NATURAL LANGUAGE PROCESSING MIDTERM EXAM

Name – Surname : Betül Berna Soylu

Number : 171805019

Class : CSE426

Question-1) Select a natural Turkish text fragment where you will find five (5) successive verbal predicates at least one of which consists of one verbal stem plus (4) inflectional morphemes. (10 points)

Sentence 1 → Bu konuyu 1 kere değil 10 kere de anlatsanız anlamayacaktım.

Sentence 2 → Sınıfın tembel öğrencisi yine bilgisayar ödevini yapmadı.

Sentence 3 → Aldığım hediyeyi sevdiğim kadar kendisi sevmedi.

Sentence 4 → Benim gördüğüm hırsızı o da görmüştü.

Sentence 5 → Türkiye'nin başkenti olan Ankara'ya hala gitmedim.

Question-2) Design and draw a finite-state automaton that models how the verbs you have found can be morphologically analyzed. (40 points)

anla, yap, sev, gör , git → verb

-me, -ma → not

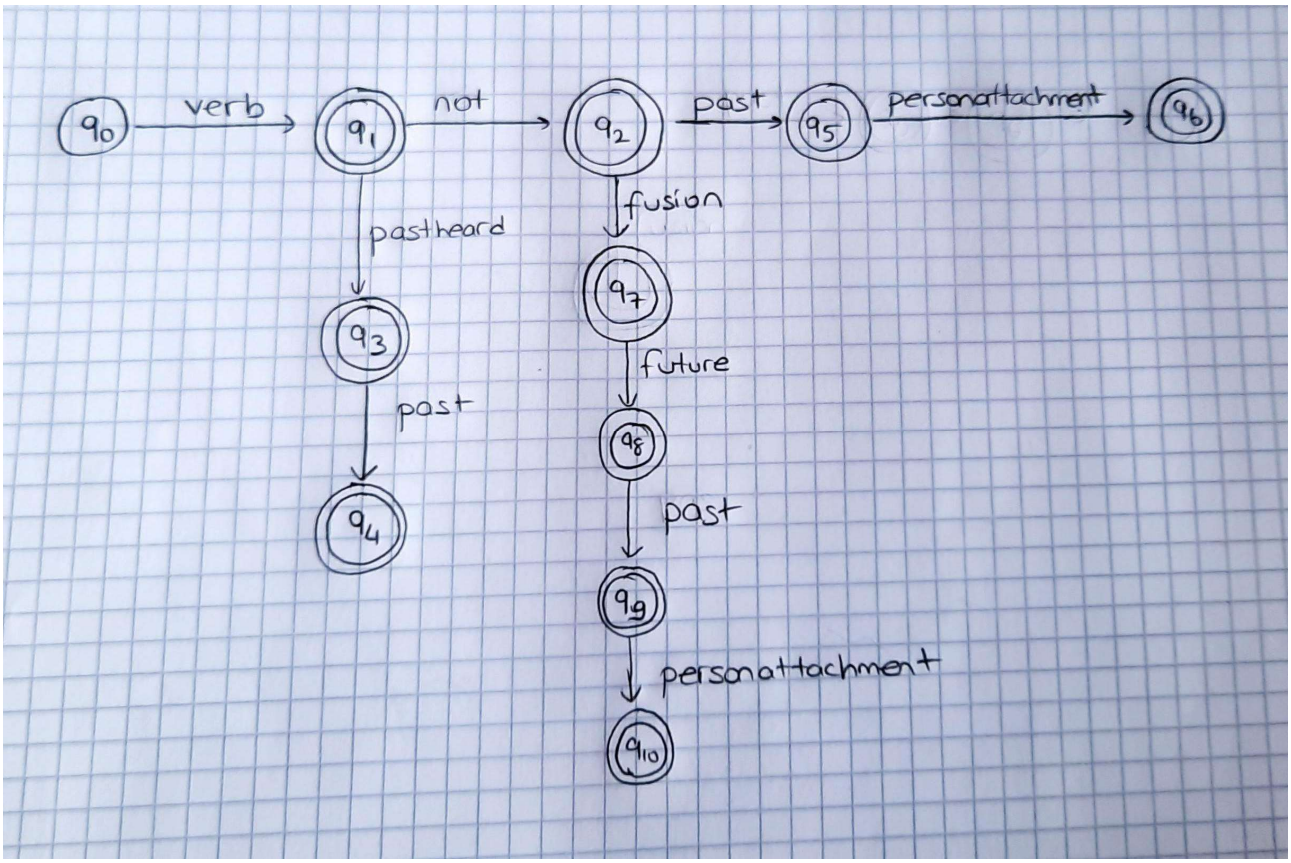
-müş → pastheard

-tü, -di, -tı, -dı → past

-y → fusion

-acak → future

-m → personattachment



Question-3) Write a Prolog program that can morpho-phonologically analyze your verbs. Show that your program correctly analyzes and detects each of the inflectional morpheme in each of your verbs with appropriate queries. (50 points)

```
1 ?- chdir('C:/Users/BS/Desktop/Prolog/').
true.

2 ?- guitracer.
% The graphical front-end will be used for subsequent tracing
true.

3 ?- set_prolog_flag(prompt_alternatives_on, groundness).

4 ?- consult('File2').

5 ?- analyzer(anlamayacaktım, X).
X = [verb, not, fusion, future, past, personattachment] ;
false.

6 ?- analyzer(yapmadı, X).
X = [verb, not, past] ;
false.

7 ?- analyzer(sevmedi, X).
X = [verb, not, past] ;
false.
```

```
1 ?- chdir('C:/Users/BS/Desktop/Prolog/').
true.

2 ?- guitracer.
% The graphical front-end will be used for subsequent tracing
true.

3 ?- set_prolog_flag(prompt_alternatives_on, groundness).

4 ?- consult('File2').

5 ?- analyzer(görmüştü, X).
X = [verb, pastheard, past] ;
false.

6 ?- analyzer(gitmedim, X).
X = [verb, not, past, personattachment] ;
false.
```

PROLOG CODE

```
initial(q0).


%durumları belirleme

final(q1).

final(q2).

final(q3).

final(q4).

final(q5).

final(q6).

final(q7).

final(q8).

final(q9).

final(q10).


%çevirilecek ifadelerin yollarını belirleme

t(q0,verb,q1).

t(q1,not,q2).

t(q1,pastheard,q3).

t(q3,past,q4).

t(q2,past,q5).

t(q5,personattachment,q6).

t(q2,fusion,q7).

t(q7,future,q8).

t(q8,past,q9).

t(q9,personattachment,q10).
```

```
%fiilleri belirleme
allomorph(yap,verb).
allomorph(anla,verb).
allomorph(sev,verb).
allomorph(gör,verb).
allomorph(git,verb).
%olumsuzluk eklerini belirleme
allomorph(ma,not).
allomorph(me,not).
allomorph(müş,pastheard).
%kaynaştırma harfini belirleme
allomorph(y,fusion)
%geçmiş zaman kelimelerini belirleme
allomorph(dı,past).
allomorph(di,past).
allomorph(tü,past).
allomorph(tı,past).
%gelecek zaman ve kişilik ekini belirleme
allomorph(acak,future).
allomorph(m,personattachment).
analyzer(String,List_of_Morphemes):-
  initial(State),
  analyzer(String,State,List_of_Morphemes).
analyzer('',State,[]):- final(State).
analyzer(String,CurrentState,[Morpheme|Morphemes]):-
  concat(Prefix,Suffix,String),
  allomorph(Prefix,Morpheme),
  t(CurrentState,Morpheme,NextState),
  analyzer(Suffix,NextState,Morphemes).
```