'ç' ye çevrilmesi;**

- c:ç <=> [ :ç | ş | k | p | t | f | s ] +:0 \_ [H:| A:];

Onbirinci kural, Türkçe dilbilgisi kurallarından biri olan ünsüz sertleşmesi kuralıdır. Yani "f", "s", "t", "k", "ç", "ş", "h", "p" harflerinden sonra "c" ile başlayan bir ek gelirse bu "c" harfi "ç" harfine dönüşür.

- haraç+cH -> haraççı

- ulus+cA -> ulusça

- **Kural 12 - 'b' ,d' ile biten sözcüklerin 'p',t' ye dönüşümü;**

- Cx:Cy => \_%+:0(X:0)[CONS]\*; where Cx in (b d) Cy in (p t) matched;

Onikinci kuralda, Türkçe'de sonu "p" yada "t" ile biten sözcükler sonunda "b" yada "d" olacak şekilde sözlükte tutuldu. Eğer sözcük herhangi bir ek almazsa ya da sessizle başlayan bir ek alırsa sondaki "b", "d" harfleri "p", "t" ye dönüşür.

- kitab+lAr -> kitaplar
- dolab+nHn -> dolabın
- tad+sH -> tadı
- **Kural 13 - Sözcük sonunda 'b' ya da 'd' olan sözcükleri 'p' ya da 't' ye çeviren kural;**
- Cx:Cy =>\_ .#.;where Cx in (b d) Cy in (p t) matched;

Türkçe'de sonu "p" ve "t" ile biten sözcükler sesli ile başlayan ek aldıklarında "p", "b" ye "t", "d" ye dönüşür. Sözlükte bu kuralı uygulayabilmek için sonu "p" ya da "t" ile biten sözcükler b ya da d ile bitecek şekilde tutuldu. Onüçüncü kural ise sözlükteki sonu "b" ya da "d" ile biten sözcükleri ek almadıkları durumlarda "p" ya da "t" ye dönüştüğünü göstermek için hazırlanmıştır.

- kitab -> kitap
- dolab -> dolap
- tad -> tat
- **Kural 14 - 'ç'nin 'c'ye dönüşmesi;**
- ç:c <=> \_ %+:0 (X:0) VOWEL;

Ondördüncü kuralda sonu "ç" ile biten bir sözcüğün sesli bir harfi ek olarak aldığıında sondaki "ç" harfinin "c" ye dönüştüğü gösterilmektedir.

- kazanç+sH ->kazancı
- genç+nHn -> gencin
- ihtiyaç+sH-> ihtiyacı

- **Kural 15 - 'k' nın 'g' ye dönüşümü;**

- k:ğ <=> \_ %+:0 [:0|VOWEL];

Türkçe'de sonu "k" ile biten bir sözcük eğer sesliyle başlayan bir ek alırsa bu sözcüğün sonundaki "k" harfi yumuşar ve "ğ" harfine dönüşür. Buna yumuşama denir. Onbeşinci kuralda Türkçe'deki bu dilbilgisi kuralı gerçekleştirilmiştir.

- tarak+Hm -> tarağım
- ayak+sH -> ayağı
- inek+sH -> ineği

- **Kural 16 - k nın g ye dönüşmesi;**

- k:g => n \_ %+:0 (X:0) VOWEL;

Onaltıncı kuralda tanımladığımız duruma bazı istisna olan sözcükler uymaz. Mesela sonu "k" karakteri ile bitmesine rağmen "k" karakterinden önce "n" harfi bulunan sözcüklerde sesli ile başlayan bir ek geldiğinde bu "k" harfi "ğ" ye değil "g" ye dönüşür. Bu kuralda bu tür istisna olan durumlar gösterildi.

- renk+sH ->rengi
- ahenk+sH-> ahengi
- denk+sH -> dengi

- **Kural 17 - g nin ğ ye dönüşmesi;**

- g:ğ => \_ %+:0 (X:0) VOWEL;

- **Kural 18 - g nin ğ ye dönüşmediği istisna durumlar;**

- g:ğ /<= [ n | r ] \_ ;

Türkçe'de sonu "g" ile biten ve Türkçe'ye başka dillerden geçen bazı sözcükler sesli ile başlayan bir ek aldığı anda sondaki "g" harfi "ğ" harfine dönüşür. Onyedinci kuralda bu durum tanımlanmıştır. Örnek olarak radyolog+yA -> radyoloğa durumları verilebilir.

Ancak bazı durumlar bu kuralı bozar. Mesela son harfi "g" olmasına rağmen sesli harfle başlayan ek aldığında "ğ" olmayan sözcükler vardır. Onsekizinci kuralda bu durum tanımlanmıştır.

- aysberg+sH+nHz -> aysberginiz
- birifing+Hm->birifingim

### 3.2 Sözlükteki Ekler ve Sözcük Kökleri

Çizelge 3.4: Sözcük Sınıfları

Sözlükteki Adı	Açıklaması	Analizdeki Çıktısı
<b>Noun</b>	<b>isim</b>	<b>+Noun</b>
<b>Verb</b>	<b>fil</b>	<b>+Verb</b>
<b>Proper Noun</b>	<b>özel isim</b>	<b>+Noun+Prop</b>
<b>Adjective</b>	<b>sıfat</b>	<b>+Adj</b>
<b>Techninal</b>	<b>Teknik sözcükler</b>	<b>+Noun</b>
<b>Duplication</b>	<b>ikilemeler</b>	<b>+Dup</b>
<b>Postposition</b>	<b>Edatlar</b>	<b>+Postp+PCNom</b>
<b>Question</b>	<b>Soru</b>	<b>+Ques</b>
<b>Determiner</b>	<b>zamirler</b>	<b>+Det</b>
<b>Number</b>	<b>sayılar</b>	<b>+Num+Card</b>
<b>Connectives</b>	<b>bağlaçlar</b>	<b>+Conj</b>
<b>Pronoun</b>	<b>zamirler</b>	<b>+Pnon</b>

Bu çalışma kapsamında sözcük kökü olarak oniki adet sözcük grubu kullanıldı. Çizelge 3.4'de listelenen sözcük sınıfları Noun, Verb, Pronoun, Adjective, Technical, Duplication, Postposition, Question, Determiner, Number, Pronoun ve Connectives gibi isimlerle adlandırılmıştır. Hazırlanan sözlükte oniki adet sözcük grubundaki sözcüklerle birlikte, bu sözcüklerin alacağı ekler bulunmaktadır.

#### • LEXICON NOUNS

- araba NOUN-POS;
- masa NOUN-POS;
- gözlük NOUN-POS;
- sandalye NOUN-POS;

- bilgisayar NOUN-POS;
- **LEXICON NOUN-POS**
- +Noun:0 NOUN-PLURAL;
- **LEXICON NOUN-PLURAL**
- +Pnon:0 NOUN-POSSESSIVE;
- +P1sg:+Hm NOUN-POSSESSIVE;
- +P2sg:+Hn NOUN-POSSESSIVE;
- +P1pl:+HmHz NOUN-POSSESSIVE;
- +P2pl:+HnHz NOUN-POSSESSIVE;
- +P3sg:+sH NOUN-POSSESSIVE-3;

Yukarıda sözlük örneği isim soylu sözcüklerin bulunduğu sözlükten alınan bir parçadır. Burada "LEXICON NOUNS" kısmı isimlerin bulunduğu sözlüğü temsil eder. Burada verilen beş adet sözcüğün yanında yaklaşık 35000 adet isim soylu sözcük daha sözlükte tutulmaktadır. "araba NOUN-POS" kısmı araba sözcüğünün alacağı eklerin "NOUN-POS" kısmında olduğunu gösterir. "LEXICON NOUN-POS" kısmı ise isim sözlüğündeki sözcüklerin ilk olarak hiç bir ek almadan yani sonlu durumlu makinede ":0" geçişiyle +Noun etiketinin atanacağını gösterir. Sözcük bu kısımda hiçbir ek almasa dahi analizinde +Noun şeklinde bir etiket bulunur bu etiket sözcüğün isim sözlüğünde bulunduğunu temsil eder. Bu kısımdan sonra ":0" geçişiyle "NOUN-PLURAL" kısmına gidilir. Burada sözcük aitlik ekleri olan "Hm", "Hn", "HmHz", "HnHz", "sH", gibi ekleri alabilir, ya da ":0" geçişiyle hiçbir ek almadan sonlu durumlu makinede bir sonraki duruma geçiş yapar. Bu örnekte anlaşıldığı üzere sözlükte hem sözcük kökleri hem de o sözcüklerin aldığı ekler bulunmaktadır.

Bu çalışmadaki sözlüğün temelini Türk Dil Kurumunun sözlüğü oluşturmaktadır. Çalışmamızın ilk aşamasında [20], Odtü Sabancı ağaç yapılı derlemi [21] üzerinde Oflazer [5]'in çözümleyicisi ile birebir aynı çözümleri üretecek şekilde sözlük ve ek listesi oluşturulmuştur. Bu sonuçlar incelendiğinde birtakım sözcüklerin

analizlerinde hem fazla analizlerin olduğu hem de bazı analizlerin içinde çok fazla üretim sınırı olduğu gözlemlendi. Bu durum sözlüğü geliştirme ihtiyacı olduğunu göstermiştir.

### 3.2.1 Sözlük Bölütleme, İyileştirme ve Genişletme

İsim sözlüğü temelde üç ayrı gruba ayrılmıştır. Normalde bayrak işaretleri kullanarak bu durumu çözebilseydik sözlük bölütlemeye gerek olmayacaktı. Ancak bayrak işaretleri kullanarak çözülemediğinden sözlük bölütleme yöntemi kullanıldı. Sözlük bölütleme yapmadan çözülememesinin sebebi birtakım sözcüklerin hiçbir ek almadığı durumda analizin hem yalın hem de yön yada çoğulluk bakımından sanki birtakım ekleri almış gibi analiz üretilmesini gerektirmesidir. İsim sözlüğü sözcüklerin temel halde iken yalın, çoğul ya da bulunma halinde olmasına göre gruplandırılmıştır. Örnek verecek olursak "masa" sözcüğü temelde yalın bir anlam taşır ama "içeri", "yukarı", "aşağı", "dışarı" gibi kelimeler yalın anlamı ile birlikte yön belirtirken "ahali", "dışışleri" gibi temelde sözlükte kök halinde buldukları halde analizinde çoğul anlamını taşıdıklarından bu şekilde bir yöntem kullanıldı. Çizelge 3.5'te örnekler incelenebilir. Bilinmeyen sözcük analizcisinin ürettiği sözcüklere etiket olarak +Guess atandı. Böylece eğer bir sözcüğün analizinde +Guess varsa bu analiz bilinmeyen sözcük analizcisinin yaptığı analiz olduğu anlaşılır. Sözlüğü (lexiconu) genişletmek amacıyla TDK veritabanındaki bütün isim olarak geçen kelimeleri sözlüğümüzde yoksa bunlar sözlüğe eklendi. Sözlüğü genişletirken yapım eki alarak başka bir sözcük haline dönüşen kelimeler varsa bu kelimedeki bizim sözlüğümüzde bulunuyorsa herhangi bir girdinin analizinde fazla analize (overgeneration) sebep olacak olan türetim sınırı (+DB derivational Boundary) etiketlerini engelleyecek bir bayrak işareti sistemi tasarlandı. Bu şekilde fazla üretim yapılması engellendi. Sözlüğü genişletirken kullandığımız diğer bir yöntem ise analiz için kullandığımız verilerin morfolojik analiz sonucu "+Guess" ile başlayan etiketlere sahip sonuçları varsa ve bu kelimeler normalde anlamlı kelimeler ise bu sözcükler sözlüğe eklenerek sözlük genişletildi.

Çizelge 3.5'te de görüldüğü gibi "ahali", "dışışleri" gibi sözcükler temel halde sözlüğümüzde bulunmaktadır. Ancak analizine bakıldığında herhangi bir şekilde çoğul eki almamsına karşın çoğul anlamı taşıdığından dolayı çoğul eki olan

**Çizelge 3.5: Sözlük Bölütleme Örneği**

<b>ahali ahali +Noun+A3pl+Pnon+Nom</b>
<b>dışışleri dışışleri+Noun+A3pl+Pnon+Nom</b>
<b>aşağı aşağı+Noun+A3sg+Pnon+Dat</b>
<b>yukarı yukarı+Noun+A3sg+Pnon+Dat</b>
<b>yukarı yukarı+Adj</b>
<b>masa masa +Noun+A3sg+Pnon+Nom</b>

"A3pl" etiketiyle analiz üretilmiştir. Bunun sebebi isim sözlüğünde bölütleme yapmamızdır. Aynı durum "yukarı", "aşağı" ya da "dışarı" gibi sözcükler için de geçerlidir. Çünkü temel halde herhangi bir ek almadıkları halde bir yön anlamı taşımaktadırlar. Bu yüzden analizlerinde temel halde "+Dat" etiketi üretilmiştir. "Masa" sözcüğü ise yalın halde bulunup temel anlamı herhangi bir şekilde başka anlam içermez. Analizine bakıldığında yalın halde tekil bir sözcüktür.

**Çizelge 3.6: Boyut İndirgeme Yapılan Ekler**

<b>Ekin etiketi</b>	<b>Açıklaması</b>	<b>Ekin kendisi</b>
<b>Dim</b>	<b>Küçümseme eki</b>	<b>+cHk</b>
<b>Ness</b>	<b>Gereklilik eki</b>	<b>+lHk</b>
<b>With</b>	<b>Beraberlik eki</b>	<b>+lH</b>
<b>Agt</b>	<b>Meslek eki</b>	<b>+cH</b>
<b>Pass</b>	<b>Edilgen yapma eki</b>	<b>+Hl</b>
<b>Caus</b>	<b>Ettirgenlik eki</b>	<b>+Hr</b>
<b>Caus</b>	<b>Ettirgenlik eki</b>	<b>+dHr</b>
<b>Caus</b>	<b>Ettirgenlik eki</b>	<b>+Ht</b>
<b>Inf1</b>	<b>Fülden isim yapma eki</b>	<b>+mA</b>
<b>Inf2</b>	<b>Fülden isim yapma eki</b>	<b>+mAk</b>
<b>+Adj+AorPart</b>	<b>Fülden isim yapma eki</b>	<b>+mHş</b>
<b>+Adj+AorPart</b>	<b>Fülden isim yapma eki</b>	<b>+mAz</b>
<b>Pass</b>	<b>Edilgen yapma eki</b>	<b>+Hn</b>
<b>Inf3</b>	<b>Fülden isim yapma eki</b>	<b>+Hş</b>
<b>Become</b>	<b>Olma eki</b>	<b>+lAş</b>
<b>Acquire</b>	<b>Aitlik eki</b>	<b>+lAn</b>
<b>Without</b>	<b>Beraber olmama eki</b>	<b>+sHz</b>

### **3.3 Boyut İndirgeme**

Yapım ekleri için yapılan çalışmada analiz sırasında minimum derecede "DB" (Derivational Boundary) olan analiz çıkması için çalışılmış olup eğer sözcüğün yapım ekini aldığı durum sözlükte varsa o sözcüğün sözkonusu olan yapım



ekini alması bayrak işaretleri kullanılarak engellenmiştir. Bu sayede bir sözcük için minimum sayıda "DB" içeren analizlerin üretilmesi sağlanmıştır. Boyut indirgeme çalışması yaptığımız ve analizinde üretim sınırı içeren ekler çizelge 3.6'da verilmiştir.

İsimden-isim yapan yapım ekleri olan "+cHk", "+lHk", "+lH", "+cH" gibi ekler (morfemler) için çizelge 3.7'de örnekler verilmiştir. Burada görüldüğü gibi "arpacık" aslında dilimize oturmuş bir kelime olduğundan bu sözcüğün "arpa" sözcüğünden "cHk" ekini alarak fazla analiz üretilip analiz sonuçlarında gösterilmesine gerek görülmediğinden analizlerimizde olabildiğince az sayıda üretim sınırı (derivational boundary) çıkartılması sağlanmıştır. Kitapçık örneğinde de benzer durum söz konusudur. Sözlükte hem kitap hem de kitapçık varsa kitap sözcüğünün "çık" eki alarak kitapçık olması fazla analiz örneğidir.

**Çizelge 3.7: +cHK Eki Örnekleri**

<b>OFLMORPH [5]</b>	arpacık arpa +Noun+A3sg+Pnon+Nom+DB+Noun+Dim+A3sg+Pnon+Nom arpacık arpacık+Noun+A3sg+Pnon+Nom kitapçık kitap+Noun+A3sg+Pnon+Nom+DB+Noun+Dim+A3sg+Pnon+Nom kitapçık kitapçık+Noun+A3sg+Pnon+Nom
<b>ITUMORPH</b>	arpacık arpacık+Noun+A3sg+Pnon+Nom kitapçık kitapçık+Noun+A3sg+Pnon+Nom

"+lHk" eki için ise burada da görüleceği gibi "ter" kelimesinden "terlik" türemiştir. Ancak "ter" sözcüğünden "terlik" sözcüğünün türetilmesi uygun değildir. Çizelge 3.8'de örneği verilmiştir.

**Çizelge 3.8: +lHk Eki Örnekleri**

<b>OFLMORPH [5]</b>	terlik ter+Noun+A3sg+Pnon+Nom+DB+Noun+Ness+A3sg+Pnon+Nom terlik terlik +Noun+A3sg+Pnon+Nom kitaplık kitaplık+Noun+A3sg+Pnon+Nom
<b>ITUMORPH</b>	terlik terlik +Noun+A3sg+Pnon+Nom kitaplık kitaplık+Noun+A3sg+Pnon+Nom

"+lH" eki için kendi sözlüğümüzde "kıdemli" sözcüğü bulunduğu ve dilimize yerleştiğinden dolayı "kıdem" kelimesine "lH" eki eklenerek "kıdemli" yapmak yerine direk "kıdemli" sözcüğünden analiz üretildi. Çizelge 3.9'da örneği verilmiştir.

"+cH" eki için de dilimize "cH" ekini alarak yerleşen bazı örnekler verebiliriz. Bunlardan birisi "boyacı" ancak Oflazer [5]'in sözlüğünde de "boyacı" sözcüğü

**Çizelge 3.9: +lH Eki Örneği**

<b>OFLMORPH [5]</b>	<b>kıdemli kıdem+Noun+A3sg+Pnon+Nom+DB+Adj+With</b>
<b>ITUMORPH</b>	<b>kıdemli kıdemli+Noun+A3sg+Pnon+Nom</b>

olduğundan "boya" sözcüğünden türetmek yerine direk olarak "boyacı" sözcüğünü kök olarak kabul ederek analiz üretmiştir. Çizelge 3.10'da örneği verilmiştir.

**Çizelge 3.10: +cH Eki Örneği**

<b>OFLMORPH [5]</b>	<b>boyacı boya +Noun+A3sg+Pnon+Nom+DB+Noun+Agt</b>
<b>ITUMORPH</b>	<b>boyacı boyacı+Noun+A3sg+Pnon+Nom</b>

Fiilden fiil yapan eklerden olan "+Hl", "+Hn", "+Ht", "+Hr", "+DHr" gibi ekler için de aynı yöntem uygulandı. Örnekler aşağıdaki tabloda gösterilmiştir. Örneğin "yap" sözcüğü sözlükte var "yapıl" sözcüğü de var "yapıldı" şeklinde girilen bir girdinin analizi "yap" kökü üzerinden değil "yapıl" üzerinden sonuç üretmektedir. Bu sayede girdinin analizinde minimum sayıda üretim sınırı (derivational boundary DB) olan kısım bulunmaktadır. Çizelge 3.11'de ve 3.12'de örnekleri verilmiştir.

**Çizelge 3.11: +Hl, +Hn Ekleri ve Örnekleri**

<b>OFLMORPH [5] +Hl</b>	<b>Yapıldı yap +Verb+DB+Verb+Pass+Pos+Past+A3sg</b>
<b>ITUMORPH +Hl</b>	<b>Yapıldı yapıl+Verb+Pos+Past+A3sg yaptı yap +Verb+Pos+Past+A3sg</b>
<b>OFLMORPH [5] +Hn</b>	<b>sağlandı sağla+Verb+DB+Verb+Pass+Pos+Past+A3sg</b>
<b>ITUMORPH +Hn</b>	<b>sağlandı sağlan+Verb+Pos+Past+A3sg sağladı sağla+Verb+Pos+Past+A3sg</b>

Çizelge 3.11'te ITUMORPH'un analizleri arasına "yaptı" kelimesinin de eklenmesinin sebebi aslında sözlükte yap kelimesi de var ancak yapıldı sözcüğü için "yapıl" kökünden analiz türetildiğini göstermek içindi. Aynı durum "sağlandı" ve "sağladı" sözcükleri için de geçerlidir. Çünkü "sağla" sözcüğü sözlükte bulunmasına rağmen bayrak işaretleri kullanılarak "sağlandı" sözcüğü "sağla" sözcüğünden türetilmemiştir.

Fiilden isim yapan ekler "+mA", "+Hş", "+mAz", "+mHş" için de aynı yöntem kullanıldı. Bununla ilgili örnekler çizelge 3.13'de gösterilmiştir.

Çizelge 3.13'de taşınmaz örneği aynı zamanda gereksiz analizlerin (overgeneration) engellendiğini gösteren bir örnektir. Çünkü sözlükte "taşınmaz" sözcüğü varsa

**Çizelge 3.12: +Hr,+Ht, +DHr Ekleri ve Örnekleri**

<b>OFLMORPH [5] +Ht</b>	<b>körelt körel+Verb+DB+Verb+Caus+Pos+Imp+A2sg</b>
<b>ITUMORPH +Ht</b>	<b>körelt körelt+Verb+Pos+Imp+A2sg</b> <b>körelti körel+Verb+Pos+Past+A3sg</b>
<b>OFLMORPH [5] +Hr</b>	<b>kaçırıldı kaç+Verb+DB+Verb+Caus+DB+Verb+Pass+Pos+Past</b> <b>kaçırdı kaç+Verb+DB+Verb+Caus+Pos+Past+A3sg</b>
<b>ITUMORPH +Hr</b>	<b>kaçırdı kaçır+Verb+Pos+Past+A3sg</b> <b>kaçırıldı kaçırıl+Verb+Pos+Past+A3sg</b>
<b>OFLMORPH [5] +dHr</b>	<b>yazdırdı yaz +Verb+DB+Verb+Caus+Pos+Past+A3sg</b>
<b>ITUMORPH +dHr</b>	<b>yazdırdı yazdır+Verb+Pos+Past+A3sg</b>

bu sözcüğün "taşı" sözcüğünden "taşın" sonrasında "taşınmaz" haline gelmiştir, denilmesi doğru bulunmamıştır. Bunun yanında fiil sözlüğünde "taşın" kökünden de "mAz" ekini alarak "taşınmaz" yapılması da uygun bulunmamıştır. Bu analizlerin engellenmesi ile fazla üretilen sonuçlar filtrelenmiştir. Benzer bir durum "dolma" örneğinde de görülmektedir. Çünkü "dol" fiili "mA" ekini alarak isimleşmiştir. Aynı zamanda "dolma" sözcüğü de isim sözlüğünde bulunmaktadır. Bu durumda "dol" fiilin "mA" ekini alarak "dolma" olması fazladan üretilen analiz olmaktadır. Çünkü zaten "dolma" sözcüğü isim sözlüğünde vardır. "Geçmişti" örneğinde ise "geçmiş" sözcüğü zaten isim sözlüğünde bulunmaktadır bu yüzden "geç" sözcüğünün "mHş" ekini alarak isimleşmiş sıfat haline gelmesi analizdeki fazlalıklardan biridir. Bunun da yine üretilmemesi uygun görülmüştür. Son örnek olan "öpüş" sözcüğü ise isim sözlüğünde "öpüş" sözcüğü bulunduğu için dolayı bu sözcüğün sözlükteki "öpmek" fiilinden "Hş" eki alarak işteşlik anlamı kazanması yada "öpmek" fiilinin yine "Hş" ekini alarak isimleşmesi analizlerdeki üretim sınırı sayısını artırmaktadır. Bunu azaltmak için isim sözlüğümüzde "öpüş" sözcüğü bulunduğu için dolayı "öpmek" fiilinin "Hş" ekini alıp işteşlik ve isim haline geçme durumu engellenmiş oldu.

İsimden fiil yapan ekler olan "+lAş" ve "+lAn" için çizelge 3.14'de örnekler gösterilmiştir. Burada "vedalaştım" örneğinde "veda" isminden "vedalaşmak" fiilinin türetilmesi uygun görülmemiştir. Çünkü sözlükte (TDK) "vedalaşmak" sözcüğü de bulunduğu için dolayı üretilen analizde çıkacak olan üretim sınırı sayısını düşürmek için "vedalaşmak" sözcüğü veda isminden türetilmesi engellenmiştir. Aynı durum "elektiriklendi" sözcüğü için de geçerlidir.

Çizelge 3.13: Fiilden İsim Yapan mA ve mAz Eki Örnekleri

OFLMORPH [5] +mA	dolma dol +Verb+Neg+Imp+A2sg dolma dol +Verb+Pos+DB+Noun+Inf2+A3sg+Pnon+Nom dolma dolma +Noun+A3sg+Pnon+Nom taşıma taş+Noun+A3sg+P1sg+Dat taşıma taşı+Verb+Neg+Imp+A2sg taşıma taşı+Verb+Pos+DB+Noun+Inf2+A3sg+Pnon taşıma taşım+Noun+A3sg+Pnon+Dat
ITUMORPH +mA	dolma dolma+Noun+A3sg+Pnon+Nom dolma dol +Verb+Neg+Imp+A2sg taşıma taş+Noun+A3sg+P1sg+Dat taşıma taşıma+Noun+A3sg+Pnon+Nom taşıma taşı+Verb+Neg+Imp+A2sg
OFLMORPH [5]+mAz	taşınmaz taşı+Verb+DB+Verb+Pass+Neg+Aor+A3sg taşınmaz taşı+Verb+DB+Verb+Pass+Neg+DB+Adj+AorPart taşınmaz taşın+Verb+Neg+Aor+A3sg taşınmaz taşın+Verb+Neg+DB+Adj+AorPart taşınmaz taşınmaz+Noun+A3sg+Pnon+Nom taşınmaz taşınmaz+Adj
ITUMORPH +mAz	taşınmaz taşınmaz+Adj taşınmaz taşın+Verb+Neg+Aor+A3sg
OFLMORPH [5] +mHş	geçmişti geç+Verb+Pos+Narr+Past geçmişti geç+Verb+Pos+DB+Adj+NarrPart+DB+Verb+Zero+Past geçmişti geçmiş+Noun+A3sg+Pnon+Nom+DB+Verb+Zero+Past+A3sg
ITUMORPH +mHş	geçmişti geçmiş+Noun+A3sg+Pnon+Nom+DB+Verb+Zero+Past geçmişti geç+Verb+Pos+Narr+Past+A3sg
OFLMORPH [5] +Hş	öpüşöp+Verb+Pos+DB+Noun+Inf3+A3sg+Pnon+Nom öpüşöp+Verb+DB+Verb+Recip+Pos+Imp+A2sg anlayışımı anla +Verb+Pos+DB+Noun+Inf3+A3sg+P2sg+Acc anlayışımı anla +Verb+Pos+DB+Noun+Inf3+A3sg+P3sg+Acc anlayışımı anlayış+Noun+A3sg+P3sg+Acc anlayışımı anlayış+Noun+A3sg+P2sg+Acc
ITUMORPH +Hş	öpüşöpüş+Verb+Pos+Imp+A2sg öpüşöpüş+Noun+A3sg+Pnon+Nom anlayışımı anlayış+Noun+A3sg+P2sg+Acc anlayışımı anlayış+Noun+A3sg+P3sg+Acc

Çizelge 3.14: lAş ve lAn Örnekleri

ITUMORPH +lAş	vedalaştım vedalaş+Verb+Pos+Past+A1sg
OFLMORPH [5] +lAş	veda+Noun+A3sg+Pnon+Nom+DB+Verb+Become+Pos+Past+A1sg
OFLMORPH [5] +lAn	elektriklendi elektrikle+Verb+DB+Verb+Pass+Pos+Past+A3sg
ITUMORPH +lAn	elektriklendi elektriklen+Verb+Pos+Past+A3sg

## 4. SONLU DURUMLU MAKİNELER

### 4.1 Sonlu Durumlu Makineler

Bu bölümde isim soylu sözcükler için ve fiiller için sonlu durumlu makinelerin çizimleri ve son halleri verilmiştir. Bu makinelerin çizimleri en son Oflazer [5]'in çalışmasında verilmiştir. O dönemden sonra yapılan çalışmalarda bu konuya değinilmiştir. Ancak yapılan çalışmalardaki sonlu durumlu makinelerin güncel çizimleri verilmemiştir. Tez çalışması kapsamında, makinelerin güncel halleri bu alanda çalışma yapanlara bir kaynak olması amacıyla verilmiştir.

### 4.2 İsim Soylu Sözcükler İçin Tasarlanan Sonlu Durumlu Makineler

İsim soylu sözcükler için hazırlanan sonlu durumlu makineler bileşik isimler, özel isimler, temel halde isim olan sözcükler ve sıfatlar olmak üzere dört ana grupta hazırlanmıştır. Bileşik isimler temel halde hem yalın halde hem de 3. tekil şahıs durumunda bulunduğundan girilen bir bileşik isim hiç ek almasa bile analizi hem yalın hem de 3.tekil şahıs olarak üretilmektedir. Çünkü "buzdolabı" denildiğinde hem temel halde bir "buzdolabı" hem de "onun buzdolabı" şeklinde bir anlamı vardır. Özel isimler ise temel halde isimlerin alabildiği eklerin "" ile ayrılmasından dolayı özel isimlere eklenen ekler (morfemler) "" karakterini içermektedir. Sıfatların aldığı ekler de yine temel haldeki isimlerin aldığı eklerden farklı olan birtakım ekler aldığından sıfatlar içinde sonlu durumlu makine tasarlanmıştır.

Bu makinelerin tasarımı yapılırken Oflazer [5]'in, yaptığı çalışmanın çıktıları referans alınmıştır. Bu analizlerdeki hatalar da düzeltilmeye çalışılmıştır. Bu hataların bir kısmı sözlük (lexicon) kaynaklı olmakla birlikte bir kısmı birtakım eklerin ardarda gelmesiyle oluşan hatalardır. Bu tür hataya örnek olarak "patlıcanın" kelimesi verilebilir. Burada normalde "lH+cA+Hn" ekleri sonlu durumlu makinede "lıcanın" şeklinde bir sıralamayı oluşturarak "pat" kelimesinden

türemesine sebep olmuştur. Bu durum aslında doğru bir analiz değildir. Bu hataya sebep olan analiz çizelge 4.1’de incelenebilir.

**Çizelge 4.1:** Fazla Üretim (Overgeneration) Örneği

<b>OFLMORF [5]</b>	<p>patlıcanın pat +Noun+A3sg+Pnon+Nom-DB+Adj+With+DB+Adj+AsIf+DB+Noun+Zero+A3sg+P2sg+Gen</p> <p>patlıcanın pat +Noun+A3sg+Pnon+Nom-DB+Adj+With+DB+Adj+AsIf+DB+Noun+Zero+A3sg+Pnon+Gen</p> <p>patlıcanın patlıcan+Noun+A3sg+P2sg+Nom</p> <p>patlıcanın patlıcan+Noun+A3sg+Pnon+Gen</p>
<b>ITUMORPH</b>	<p>patlıcanın patlıcan+Noun+A3sg+Pnon+Gen</p> <p>patlıcanın patlıcan+Noun+A3sg+P2sg+Nom</p>

Buna benzer bir durum "daire" kelimesinde de mevcuttur. "Daire" kelimesinin analizinde "dair" edatının sanki "e" hal ekini alarak "daire" haline gelmiş gibi bir analiz üretmektedir. Ancak bu durum da yine doğru bir analiz değildir."Daire" kelimesi "dair" sözcüğünden türememesi gerekmektedir. Aslına bakıldığında sözlükte "dair" sözcüğü bulunduğundan bu sözcüğe "e" yönelme eki geldiğinde otomatik olarak "dair+e" ifadesi üretilmiştir. Ancak bu ve bunun gibi birçok istisnai durum kullandığımız veri kümelerinde karşılaştıkça bayrak işaretleri kullanılarak düzeltilmiştir. Çizelge 4.2 ’de bu durumun çıktısı gösterilmiştir.

**Çizelge 4.2:** İstisna Durumun Bayrak İşareti ile Düzeltilmesi

<b>OFLMORF [5]</b>	<p>Daire dair +Postp+PCDatDB+Noun+Zero+A3sg+Pnon+Dat</p> <p>Daire daire +Noun+A3sg+Pnon+Nom</p>
<b>ITUMORPH</b>	<p>Daire daire +Noun+A3sg+Pnon+Nom</p>

### 4.3 Fiil Soylu Sözcükler İçin Tasarlanan Sonlu Durumlu Makineler

Fiil soylu sözcüklerin yapısı isim soylu sözcüklerin aldığı eklere göre çok daha karmaşık bir yapıya sahiptir. Çünkü fiiller "işteş", "etken", "edilgen", "dönüşlü" gibi fiilden fiil yapan ekleri almasının yanında "geçmiş", "gelecek", "şimdiki" ve "geniş" zaman ekleri gibi ekleri alır. Bunların yanında sıraya göre birinci tekil şahıs, ikinci tekil şahıs, üçüncü tekil şahıs eklerini ve bu eklerin çoğullarını belli bir sıralamaya göre alır. Bu durum fiillerin yapısını biraz daha karmaşık hale getirir.

#### 4.4 Bayrak İşaretleri

Bayrak işaretleri aslında Xerox'un bir uygulaması olan Xfst aracının bir özelliğidir. Bayrak işaretleri kullanımı sayesinde birtakım kısıtların hazırlanması, sonlu durumlu makineye hafıza kazandırılması ve analizlerin biraz daha sadeleştirilmesi amaçlanmıştır. Başka diller için biçimbilimsel çözümlemede daha önce bayrak işaretlerinin kullanıldığı çalışmalar vardır. Bunlardan bazıları Endonezya dili için Pisceldo [11], Arapça için Attia [12], ve Farsça için Megerdooian [13]'in çalışmalarıdır. Bayrak işaretleri aslında sözlükteki sözcüklerin sonuna eklenen birtakım karakterlerden oluşmaktadır. Bu eklenen karakterler sayesinde bu sözcüğün hangi eki (morfemi) alıp alamayacağını kontrol edebiliriz. Sonlu durumlu makinenin çalışma prensibine göre şu an bulunan durumdan bir sonraki duruma geçiş şu an bulunan durumda gelen girdiye göre yapılmaktadır. "X" girdisi girildiğinde birinci duruma,"y" girdisi geldiğinde ikinci duruma geçiş şeklinde çalışır. Eğer sözcük ya da morfem bayrak işaretleri içeriyorsa bu çalışma esnasında görülür ve bu duruma göre bir sonraki duruma nereden devam edileceğine karar verilir.

##### 4.4.1 Birleştirme Testi

Birleştirme testi diğer bayrak işaretleri yöntemi arasındaki en sık kullanılan yöntemdir. @U.feature.value@ şeklinde bir şablonu vardır. Çizelge 4.3'te örnekleri verilmiştir. Başında ve sonunda kullanılan "@" karakteri her bayrak işaretinde kullanılan ortak bir karakterdir. Özellik (Feature) ve değer (value) isimleri kod geliştirici tarafından rastgele olarak belirlenir. Bir sözcüğün sahip olduğu bayrak işaretindeki özellik ile ekteki özellik birbiriyle aynıysa bu dizilimin geçerli olabilmesi için değer kısımlarının da aynı olması gerekir. Eğer değer kısmı farklıysa o dizilim sonlu durumlu makine tarafından engellenir.

Çizelge 4.3'de verilen birleştirme testi şablonlarına örnek olarak, isim sözlüğünde zaten "dövüş" sözcüğü bulunduğundan, "döv" kelimesinin "+Hş" morfemini fiilden isim yapan ek olarak alamayacağı düşünülürse, bu durumda "döv@U.Hşyapım.yok@+Hş@U.Hşyapım.var@" şeklindeki bir dizilim sonlu durumlu makine tarafından engellenecektir. Çünkü burada özellik (feature)

**Çizelge 4.3: Birleştirme Testi Örnekleri**

@U.Case.nom@
@U.Case.ACC@
@U.Case.Gen@
@U.num.sing@
@U.num.plur@
@U.gender.masc@
@U.gender.fem@

değerleri "Hşyapım" ifadesi (feature değeri) aynı ancak "value" değerleri olan "yok" ve "var" ifadeleri birbirinden farklı olduğundan bu analiz üretilmeyecektir. Ancak burada sözlükteki bütün "+Hş" morfeplerini alamaz anlamı çıkmamalıdır. Çünkü sözlükte işteşlik eki olan "+Hş" morfemi de var ve bu işteş yapan morfepe @U.Hşyapım.yok@ bayrak işaretini bulundurmadığından "döv" sözcüğü bu morfemi alır. Eğer her ikisinin de hem "feature" hem de "value" değerleri birbiriyle aynı olsaydı "döv" sözcüğü fiilden isim yapan "+Hş" ekini alabilirdi ve şablonu "döv@U.Hşyapım.var@+Hş@U.Hşyapım.var@" şeklinde olurdu. Bayrak

**Çizelge 4.4: Analiz Sırasında Birleştirme Testi ve Çıktıları**

<b>terlik terlik +Noun+A3sg+Pnon+Nom</b>
<b>ter@U.yapımlHk.yok@+lHk@U.yapımlHk.var@ (engellendi )</b>
<b>kitaplık kitaplık+Noun+A3sg+Pnon+Nom</b>
<b>kitap@U.yapımlHk.yok@+lHk@U.yapımlHk.var@ (engellendi )</b>
<b>kitapçık kitapçık+Noun+A3sg+Pnon+Nom</b>
<b>kitap@U.yapımcHk.yok@+cHk@U.yapımcHk.var@ (engellendi )</b>
<b>Pathıcan pathıcan+Noun+A3sg+Pnon+Gen</b>
<b>Pat+lH@U.lHcan.X@+cA@U.lHcan.Y@+Hn (engellendi)</b>

işaretleri bir sözcüğün herhangi bir morfemi almasını engellediği gibi sonlu durumlu makinede ardı ardına gelmesi istenmeyen morfeplerin engellenmesi için kullanılabilir Çizelge 4.4'te gösterilen ilk üç örnek bir sözcüğün özel bir morfemi almaması için hazırlanan örneklerdi. Ancak en son örnekte "+lH" morfemi ile "+cA" morfeminin ardısıra gelmesini engellemek amacıyla bu yöntemi morfepe



dizilimini engellemek için kullandık. Çünkü "pat" sözcüğünden sonra "+IH" ekini alarak "patlı" sonrasında "+cA" morfemini alıp "patlıca" en son "+n" morfemini alarak "patlıcan" haline gelmesi anlamsızdır. Bu yüzden sonlu durumlu makinede bu şekilde anlamsız olan birtakım yolların engellenmesi gerekmektedir.

#### 4.4.2 Olumsuzluk Testi

Sonlu durumlu makine @N.feature.value@ şeklinde bir bayrak işareti ile karşılaştığında buradaki "value" değerinin tümleyeni olan bütün birleştirme bayrak işaretleri doğru kabul edilir örnek verirse @U.yapımlHk.yok@ bayrak işaretine sahip olan bir sözcüğün sonuna eklenen bir morfemde @N.yapımlHk.yok@ şeklinde bir bayrak işareti varsa o sözcük bu bayrak işaretinin olduğu morfemleri ek olarak alamaz. Örnek olarak bakan sözcüğü bizim sonlu durumlu makinemizde "lık" ekini almaz çünkü sözlükte temel halde bakanlık sözcüğü de var. Ancak "lık" haricindeki diğer bütün ekleri alabilir. Dolayısıyla bakan sözcüğüne "lık" ekini alamaması için bir bayrak işareti konulmalıdır. Öncelikle bakan sözcüğünün sonuna "bakan@N.lık.var@" şeklinde bir bayrak işareti "lık" ekine de +lHk@U.lık.var@ şeklinde bir bayrak işareti konuldu. Böylece bakan sözcüğü "lık" eki haricinde isim köklü sözcüklerin alabileceği bütün ekleri alabilir hale gelmiştir.

#### 4.4.3 Gereksinim Testi

Gereksinim testi ile sözlükte bulunan bazı özel kelimelerin istisnai olarak alacağı eklere eklenerek bu morfemi sadece bu özel sözcüklerin alması sağlanır. @R.feature.value@ şeklinde bir şablonu vardır. Sözlükteki +Hş (Dönüştürme eki) sözlükteki bütün fiilere gelmeyen sadece dönüştürme anlamını taşıyan sözcüklere gelen bir ektir. Bu yüzden "+Hş" ekini alacak olan sözcükte @U.donuslu.var@ şeklinde bir bayrak işareti bulunması gerekir. Çünkü @R.donuslu@ şeklindeki bir bayrak işareti "donuslu" özelliğini taşıyan bayrak işaretine gereksinim duyar. Bu özelliğin bulunmadığı sözcükler bu eki alamaz.

#### 4.4.4 İzin vermeme testi

Çalışma kapsamında, sıfat olan sözcüklerin temelde isim kökünde olması sebebiyle ":0" geçişiyle isim makinesine geçişi sağlandı. Ancak isim makinesindeki bazı morfemler sıfat makinesinden gelen bazı sözcüklere eklenemeyeceğinden

**Çizelge 4.5:** Gereksinim Testi Örnekleri

<b>Öp@U.donuslu.var@+Hş@R.donuslu@</b>
<b>İt@U.donuslu.var@+Hş@R.donuslu@</b>
<b>kak@U.donuslu.var@+Hş@R.donuslu@</b>
<b>döv@U.donuslu.var@+Hş@R.donuslu@</b>
<b>böl@U.donuslu.var@+Hş@R.donuslu@</b>

izin verememe testi kullanılarak ":0" geçişiyle isim makinesine bağlanan sıfatların birtakım morfemleri alması engellendi. İzin vermeme testinin şablonu @D.feature@ şeklindedir. Örnek olarak "sarı" kelimesinin isim makinesine geçiş yaptığında "+cH" ekini alamayacağını düşünelim. Bu durumda "sarı@U.Adj.var@+cH@D.Adj@" şeklindeki bir dizilim ile "sarı" sözcüğünün "+cH" ekini alması izin verememe testi ile engellenebilir.

#### **4.5 Sözlükte Olmayan Kelimelerin Analizi**

Temelde sözlük hazırlama ve o sözlüğü genişleterek dildeki olabilecek bütün sözcükleri o sözlüğe ekleme ve genişletme işi çok zaman alan meşakkatli bir iştir. Bu yüzden her kelimenin o sözlükte bulunması imkansızdır. Ancak o sözcüğün aldığı eklere göre sanki sözlükte varmış gibi analiz üretebilen bir sistem geliştirilmiştir. Bu sistem için Xerox'un aracı olan Xfst kullanılmıştır. Bu çalışmada öncelikle sözlükte bulunmayan isimleri tutan bir "guessnounroot" sözcüğü sözlükte olmayan bütün isimleri tutacak şekilde yerleştirildi. Girilen herhangi bir sözcüğün sonuna gelen morfemlere göre sondan ekleri atarak kökü bulunmaya çalışıldı. Bu sayede biçimbilimsel çözümleyicinin sonuç üretmediği sözcükler için bu sözcük sözlükte olmasa bile girilen sözcük için bir sonuç çıkartabilmek amacıyla olmayan kelimelerin de biçimbilimsel analizi yapıldı. Bu yöntem aracımızın ürettiği analiz sayısını artırarak kapsam (Covarege) başarımını yükseltti. Bunun yanısıra sözcüklerin analizlerine bakarak normalde sözlükte olması gerektiği halde sözlükte bulunmayan kelimelerin neler olduğunu anlamak için türetilen sözcüğe "+Guess" etiketini koyuldu. Bu sayede çok hızlı bir şekilde

sözlükte olması gereken ama olmayan sözcüklerin sözlüğe eklenmesi kolaylaştı.  
Lexicon Nouns araba noun-root; masa noun-root; bardak noun-root ; Guesnounroot  
noun-root;

Lexicon Verbs gel Verb-root; git Verb-root; koş Verb-root; döv Verb-root;  
Guessverbroot Verb-root;

Yukarıdaki gösterim bir bilinmeyen sözcükler için hazırlanan sözlük örneğidir. Bu sözlükte bulunmayan isim soylu sözcüklere eklenen morfemleri alan sözcükler "Guessnounroot" ile fiil soylu sözcükleri tutan bilinmeyen fiiller ise "Guessverbroot" ile tutulmaktadır. Eğer bir kelime sözlükte yoksa aldığı eklere göre isim yada fiil olarak analiz edilir çizelge 4.6'da bilinmeyen sözcükler için geliştirilen sistemin çıktısı örnekleri vardır. Bu örneklerden "Slobodan" sözcüğünü incelersek burada "Slobodan" sözcüğü sonlu durumlu makinede tanımlanmış morfemlere göre olası şekilde analiz edilmiş olup ilk analizde "slobo+dan" ("slobo" sözcüğü "dan" ekini almış gibi analiz edilmiş) sonraki analizde "sloboda+n" ("sloboda" sözcüğü kök olarak algılanıp senin "sloboda'n" anlamında bir analiz gerçekleşmiş) en son analiz ise "slobodan" ("slobadan" burada yalın halde hiç ek almamış gibi analiz edilmiş) şeklindedir. Diğer bir örnekte de "antlaşma" sözcüğü yanlış yazılarak "anlaşması'nı" şeklinde girdi olarak verildiğinden ve bu sözcük sözlükte bulunmadığından analizi ITUMORPH değil bilinmeyen kelimeler için hazırlanan sonlu durumlu makine üretmiştir. Burada "nı" morfemi sadece özel isimler için gelebileceğinden dolayı analiz "anlaşması+'nı" şeklinde üretilmiştir. Benzer durum "Goodfellas'ta" sözcüğü için de geçerlidir. Sonlu durumlu makineye göre "Goodfellas" kök "'ta" eki ise bulunma hali olarak algılanıp sonuç bu şekilde üretilmiştir. "Sheila'yı" sözcüğünde de "Sheila" kök olarak algılanarak sonlu durumlu makinede "yı" morfemi ek olarak kabul edilerek buna göre analiz üretilmiştir. Xfst de ya da Hfst de hazırlanan ses olaylarının kurallarını kontrol (debug) etmek için herhangi bir araç yoktur. Eğer bir sözcükte ses olayı varsa ve o sözcükteki ses olayını gerçekleyen kural çalışmıyorsa yine analiz üretilemez. Bu tür durumlarda da bilinmeyen kelimeler için tasarlanan sonlu durumlu makine devreye girer ve anlarız ki o kelime sözlükte olduğu halde sonuç bilinmeyen sözcükler için

Çizelge 4.6: Bilinmeyen Sözcüklerin Analizi

<p>Mumçilo mumçilo+Guess+Noun+A3sg+Pnon+Nom Mumçilo mumçilo+Guess+Noun+Prop+A3sg+Pnon+Nom</p>
<p>Pelisiç pelisiç+Guess+Noun+A3sg+Pnon+Nom Pelisiç pelisiç+Guess+Noun+Prop+A3sg+Pnon+Nom</p>
<p>Slobodan slobo +Guess+Noun+A3sg+Pnon+Abl Slobodan sloboda +Guess+Noun+A3sg+P2sg+Nom Slobodan slobodan +Guess+Noun+A3sg+Pnon+Nom Slobodan slobodan +Guess+Noun+Prop+A3sg+Pnon+Nom</p>
<p>Anlaşması'nı anlaşma+Guess+Noun+Prop+A3sg+P3sg+Acc</p>
<p>Goodfellas'ta goodfellas +Guess+Noun+Prop+A3sg+Pnon+Loc</p>
<p>Sheila'yı sheila +Guess+Noun+Prop+A3sg+Pnon+Acc</p>
<p>Cruise cruis +Guess+Noun+A3sg+Pnon+Dat Cruise cruise +Guess+Noun+A3sg+Pnon+Nom</p>
<p>müstakilen müstaki+Guess+Noun+A3sg+Pnon+Nom<sup>DB</sup>+Verb+Acquire+Pos+Imp+A2sg müstakilen müstakile+Guess+Noun+A3sg+P2sg+Nom müstakilen müstakilen+Guess+Noun+A3sg+Pnon+Nom müstakilen müstakilen+Guess+Noun+Prop+A3sg+Pnon+Nom</p>
<p>makrogöstergelerine makrogösterge+Guess+Noun+A3pl+P3pl+Dat makrogöstergelerine makrogösterge+Guess+Noun+A3pl+P2sg+Dat makrogöstergelerine makrogösterge+Guess+Noun+A3pl+P3sg+Dat makrogöstergelerine makrogösterge+Guess+Noun+A3sg+P3pl+Dat makrogöstergelerine makrogöstergeler+Guess+Noun+A3sg+P2sg+Dat makrogöstergelerine makrogöstergeler+Guess+Noun+A3sg+P3sg+Dat makrogöstergelerine makrogöstergeleri+Guess+Noun+A3sg+P2sg+Dat makrogöstergelerine makrogöstergelerin+Guess+Noun+A3sg+Pnon+Dat makrogöstergelerine makrogöstergelerine+Guess+Noun+A3sg+Pnon+Nom makrogöstergelerine makrogöstergelerine+Guess+Noun+Prop+A3sg+Pnon+Nom</p>

hazırlanan çözümleyiciden gelmiş ve o ses olayı çalışmamış bu şekilde kontrol (debug) işlemi yapılmış olur.

#### 4.6 Tam Sayıların Analizi

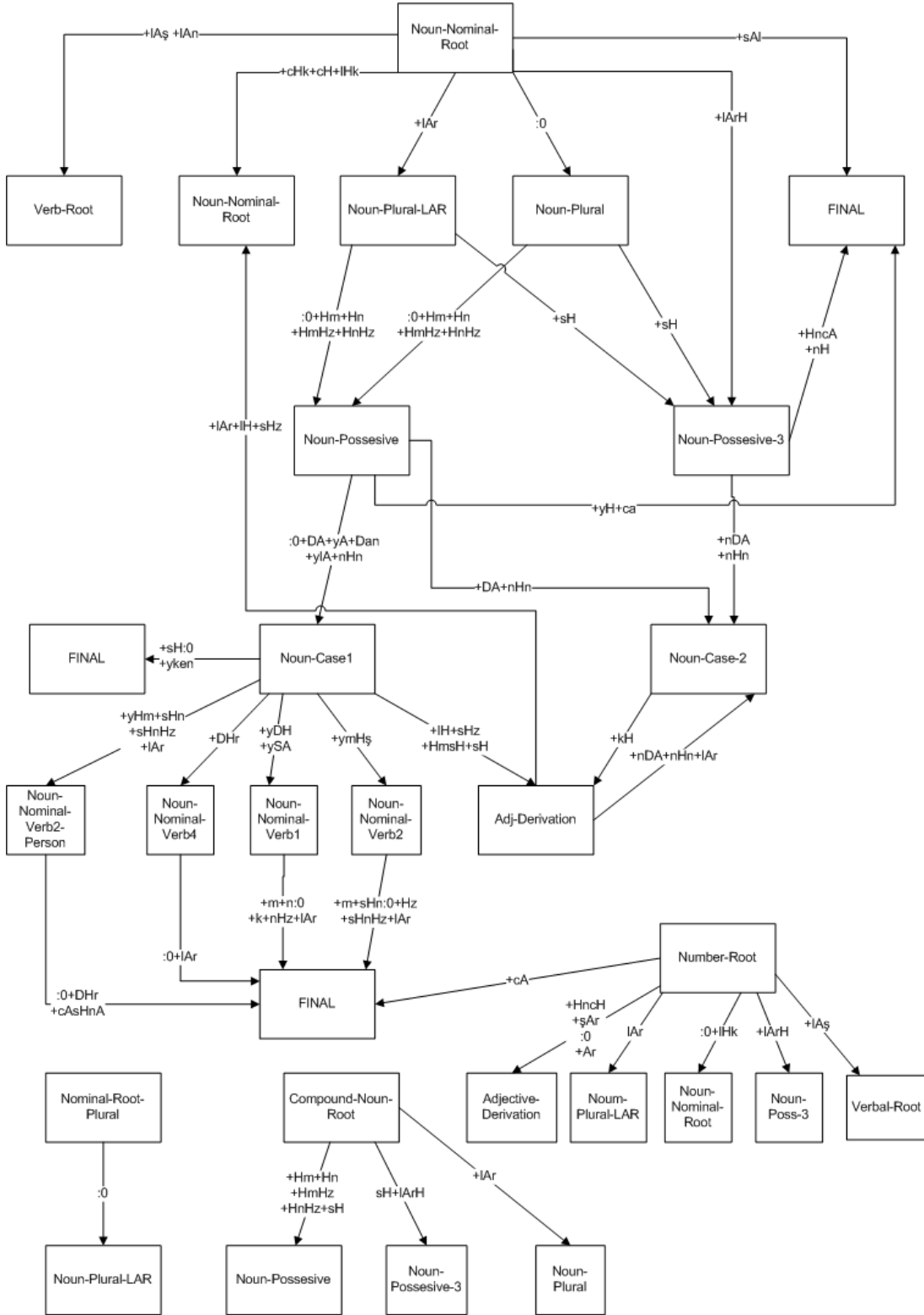
Sayıların analizinin yapılabilmesi için de aslında sözlükte bulunmayan kelimelerin analizinde uyguladığımız yöntemle benzer bir uygulama kullanıldı. Xerox'un Xfst aracı kullanılarak öncelikle girilen sayı için Xfst'de define digit[0|1|2|3|4|5|6|7|8|9] şeklinde sayıları yazarken kullanılacak olan rakamlar tanımlandı. Bundan sonra girilecek her bir sayıyı tutacak olan "numberroot" isminde bir değişken sözlüğe konuldu. Burada tanımlanan digit değişkeni ile "numberroot" değişkeni eşleştirildi. Girilen sayının basamak sayısı ne kadar fazla olursa olsun analiz üretebilmesi sağlanmıştır. Bunun yanında ondalık sayılar içinde analiz üretilmektedir. Çizelge 4.7'de çıktı örnekleri incelenebilir. Mesela "3221" sayısının analizinde "3221+Num+Card" şeklinde bir sonuç üretilmiş burada "+Num+Card" ifadesi üretilen analizin sayı olduğunu ifade etmektedir.

**Çizelge 4.7:** Tam Sayıların Analizi

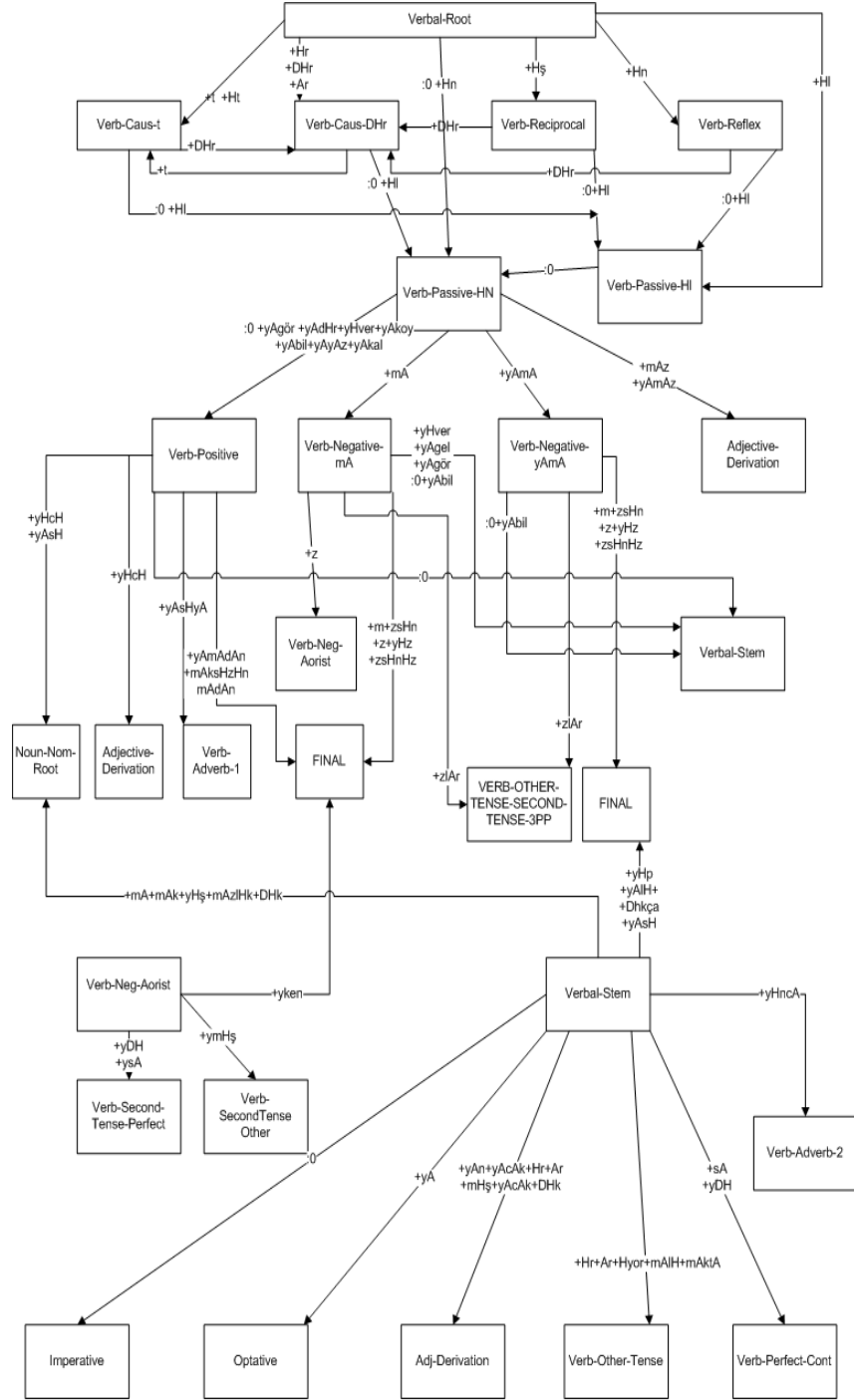
<b>2343 +Num+Card</b>
<b>435678899879900 +Num+Card</b>
<b>3221+Num+Card</b>
<b>3455+Num+Card</b>

#### 4.7 Noktalama İşaretlerinin Analizi

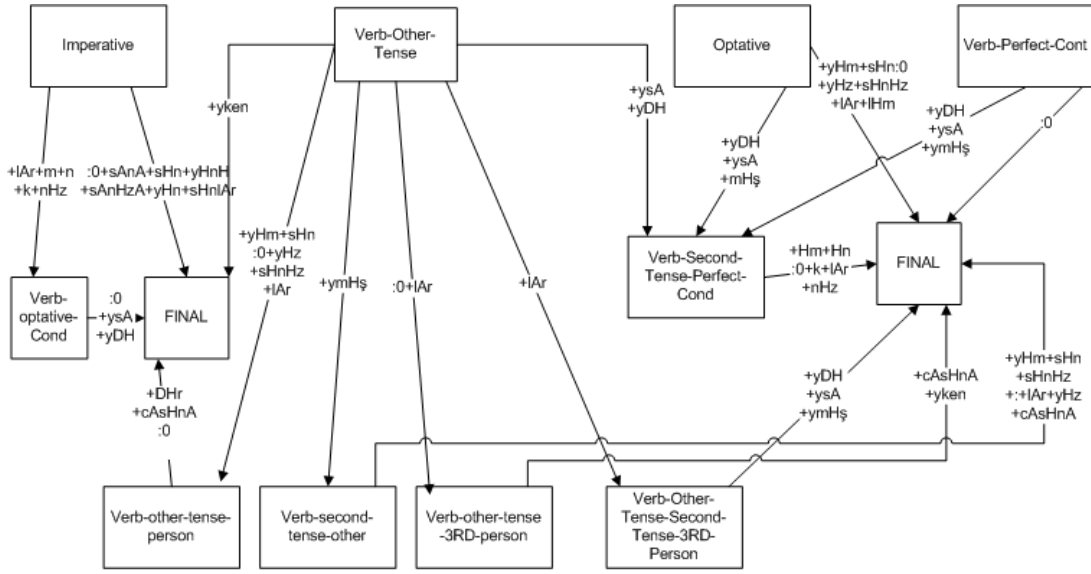
Noktalama işaretleri için de analiz yapılabilmesi için sözlüğe (lexicon) mevcut olan noktalama işaretleri eklenmiş olup bu karakterlerden herhangi biriyle karşılaşıldığında analizi yapılabilmektedir. Noktalama işaretlerinin analizi "; +PUNC", ". +PUNC", ", +PUNC", ". +PUNC" şeklindedir. Yapılan analizlere bakıldığında burada nokta için ". +PUNC" şeklinde bir analiz üretildiği görülmektedir. Bu analizde "+PUNC" kısmı "Punctuation" ifadesinin kısaltması olup noktalama işareti anlamında kullanılmıştır.



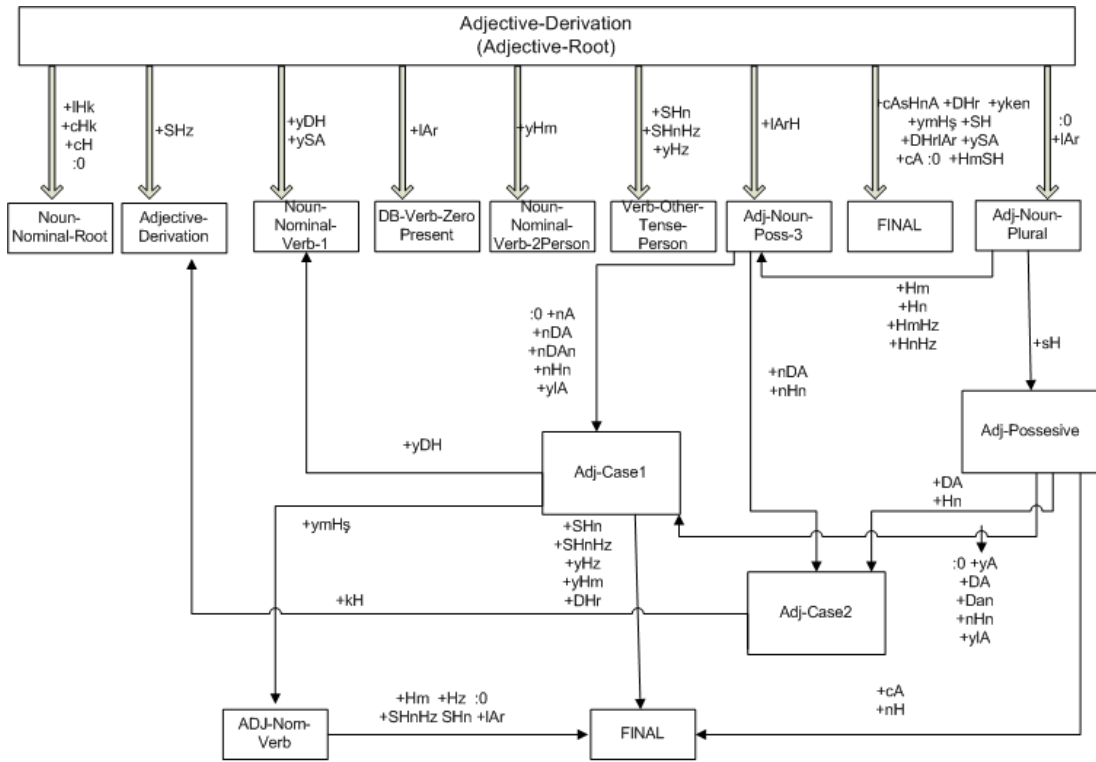
Şekil 4.1: İsimler İçin Sonlu Durumlu Makine.



**Şekil 4.2:** Fiiller İçin Sonlu Durumlu Makine.



Şekil 4.3: Fiiller İçin Sonlu Durumlu Makine.



Şekil 4.4: Sıfatlar İçin Sonlu Durumlu Makine.



## 5. DEĞERLENDİRME

Bütün testler yapılırken kullanılan veri kümeleri ituvalidationset [17], metusabancı [17], [18], [21] ve test [19] veri kümeleridir. Bu veri kümelerinin içinde toplam yaklaşık 1000000 adet sözcük bulunmaktadır. ITUMORPH'un başarımını test etmek için öncelikle üst seviyede çalışan bir sistem olan biçimbilimsel belirsizlik gidericinin başarımına etkisi ölçülmüştür. Yapılan testler sonucu, tez kapsamında geliştirilen biçimbilimsel çözümleyicinin üst DDI [15] araçlarının başarımına olan etkisi gösterilmiştir: ITUMORPH'un kullanılmasının, farklı test kümelerinde biçimbilimsel belirsizlik giderme başarımlarını ortalama %2,83 oranında artırdığı gözlemlenmiştir.

### 5.1 Kapsam Testi

Kapsam testi biçimbilimsel çözümleyicinin başarımını gösteren yöntemlerden biridir. Bu yöntem girdi olarak verilen sözcüklerin kaçı için sonuç üretilip, kaçına sonuç üretilmediğini gösterir. Kapsam testi sonuçları çizelge 5.2'de verilmiştir. Burada testleri yaparken Oflazer [5]'in aracında bilinmeyen kelimelerin analizini yapan bir versiyonu olmadığı için karşılaştırma Oflazer [5] ile ITUMORPH ve bilinmeyen kelimeler için geliştirilen sistem (BKA) ile yapıldı.

**Çizelge 5.1:** Bilinmeyen Kelimeler Analizcisi Kullanılmadan Yapılan Kapsam Testi Sonuçları

	<b>TEST</b>	<b>MST</b>	<b>IVS</b>
<b>OFLMORPH [5]</b>	<b>%77.06</b>	<b>%66.77</b>	<b>%74.10</b>
<b>ITUMORPH</b>	<b>%83.40</b>	<b>%80.97</b>	<b>%83.92</b>
<b>ITUMORPH + BKA</b>	<b>%84.56</b>	<b>%82.96</b>	<b>%86.66</b>

## 5.2 Boyut İndirgeme Testi

Boyut indirgeme testi, biçimbilimsel çözümleyiciye girdi olarak verilen sözcüğün analizinde bulunan üretim sınırı sayısının ne kadar azaltıldığını gösteren bir testtir. Biçimbilimsel çözümleyicinin, girdi olarak verilen bir sözcük için mümkün olduğunca az sayıda üretim sınırı (Derivational Boundary DB) üretmesi, o biçimbilimsel çözümleyicinin boyut indirgeme yönünden başarılı bir sonuç elde ettiği anlamına gelmektedir. Boyut indirgeme testini uygularken her iki sonlu durumlu makineye de yaklaşık bir milyon kelime (965000) girdi olarak verilmiştir. Çizelge 5.3'te girdi olarak verilen yaklaşık bir milyon sözcüğün analiz sonuçlarında toplamda kaç adet üretim sınırı ve yine bir milyon kelimeye karşılık toplamda kaç adet analiz üretildiğini göstermektedir. Örneğin ITUMORPH, girdi olarak verilen bir milyon kelime için yaklaşık 1600000 adet analiz üretmiş ve bu analizlerin içindeki toplam üretim sınırı sayısı yaklaşık 350000 adettir. Çizelge 5.4 sözcük başına düşen ortalama analiz sayısını ve ortalama üretim sınırı (DB) sayısını göstermektedir.

**Çizelge 5.2:** Üretim Sınırı ve Analiz Sayısı

	<b>ÜRETİM SINIRI SAYISI</b>	<b>ANALİZ SAYISI</b>
<b>ITUMORPH</b>	<b>350129</b>	<b>1665350</b>
<b>OFLMORPH [5]</b>	<b>524322</b>	<b>1862704</b>

**Çizelge 5.3:** Üretim Sınırı ve Analiz Oranı

	<b>Ortalama Analiz Sayısı</b>	<b>Ortalama DB Sayısı</b>
<b>ITUMORPH</b>	<b>1,75</b>	<b>0,36</b>
<b>OFLMORPH [5]</b>	<b>2,01</b>	<b>0,54</b>

## 6. SONUÇ VE ÖNERİLER

Bu çalışmada, tez kapsamında geliştirilen ITUMORPH ile biçimbilimsel çözümleme konusunda şimdiye kadar en iyi biçimbilimsel çözümleyici olarak kabul edilen Oflazer [5]'in çalışmasından daha iyi sonuçlar elde edildiği ispatlanmıştır. Bu başarı hem Xerox [2]'un geliştirdiği bayrak işaretleri (flag diacritics) sayesinde hem de yapım ekleri üzerinde yapılan çalışmayla elde edilmiştir. Çünkü dilimizde birçok istisna sözcük bulunmaktadır. Bu istisna sözcüklerin alacağı ek grupları, ya da ek dizimleri ancak bayrak işaretleri yöntemi kullanıldığında istenildiği ölçüde iyi analizler elde edilebilir. Bayrak işaretlerine alternatif olarak sözlük bölütleme yöntemi de kullanılabilirdi. Ancak sözlükte sözcük bölütleme yöntemi kullanılsaydı hem sözlük daha karmaşık olur, hem de sözlüğü hazırlamak çok daha fazla zaman alırdı. ITUMORPH'un sayılar için ve noktalama işaretleri için analiz üretmesi, girdi olarak verilen sözcüklerin analizlerinin daha yüksek bir yüzde ile yapılmasını sağlamıştır. Yani kapsam testi başarımını artırmıştır. Yapım ekleri için yapılan çalışmayla da hem fazla analizlerin önüne geçilmiş, hem de girdi olarak verilen sözcüğün analizinin daha anlaşılır olması sağlanmıştır. Bilinmeyen kelime analizcisi (Unknown word analyzer) bir biçimbilimsel çözümleyici için olmazsa olmazdır. İyi bir biçimbilimsel çözümleyicinin tamamlayıcısıdır. ITUMORPH ile birlikte çalışan bilinmeyen kelimeler analizcisi hem sözlüğün çok hızlı şekilde genişletilmesine hem de girdi olarak verilen her sözcük için aldığı ek gruplarına göre bir analiz üretilmesini sağlamıştır. Bu yüzden biçimbilimsel çözümleyici için bilinmeyen kelime analizcisi çok önemlidir. Bu tez kapsamında yapılan biçimbilimsel çözümleyiciye [tools.nlp.itu.edu.tr](http://tools.nlp.itu.edu.tr) adresinden ulaşılabilmesi sağlanmıştır.

Yapılan testlerde ve karşılaştırmalarda da görüldüğü gibi uygulanan yöntemlerden bayrak işaretlerinin hem fazla analizlerin engellenmesinde hem de istisna olan sözcüklerin alacağı ya da alamayacağı eklerin belirlenmesinde ne derece iyi bir yöntem olduğu anlaşılmıştır. Yapım ekleri üzerinde yapılan çalışmayla, fazla analizlerin engellenmesinin yanında, analizlerin daha anlaşılır hale gelmesi

sađlanmıřtır. Bylece etiketleme yapacak olan kiřilerin daha tutarlı veri kmeleri oluřturmaları sađlanacaktır. Szlkte yapılan iyileřtirilmeler, biimbilimsel zmleyicinin kapsam (covarage) yzdesini artırmıřtır. Bu durum kapsam testi sonularından anlařılmaktadır. Sonraki alıřmalarda, biimbilimsel zmleyici ile entegre alıřan bilinmeyen kelime analizcisinin XFST [2], haricinde HFST - Helsinki Finite State Transducer [4] zerinde de alıřabilir hale getirilmesi sađlanabilir.

## KAYNAKLAR

- [1] **Yılmaz, S.** (2011). TÜRKÇE İÇİN İYİLEŞTİRİLMİŞ BİÇİMBİLİMSEL ÇÖZÜMLEYİCİ, *Tez çalışması*.
- [2] **Beesley, K.R. ve Karttunen, L.** (2003). Finite-state morphology: Xerox tools and techniques, *CSLI, Stanford*.
- [3] **Petrov, S., Das, D. ve McDonald, R.** (2011). A universal part-of-speech tagset, *arXiv preprint arXiv:1104.2086*.
- [4] **Lindén, K., Axelson, E., Hardwick, S., Pirinen, T.A. ve Silfverberg, M.**, (2011). Hfst—framework for compiling and applying morphologies, *Systems and Frameworks for Computational Morphology*, Springer, s.67–85.
- [5] **Oflazer, K.** (1994). Two-level description of Turkish morphology, *Literary and linguistic computing*, **9**(2), 137–148.
- [6] **Hankamer, J.** (1986). Finite state morphology and left-to-right morphology, *West Coast Conference on Formal Linguistics, Bildiri Kitab*.
- [7] **Eryigit, G. ve Adali, E.** (2004). An affix stripping morphological analyzer for Turkish, *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications, Innsbruck, Austria*, s.299–304.
- [8] **Sak, H., Güngör, T. ve Saraçlar, M.** (2008). Turkish Language Resources: Morphological Parser, Morphological Disambiguator and Web Corpus, *GoTAL 2008, cilt5221 of LNCS*, Springer, s.417–427.
- [9] **Akın, A.A. ve Akın, M.D.** (2007). Zemberek, an open source NLP framework for Turkic Languages, *Structure*.
- [10] **Cöltekin, C.** (2010). A Freely Available Morphological Analyzer for Turkish., *LREC*.
- [11] **Pisceldo, F., Mahendra, R., Manurung, R. ve Arka, I.W.** (2008). A two-level morphological analyser for the indonesian language, *Australasian Language Technology Association Workshop 2008, cilt 6*, s.142–150.
- [12] **Attia, M.** (2006). An Ambiguity-Controlled Morphological Analyzer for Modern Standard Arabic Modelling Finite State Networks, *Challenges of Arabic for NLP/MT Conference, The British Computer Society, London, UK*, Citeseer.
- [13] **Megerdooian, K.** (2006). Extending a Persian Morphological Analyzer to Blogs, *Proceedings of the Second Workshop on Persian Language and Computers*.
- [14] **Say, B., Zeyrek, D., Oflazer, K. ve Özge, U.** (2002). Development of a corpus and a treebank for present-day written Turkish, *Proceedings of the eleventh international conference of Turkish linguistics*, s.183–192.

- [15] **Sak, H., Güngör, T. ve Saraçlar, M.**, (2008). Turkish language resources: Morphological parser, morphological disambiguator and web corpus, *Advances in natural language processing*, Springer, s.417–427.
- [16] **Koskenniemi, K.** (1983). Two-Level Model for Morphological Analysis., *IJCAI*, cilt 83, s.683–685.
- [17] **Eryiğit, G.** (2012). The Impact of Automatic Morphological Analysis & Disambiguation on Dependency Parsing of Turkish, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC)*, Istanbul, Turkey.
- [18] **Oflazer, K., Say, B., Hakkani-Tür, D.Z. ve Tür, G.**, (2003). Building a Turkish treebank, *Treebanks*, Springer, s.261–277.
- [19] **Hakkani-Tür, D.Z., Oflazer, K. ve Tür, G.** (2002). Statistical morphological disambiguation for agglutinative languages, *Computers and the Humanities*, **36**(4), 381–410.
- [20] **Şahin, M., Sulubacak, U. ve Eryiğit, G.** (2013). Redefinition Of Turkish Morphology Using Flag Diacritics, *Proceedings of The Tenth Symposium on Natural Language Processing (SNLP-2013)*, Phuket, Thailand.
- [21] **Atalay, N.B., Oflazer, K., Say, B. ve diğerleri** (2003). The annotation process in the Turkish treebank, *Proc. of the 4th Intern. Workshop on Linguistically Interpreteted Corpora (LINC)*.

## ÖZGEÇMİŞ

**Ad Soyad:** Muhammet Şahin

**Doğum Yeri ve Tarihi:** Karabük - 01.06.1984

**Adres:**

**E-Posta:** muhammetsahin@itu.edu.tr

**Lisans:** Gebze Yüksek Teknoloji Enstitüsü

**Y. Lisans:** İstanbul Teknik Üniversitesi

**Mesleki Deneyim ve Ödüller:**

**Yayın ve Patent Listesi:**

### TEZDEN TÜRETİLEN YAYINLAR/SUNUMLAR

▪ **Muhammet, Ş.**, Gülşen, E., Umut, S., Redefinition Of Turkish Morphology Using Flag Diacritics, *SNLP 2013, 2013 10st.*