Gebze Technical University
Department of Computer Engineering
CSE 654 Natural Language Processing
HW1 Report

H.Yavuz ERZURUMLU
181041004

Abstract

In this project, I will design and implement a **finite state transducer(FST)** using **foma** compiler. The fst must handle nouns and accusatives.In Turkish grammer, words ending in -P -Ç -T -K change to -B -C -D -Ğ when suffixed with a vowel.

You can access the whole source code and paper which used in this report from:

- https://github.com/freeloki/CSE654-NLP-Homework01

1 Introduction

The foma compiler is essentially a tool for converting regular expressions to finite automata and transducers. It supports a variety of operations (many more than are found in search-regex formalisms such as the Python re module). The interface also includes tools for performing various tests on automata and transducers, passing words through transducers (doing translations), and importing and exporting transducers in various formats.

2 Turkish Consonant Mutation

Nouns ending hard Un-Voiced Consonants **P, Ç, T, K** mutate to voiced **B, C, D, G, Ğ** when a vowel suffix is added.

2.1 Examples of Turkish Consonant Changes

- kita**p** (book) → kita**b**ın (your book)
- öğü**t** (advice) → öğü**d**üm (my advice)
- ta**t** (taste) → ta**d**ı (its taste)
- ila**ç** (medicine) → ila**c**ı (his medicine)
- ağa**ç** (tree) → ağa**c**ın (the tree's)

3 The foma FST compiler

The *foma* program runs with a read-eval-print loop (REPL), like IPython/Jupyter. That means that each command given is executed and the output is printed, and a new prompt is displayed. You can run scripts of commands by either launching foma with the foma -l flag, or by typing source filename inside foma.

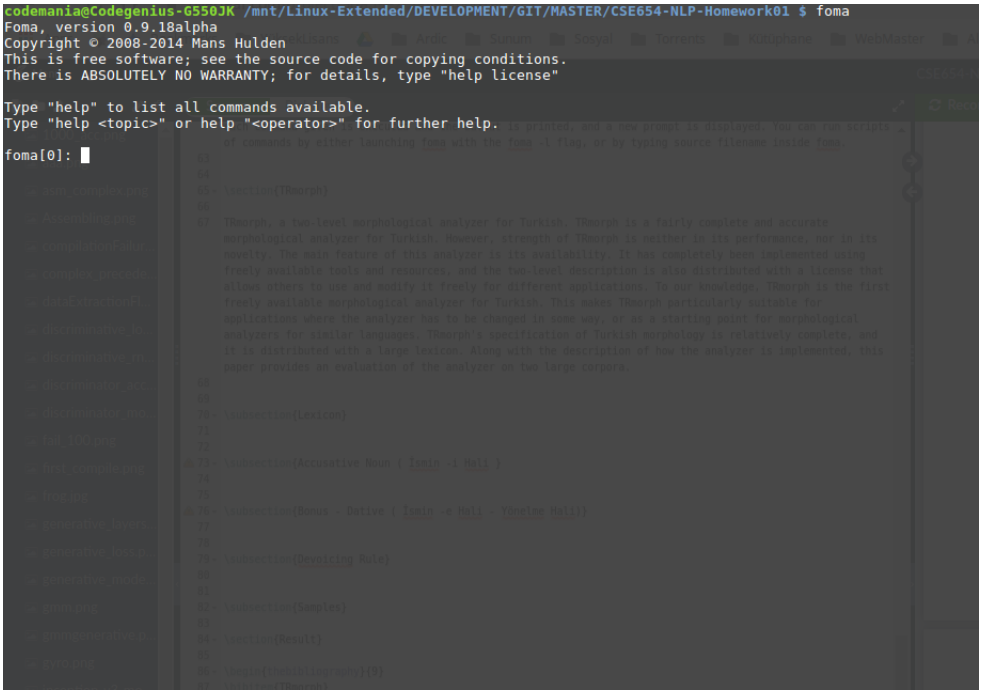

Figure 1: Foma tool.

4 TRmorph[1]

TRmorph[2], a two-level morphological analyzer for Turkish. TRmorph is a fairly complete and accurate morphological analyzer for Turkish. However, strength of TRmorph is neither in its performance, nor in its novelty. The main feature of this analyzer is its availability. It has completely been implemented using freely available tools and resources, and the two-level description is also distributed with a license that allows others to use and modify it freely for different applications. To our knowledge, TRmorph is the first freely available morphological analyzer for Turkish. This makes TRmorph particularly suitable for applications where the analyzer has to be changed in some way, or as a starting point for morphological analyzers for similar languages. TRmorph's specification of Turkish morphology is relatively complete, and it is distributed with a large lexicon. Along with the description of how the analyzer is implemented, this paper provides an evaluation of the analyzer on two large corpora.

4.1 Lexicon

I have used all nouns in TRmorph's lexicon file. Thats why I have 12189 state.

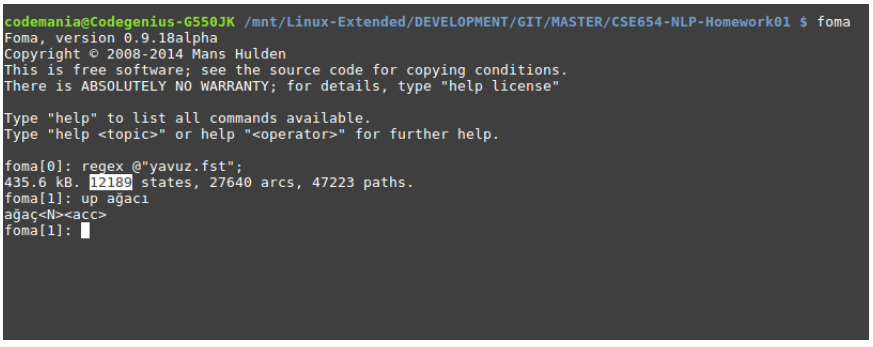

Figure 2: FST State

```
Multichar_Symbols ^I
                  ^(s) ^(y)
                  @RB @MB
                  @NCOMP

LEXICON N
%<N%>:@RB       Ncont;

LEXICON Ncont
    NcompCont; ! includes final


LEXICON NcompCont
            ENDLEX;
            NCaseNoGen;

LEXICON  NCaseNoGen
%<acc%>:^(y)^I@MB   ENDLEX;
%<dat%>:^(y)^A@MB   ENDLEX;

LEXICON ENDLEX #;
```

Figure 3: Lexical States

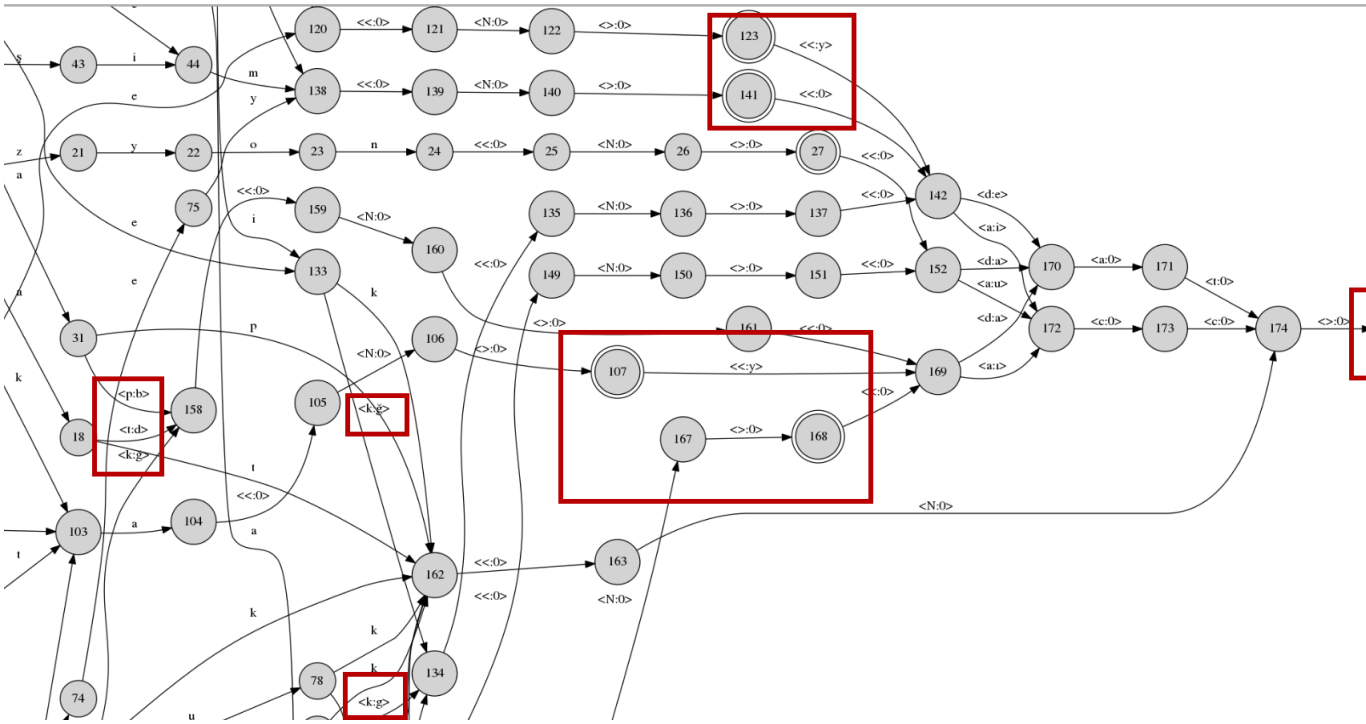

Figure 4: Dependency Graph

4.2 Accusative Noun (İsmin -i Hali

This simple regex handles accusatives:
%<acc%>:(̂y)Î@MB ENDLEX;

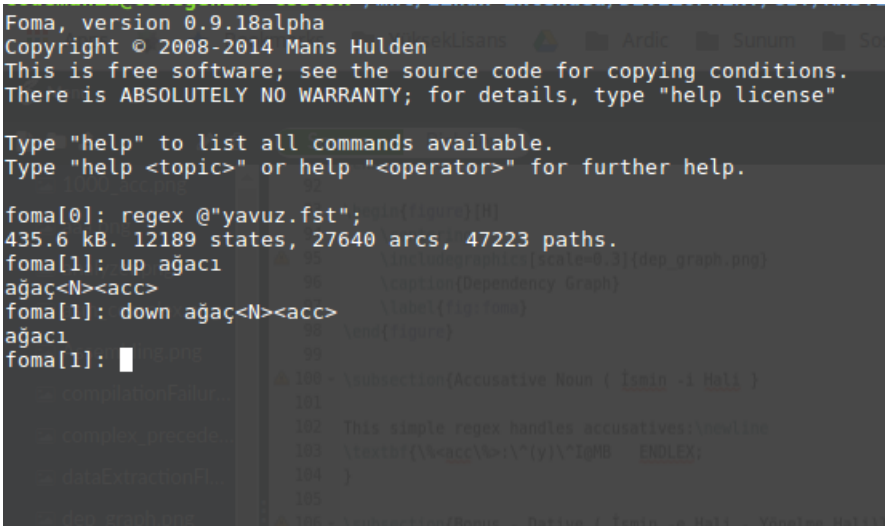

Figure 5: Accusative with devoicing

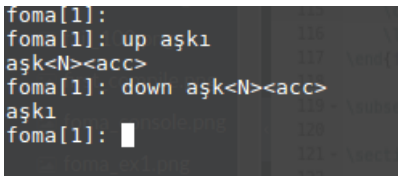

Figure 6: Accusative without devoicing

4.3 Bonus - Dative (İsmin -e Hali - Yönelme Hali)

Using same method we can handle datives as well:
%<dat%>:(̂y)Â@MB ENDLEX;

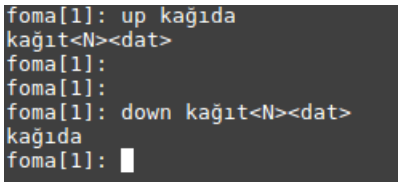

Figure 7: Dative with devoicing

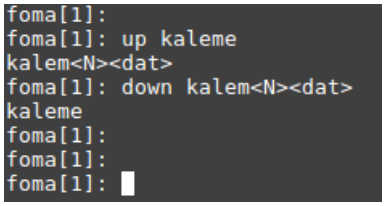

Figure 8: Dative without devoicing

4.4 Devoicing Rule

```
define FSDevoicing  [%^c -> c,
                     %^g -> ğ,
                     %^K (->) ğ,
                     %^K -> y,
                     %^p -> b,
                     %^t -> d ||     _ IntSym* [ Vowel | Vxx ]  ]
                .o.  %^k -> ğ || \n _ IntSym* [ Vowel | Vxx ]
                .o.  %^k -> g || n  _ IntSym* [ Vowel | Vxx ]
                .o. [%^c -> ç,
                     %^g -> g,
                     %^K -> k,
                     %^p -> p,
                     %^t -> t,
                     %^k -> k];
```

Figure 9: Devoicing Rule

4.5 Samples

```
???
apply up> asidi
asit<N><acc>
apply up> aşkı
aşk<N><acc>
apply up> bayrağı
bayrak<N><acc>
apply up> bilişimi
bilişim<N><acc>
apply up> çadırı
çadır<N><acc>
apply up> camı
cam<N><acc>
apply up> çarığı
çarık<N><acc>
apply up> cengi
cenk<N><acc>
apply up> çırağı
çırak<N><acc>
apply up> denklemi
denklem<N><acc>
apply up> dolabı
dolap<N><acc>
apply up> esası
esas<N><acc>
apply up> evladı
evlat<N><acc>
apply up> fayı
fay<N><acc>
apply up> fındığı
fındık<N><acc>
apply up> gamı
gam<N><acc>
apply up> gömleği
gömlek<N><acc>
apply up> hızı
hız<N><acc>
apply up> hududu
hudut<N><acc>
apply up> icatı
???
apply up> icadı
icat<N><acc>
apply up> ırkı
ırk<N><acc>
apply up> ışığı
ışık<N><acc>
apply up> işlemi
işlem<N><acc>
apply up> jeli
jel<N><acc>
apply up> jeofiziği
jeofizik<N><acc>
apply up> kaçığı
kaçık<N><acc>
apply up> kuzeyi
kuzey<N><acc>
apply up> leyleği
leylek<N><acc>
```

Figure 10: Samples 1

Figure 11: Samples 2

5 Result

TRmorph is very useful tool for Turkish morphological analyzer. I have implemented very simple version of it which only handles **N , Accusative, Dative** one. We can easily define and implement regular expressions via **foma**.

References

[1] Çağrı Çöltekin (2010) A Freely Available Morphological Analyzer for Turkish. In: Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)

[2] `https://github.com/coltekin/TRmorph` free morphological analyzer repo.

[3] `https://fomafst.github.io`

[4] http://sureksiz-sert-unsuzlerin-yumusamasi.nedir.org/