PARSING TURKISH SENTENCES FOR NATURAL LANGUAGE WATERMARKING

Ersin İhsan Ünkar

June 2007

Submitted to the Department of Computer Engineering
in partial fulfilment of the requirements
for the degree of
Bachelor of Science
in
Computer Engineering
Boğaziçi University

Abstract

This document explains the procedures, methodologies, and design decisions for transforming a Treebank for Turkish Language to a hierarchical model that is efficient to use in Natural Language Watermarking. It introduces a very common method with some changes in order to allow researchers to design and build beter tools for Natural Language Processing.

This document also presents a phrase-structural grammar parser for Turkish language and describes it in its theoretical and practical aspects.

Contents

1 Introduction **4**
- 1.1 What is Natural Language Watermarking? 4
- 1.2 Literature Overview 4

2 Sentence Watermarking and Parsing **5**
- 2.1 Description 5
- 2.2 Motivations behind a new model: Watermarking 5
- 2.3 Dependency Parsing 6
  - 2.3.1 METU-SabancıTurkish Treebank 6
  - 2.3.2 Wordnode-Relationarc Dependency Tree 7
  - 2.3.3 Wordnode-Relationnode Dependency Tree and Treebank Transformation 8
  - 2.3.4 Transformation Application 10

3 Phrase-Structure Parsing **15**
- 3.1 Transformational Grammars 15
- 3.2 Definite Clause Grammars and Prolog 15
- 3.3 Conversion from Tranformational Grammar to Context Free Grammar 17

4 Practice and Results **49**
- 4.1 Dependency Parsing 49
- 4.2 Phrase Structure Parsing 51

5 Conclusion **60**

6 Appendix A **62**
- 6.1 Transform.java 62

7 Appendix B **66**
- 7.1 Meskill's Transformational Grammar 66

8 Appendix C **70**
- 8.1 Context-Free Grammar used for parser 70

9 Appendix D **93**
- 9.1 grammarall.txt 93

1 Introduction

1.1 What is Natural Language Watermarking?

Sentence watermarking is embedding additional information in texts considering subliminal difference, hidden information transport of content and authorship authentication, and finally enriching the text with metadata.[1]

1.2 Literature Overview

Digital text is one of the most used types of data in daily life. As emergence of Internet is thought, Internet has become the main medium for information sharing and gathering. Many publications from daily newspapers to journal articles are publicly available, and e-mail is the main communication tool of many people. Digital text can be seen a very good host for information hiding.
Available methods for information hiding in natural language text can be grouped under two categories. First group is based on generating a new text document for a message. Second group is based on linguistic modifications on the text[2].
Studies for text watermarking of Turkish has recently started. Earlier studies investigated synonym substitution and punctuation alternations[3]. But it is observed that syntactic manipulations will widen information hiding area in a given text[1]. This is due to the constituent order independence of Turkish[4] and agglunative structure of it, where features such as case, person, tense, aspect, modality and voice are expressed as suffixes[5].
Second instinct for consideration of morpho-syntactic modifications is that its negative impact on semantics is less compared to other methods according to the tests given to people. Semantic equivalence, i.e. there should no or little difference in meaning between original and marked text, is the top issue for this study which makes almost impossible to realize that the marked sentence is altered by a computer-based system at first sight for a 3rd party.
As natural language processing is involved, sentence parsing and treebanks are of major concern. Treebanks are generally organized in two main parsing methods:

1. Dependency Parsing
2. Phrase Structure Parsing

And in most of the cases, a representation in one type of parsing can be changed into the other type since syntax and semantics are closely related with each other[6]. And one step further is to combine these two in a single model where whole information - which word is semantically related with another and how phrases are organized within the sentence - about a sentence is condensened[7].During our studies on watermarking subject, we encountered some difficulties with the structure of Turkish treebank used, and this led us to build a new format for treebank.
Later studies led me to build a phrase structure grammar(PSG) parser. Although Turkish does not have a basic set of context-free grammar rules, transformational studies[8] were useful to create a grammar from scratch. Parser is implemented using an extension to Prolog called Definite Clause Grammar which is an adaptation of context-free grammar.[9]
In this paper I will present the design decisions and methodologies and procedures to transform the treebank,to measure the success of the model I propose for a Turkish treebank and a PSG parser for Turkish language.

2 Sentence Watermarking and Parsing

2.1 Description

The aim of this project is to build a model for representation of Treebank to make processing of Turkish language for watermarking more efficient and create a parser for Turkish language. It presents a transforming algorithm for an available Turkish Treebank,measures the success of proposed model. In section 3, design of a Turkish parser is explained in details with CFG rule generation of Turkish from a tranformational grammar.

2.2 Motivations behind a new model: Watermarking

Sentence watermarking is a method which modifies sentences linguistically to encode a message in a text. The key point in watermarking is to weave through text by applying feasible syntactic watermarking tools on each occurence by using outer references such as dictionary or wordnet. *(Refer to Figure 1)*
As the feasible tools, which are morpho-syntactic modification tools as stated before, the only path of original text is multiplexed by a combinatorial factor

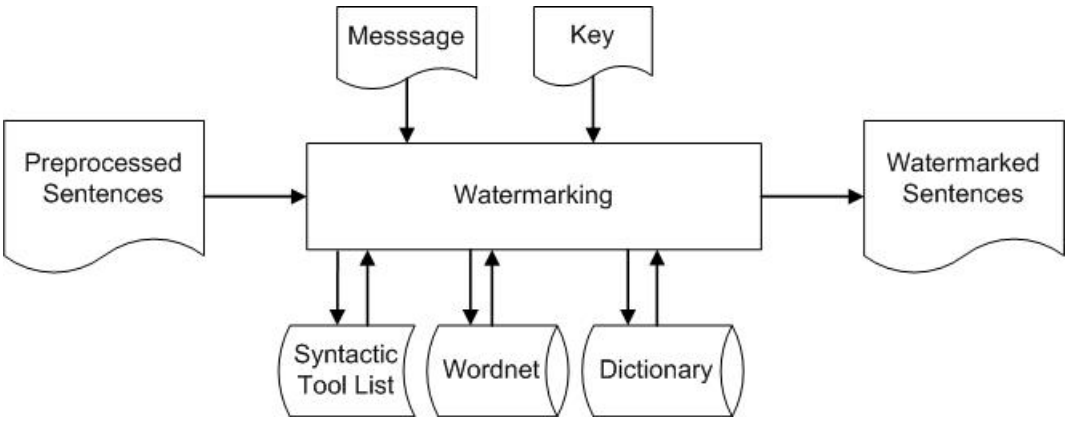

Figure 1: High Level Design of Watermarking

of these tools, and one of these will be selected according to a signature algorithm. *(Refer to Figure 2)*

2.3 Dependency Parsing

2.3.1 METU-Sabancı Turkish Treebank

The Turkish Treebank is a subcorpus of the METU Turkish Corpus, which is 2-million word corpus of post-1990 written Turkish, sampled from various genres. It contains nearly 5600 sentences in a special format, which has an example below for the sentence "Gögsü körük gibi inip kalkıyordu.":

```
<S>
<W IX="1" LEM="" MORPH="" IG="[(1,'gs+Noun+A3sg+P3sg+Nom')]"
REL="[4,1,(SUBJECT)]"> Gögsü </W>
<W IX="2" LEM="" MORPH="" IG="[(1,'krk+Noun+A3sg+Pnon+Nom')]"
REL="[3,1,(OBJECT)]"> körük </W>
<W IX="3" LEM="" MORPH="" IG="[(1,'gibi+Postp+PCNom')]"
REL="[4,1,(MODIFIER)]"> gibi </W>
<W IX="4" LEM="" MORPH="" IG="[(1,'inip_kalk+Verb+Pos+
Prog1+Past+A3sg')]"REL="[5,1,(SENTENCE)]" ORG_IG1="[(1,'in+Verb+Pos')
(2,'Adv+AfterDoingSo')]" ORG_IG2="[(1,'kalk+Verb+Pos+Prog1+
Past+A3sg')]"> inip_kalkıyordu </W>
<W IX="5" LEM="" MORPH=" " IG="[(1,'.+Punc')]"
```

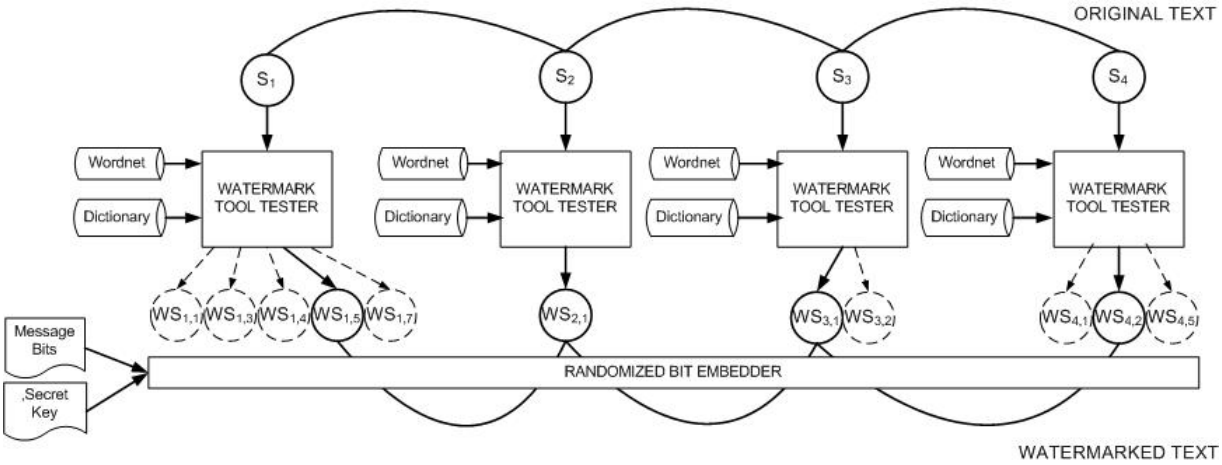

Figure 2: Watermarking in details

```
REL="[,( )]"> .  </W>
</S>
```

In this representation <S> tag identifies a sentence and <W> tag identifies words in the sentence. Attributes of <W> tag is as following:
IX: word place in the sentence
LEM: root in the Turkish lexicon
MORPH: morphological root of the word.
IG: morphological analysis of the word.
REL: relative dependence to other words. Composed of a relation and a reference
ORIG_IGn: separate analysis of multi-word structures.
Relation set from the same study is composed of sentence, subject, object, modifier, determiner, focus-particle, question particle, vocative, classifier, dative adjunct, ablative adjunct, locative adjunct, instrumental adjunct, co-ordination, relativizer, possesive and at the syntactic level surface syntactic relations are encoded using this set regarding relation links form a graph[10]. *(Refer to Figure 3)*
This representation is instrumental for many tools, but some tools require a detailed analysis in hierarchical structure of syntax. Considerations on these tools led us to build a new representation model and a tool for pre-processing of the sentences in METU-SabancıTreebank. *(Refer to Figure 4)*

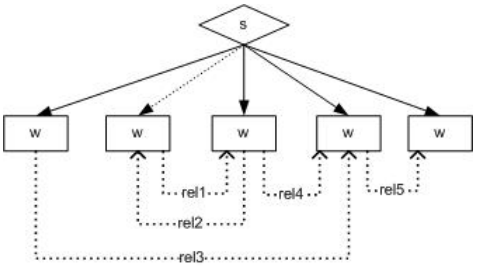

Figure 3: Structure of a Treebank Sentence

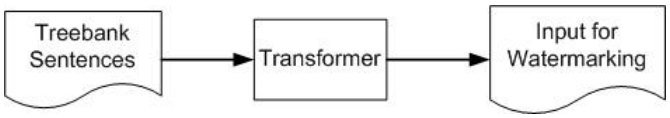

Figure 4: Pre-Processing for Watermarking

2.3.2 Wordnode-Relationarc Dependency Tree

Trees are the most used way of modeling a hierarchy and observations for Turkish show that[5] dependency links between words when drawn are originating one from a word and ending at just one word. This property ensures that a Turkish sentence can be represented as a typed dependency tree, where all the dependencies are shown by means of word nodes and relation arcs.
It is trivial to transform Treebank sentences into such representation. Traversing all the nodes and appending a node as a child of another when there is a referential link from one to the other complete the whole task. This rep-

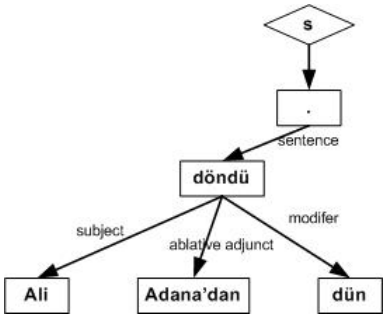

Figure 5: Tree for sentence "Ali Adana'dan döndü dün." and "Ali Adana'dan dün döndü."

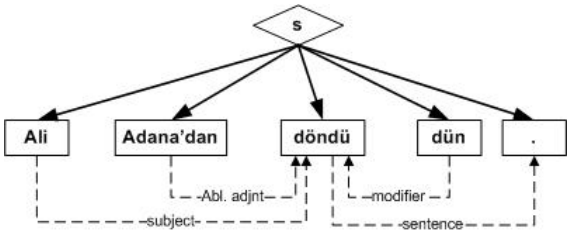

Figure 6: Non-structural representation for sentence "Ali Adana'dan dün döndü."

resentation may be very useful for static-constituent-order languages, but as it is seen on the figure for two example sentences, although order relation is known between sibling nodes, this information is lost for nodes in different levels. *(Refer to Figure 5, Figure 6)* This is due to the free-constituent order property of Turkish language and causes two different sentences having the same tree.

2.3.3 Wordnode-Relationnode Dependency Tree and Treebank Transformation

Since order of the constituents is important in our studies, we are led to a second type of tree representation where both words and relations are nodes. This type of representation prevents the ambiguity in a tree by encoding relations not as arcs from parent to child but as functional nodes at level of the independent word. *(Refer to Figure 8, Figure 9)* This is theoretically done by the algorithm below:

1. Create functional node for each word node labeled as relation of that word. *(Refer to Figure 7)*

2. Append the functional node as another functional node's child if there's a relation from that to the other.

3. Correct the order of the functional nodes according to the order of their word nodes.

Transformation algorithm will be detailed in the following section because of the exceptions resulting XML encoding of the sentences.

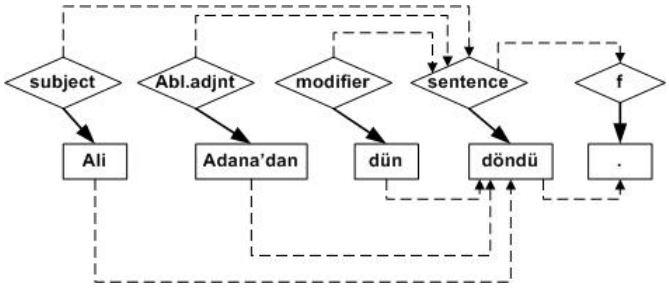

Figure 7: Sentence "Ali Adana'dan dün döndü." after step 1

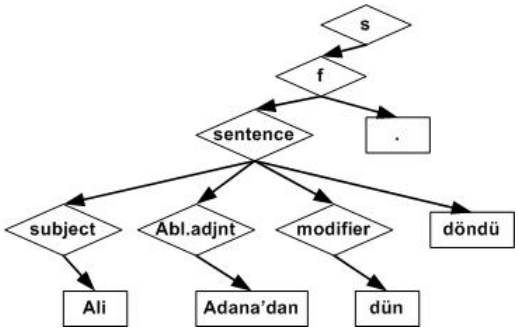

Figure 8: Tree repsresentation for sentence "Ali Adana'dan dün döndü."

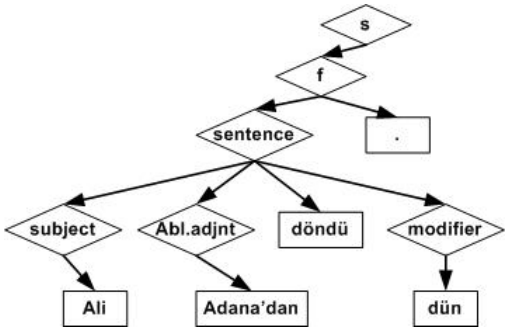

Figure 9: Tree repsresentation for sentence "Ali Adana'dan döndü dün."

2.3.4 Transformation Application

Transformation application is coded in Java language because of its object orientation and widely-used XML API. Since treebank sentences are not so well-formed there are some additional steps in algorithm to eliminate the exceptions that may occur. The algorithm *(Complete listing of the code is in Appendix A.)* is as follows:

```
parse XML file of treebank sentence.
create root node s
for each node ns *step 1
  ix = order(ns)
  rel = relation(ns)
  ref = reference(ns)
  create a functional node fix with function = rel
    and reference = ref
  create the same node ns
  append ns to fix as its child
endfor
for each functional node nf in increasing order
  if relation(nf) is null then *step 2
    if order(nf) is last
      append nf to root s as its child
    else
      append nf to functional node before as its child
    end if
  else
    ref=reference(nf)
    find fref node
    for each child of fref as ch in increasing order
      if(order(nf)<order(ch)) then
        insert nf into fref as its child before ch
        exit for
      endif
    endfor
    append nf to fref as its child * step 3
  endif
endfor
```

print tree into a new file.
An example of a detailed sentence, "Olası Irak operasyonunda Türkiye'den taleplere ilişkin son kararın alınacağı MGK öncesinde Ankara'ya gelen ABD'nin iki bakan yardımcısı yirmi milyar dolarlık paket açacak.", in Treebank representation and new representaion is shown *(Refer to Figure 10)*below in order to clarify the output structure:
Treebank_form.xml

```
<S>
<W IX="1" LEM="" MORPH=" " IG="[(1,'olas+Adj')]"
REL="[3,1,(MODIFIER)]"> Olası</W>
<W IX="2" LEM="" MORPH=" " IG="[(1,'Irak+Noun+Prop+A3sg+Pnon+
Nom')]" REL="[3,1,(CLASSIFIER)]"> Irak </W>
<W IX="3" LEM="" MORPH=" " IG="[(1,'operasyon+Noun+A3sg+P3sg+
Loc')]" REL="[8,1,(MODIFIER)]"> operasyonunda </W>
<W IX="4" LEM="" MORPH=" " IG="[(1,'Trkiye+Noun+Prop+A3sg+
Pnon+Abl')]" REL="[5,1,(MODIFIER)]"> Türkiye'den </W>
<W IX="5" LEM="" MORPH=" " IG="[(1,'talep+Noun+A3pl+Pnon+Dat')]"
REL="[6,1,(OBJECT)]"> taleplere </W>
<W IX="6" LEM="" MORPH=" " IG="[(1,'ilikin+Postp+Dat')]"
REL="[8,1,(MODIFIER)]"> ilişkin </W>
<W IX="7" LEM="" MORPH=" " IG="[(1,'son+Adj')]"
REL="[8,1,(MODIFIER)]"> son </W>
<W IX="8" LEM="" MORPH=" " IG="[(1,'karar+Noun+A3sg+Pnon+
Gen')]" REL="[9,2,(SUBJECT)]"> kararın </W>
<W IX="9" LEM="" MORPH=" " IG="[(1,'al+Verb')(2,'Verb+Pass+Pos')
(3,'Adj+FutPart+P3sg')]" REL="[10,1,(MODIFIER)]"> alınacağı</W>
<W IX="10" LEM="" MORPH=" " IG="[(1,'mgk+Noun+A3sg+Pnon+Nom')]"
REL="[11,1,(CLASSIFIER)]"> MGK </W>
<W IX="11" LEM="" MORPH=" " IG="[(1,'nce+Noun+A3sg+P3sg+Loc')]"
REL="[22,1,(MODIFIER)]"> öncesinde </W>
<W IX="12" LEM="" MORPH=" " IG="[(1,'Ankara+Noun+Prop+A3sg+
Pnon+Dat')]" REL="[13,1,(DATIVE.ADJUNCT)]"> Ankara'ya </W>
<W IX="13" LEM="" MORPH=" " IG="[(1,'gel+Verb+Pos')(2,'Adj+
PresPart')]" REL="[17,1,(MODIFIER)]"> gelen </W>
<W IX="14" LEM="" MORPH=" " IG="[(1,'Abd+Noun+Prop+A3sg+Pnon+
Gen')]" REL="[17,1,(POSSESSOR)]"> ABD'nin </W>
<W IX="15" LEM="" MORPH=" " IG="[(1,'iki+Num+Card')]"
REL="[17,1,(MODIFIER)]"> iki </W>
```

```
<W IX="16" LEM="" MORPH=" " IG="[(1,'bakan+Noun+A3sg+Pnon+Nom')]"
REL="[17,1,(CLASSIFIER)]"> bakan </W>
<W IX="17" LEM="" MORPH=" " IG="[(1,'yardmc+Noun+A3sg+P3sg+Nom')]"
REL="[22,1,(SUBJECT)]"> yardımcısı</W>
<W IX="18" LEM="" MORPH=" " IG="[(1,'yirmi+Num+Card')]"
REL="[19,1,(MODIFIER)]"> yirmi </W>
<W IX="19" LEM="" MORPH=" " IG="[(1,'milyar+Num+Card')]"
REL="[20,1,(MODIFIER)]"> milyar </W>
<W IX="20" LEM="" MORPH=" " IG="[(1,'dolar+Noun+A3sg+Pnon+Nom')
(2,'Adj+FitFor')]" REL="[21,1,(MODIFIER)]"> dolarlık </W>
<W IX="21" LEM="" MORPH=" " IG="[(1,'paket+Noun+A3sg+Pnon+Nom')]"
REL="[22,1,(OBJECT)]"> paket </W>
<W IX="22" LEM="" MORPH=" " IG="[(1,'a+Verb+Pos+Fut+A3sg')]"
REL="[23,1,(SENTENCE)]"> açacak </W>
<W IX="23" LEM="" MORPH=" " IG="[(1,'.+Punc')]"
REL="[,( )]"> .  </W>
</S>
.
```

Wordnode–Relationnode.xml

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<SENTENCE>
 <F23 Function="" ID="23" IX="23">
  <F22 Function="SENTENCE" ID="22" IX="22">
   <F11 Function="MODIFIER" ID="11" IX="11">
    <F10 Function="CLASSIFIER" ID="10" IX="10">
     <F9 Function="MODIFIER" ID="9" IX="9">
      <F8 Function="SUBJECT" ID="8" IX="8">
       <F3 Function="MODIFIER" ID="3" IX="3">
        <F1 Function="MODIFIER" ID="1" IX="1">
         <W IX="1" LEM="" MORPH=" " REL="[3,1,(MODIFIER)]">
              Olası</W>
        </F1>
        <F2 Function="CLASSIFIER" ID="2" IX="2">
         <W IX="2" LEM="" MORPH=" " REL="[3,1,(CLASSIFIER)]">
              Irak</W>
        </F2>
```

```
    <W IX="3" LEM="" MORPH=" " REL="[8,1,(MODIFIER)]">
        operasyonunda</W>
   </F3>
   <F6 Function="MODIFIER" ID="6" IX="6">
    <F5 Function="OBJECT" ID="5" IX="5">
     <F4 Function="MODIFIER" ID="4" IX="4">
      <W IX="4" LEM="" MORPH=" " REL="[5,1,(MODIFIER)]">
          Türkiye'den</W>
     </F4>
     <W IX="5" LEM="" MORPH=" " REL="[6,1,(OBJECT)]">
         taleplere</W>
    </F5>
    <W IX="6" LEM="" MORPH=" " REL="[8,1,(MODIFIER)]">
        ilişkin</W>
   </F6>
   <F7 Function="MODIFIER" ID="7" IX="7">
    <W IX="7" LEM="" MORPH=" " REL="[8,1,(MODIFIER)]">
        son</W>
   </F7>
   <W IX="8" LEM="" MORPH=" " REL="[9,2,(SUBJECT)]">
       kararın</W>
  </F8>
  <W IX="9" LEM="" MORPH=" " REL="[10,1,(MODIFIER)]">
      alınacağı</W>
 </F9>
 <W IX="10" LEM="" MORPH=" " REL="[11,1,(CLASSIFIER)]">
     MGK</W>
</F10>
<W IX="11" LEM="" MORPH=" " REL="[22,1,(MODIFIER)]">
    öncesinde</W>
</F11>
<F17 Function="SUBJECT" ID="17" IX="17">
 <F13 Function="MODIFIER" ID="13" IX="13">
  <F12 Function="DATIVE.ADJUNCT" ID="12" IX="12">
   <W IX="12" LEM="" MORPH=" " REL="[13,1,(DATIVE.ADJUNCT)]">
       Ankara'ya</W>
  </F12>
  <W IX="13" LEM="" MORPH=" " REL="[17,1,(MODIFIER)]">
```

```
      gelen</W>
  </F13>
  <F14 Function="POSSESSOR" ID="14" IX="14">
   <W IX="14" LEM="" MORPH=" " REL="[17,1,(POSSESSOR)]">
       ABD'nin</W>
  </F14>
  <F15 Function="MODIFIER" ID="15" IX="15">
   <W IX="15" LEM="" MORPH=" " REL="[17,1,(MODIFIER)]">
       iki</W>
  </F15>
  <F16 Function="CLASSIFIER" ID="16" IX="16">
   <W IX="16" LEM="" MORPH=" " REL="[17,1,(CLASSIFIER)]">
       bakan</W>
  </F16>
  <W IX="17" LEM="" MORPH=" " REL="[22,1,(SUBJECT)]">
      yardımcısı</W>
 </F17>
 <F21 Function="OBJECT" ID="21" IX="21">
  <F20 Function="MODIFIER" ID="20" IX="20">
   <F19 Function="MODIFIER" ID="19" IX="19">
    <F18 Function="MODIFIER" ID="18" IX="18">
     <W IX="18" LEM="" MORPH=" " REL="[19,1,(MODIFIER)]">
         yirmi</W>
    </F18>
    <W IX="19" LEM="" MORPH=" " REL="[20,1,(MODIFIER)]">
        milyar</W>
   </F19>
   <W IX="20" LEM="" MORPH=" " REL="[21,1,(MODIFIER)]">
       dolarlık</W>
  </F20>
  <W IX="21" LEM="" MORPH=" " REL="[22,1,(OBJECT)]">
      paket</W>
 </F21>
 <W IX="22" LEM="" MORPH=" " REL="[23,1,(SENTENCE)]">
     açacak</W>
</F22>
<W IX="23" LEM="" MORPH=" " REL="[,( )]">
     .</W>
```

```
 </F23>
</SENTENCE>
```

3 Phrase-Structure Parsing

3.1 Transformational Grammars

Transformational grammar, or transformational-generative grammar, is a grammar of a natural language, that has been developed in a Chomskian tradition. Noam Chomsky developed the idea that each sentence in a language has two levels of representation: a deep structure and a surface structure.The deep structure represented the core semantic relations of a sentence, and was mapped on to the surface structure via transformations.
Since Turkish is researched by computational linguists very little, already done studies were by regular linguists who followed Chomskian manner. One of these detailed studies was Meskill's Transformational Analysis of Turkish Syntax but it necessiates a conversion to context-free grammar which is admitted to be most appropriate model for computer modelling.
Conversion processes are explained in details in section 3.3

3.2 Definite Clause Grammars and Prolog

Definite Clause Grammar (DCG) is a Prolog preprocessor that takes DCG rules and adds linked difference lists to the goals. DCG provides a syntax for writing more readable grammar parsing rules, without including the linked difference lists.
The syntax of DCG is:

- —>operator indicates a DCG rule, replacing the normal neck (:-) used for Prolog clauses. The preprocessor adds extra arguments for the difference lists. For example, the sentence grammar rule can be written without difference lists using a DCG rule:

  ```
  sentence --> subject, verb, object.
  ```

- Each goal is assumed to refer to the head of a DCG rule, and the preprocessor adds two extra arguments for the difference list.

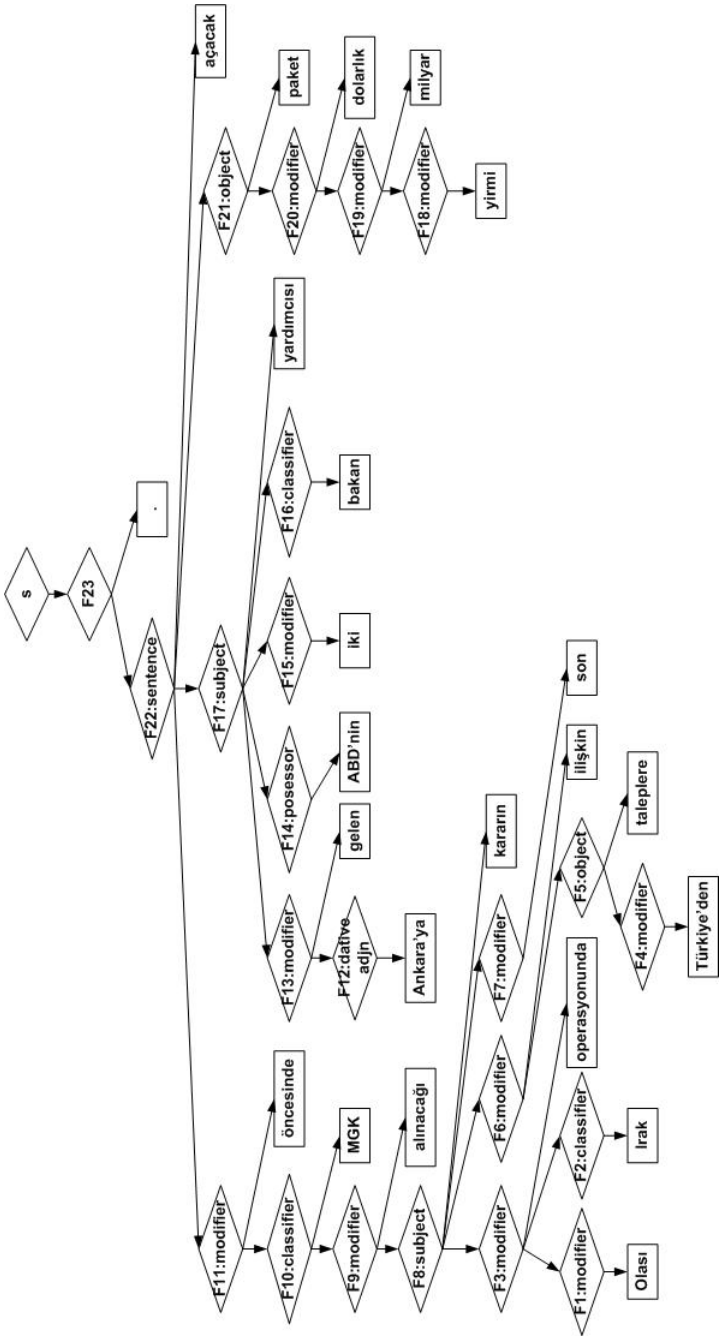

Figure 10: Schema for transformed sentence:"Olası Irak operasyonunda Türkiye'den taleplere ilişkin son kararın alınacağı MGK öncesinde Ankara'ya gelen ABD'nin iki bakan yardımcısı yirmi milyar dolarlık paket açacak."

- Curly braces are used to isolate normal Prolog goals from the DCG preprocessor. For example:

  ```
  subject --> modifier, noun, {write('found subject')}.
  ```

- Square brackets [], list notation, are used to indicate terminal symbols of the grammar. For example:

  ```
  noun --> [cat]
  ```

Results of the DCG preprocessor can be seen using **listing**:

```
?- listing(sentence).
user:sentence(_X1, _X2) :-
    subject(_X1, _X3),
    verb(_X3, _X4),
    object(_X4, _X2).
yes
```

Valid lists are generated using the root rule:

```
?- sentence(X,[]).
X = [the, cat, chases, the, cat] ;
X = [the, cat, chases, the, mouse] ;
...
```

3.3 Conversion from Tranformational Grammar to Context Free Grammar

As it is mentioned in 3.1 tranformational grammars are not so suitable for computational processing and this chapter explains how conversion to a context-free grammar is carried out.
Rules for PSG parser is compiled from Meskill's study *(Transformational rules used in this project is listed in Appendix B)*. Below are the conversion sessions observed to generate the whole set of context-free grammar that will be used for definite clause grammar implementation in Prolog. Conversion sessions are ordered from lower (simple-structures) to higher (complex structures e.g. verb phrases or sentences) levels. Each session is titled as a group and at each group there are some actions taken that can be expressed as below in summary:

- ***using*** indicates that a rule, which has the number written next to it, from Meskill's study is used.
- ***generated*** indicates that a rule is generated using the rules or lately generated rules.
- ***setting*** indicates that a rule is named as a different non-terminal.
- ***roman numerals*** are used for later referencing.
- ***expansion*** indicates that lately addressed rule is expanded and all the possible generations are addressed.

Complete listing of all the rules generated after the following sessions are in Appendix C. Meskill's thesis is the major source for this study. Terms N_{mobile}, $N_{animate}$, $n_{inanimate}$, $n_{animate\ transport}$, $n_{animate}$, $n_{animate\ human}$, $n_{animate\ proper}$, $n_{animate\ human\ proper}$, $N_{immmobile}$, Nom, n_{item}, n_{locale}, $n_{locale\ proper}$, n_{number}, $n_{substance}$, $Nominal$, N_{time}, $n_{time\ a}$, $n_{time\ b}$, $n_{time\ c}$, $verb_{impersonal}$, $Verb$, $n_{abstract}$, $posposition_a$, $posposition_b$, $posposition_c$, $posposition_d$, $posposition_e$, $particle_a$, $particle_b$, adj_a, adj_b, adj_c, adj_d, $Verb_{mobile}$, $Verb_{mobile\ a}$, $Verb_{mobile\ b}$, $Adjunct$, $Verb_{transitive}$, $Verb_{transitive\ human}$, $verb_{transitive\ mobile\ 1}$, $verb_{transitive\ human\ mobile\ 1}$, $verb_{transitive\ mobile\ 2}$, $verb_{transitive\ human\ mobile\ 2}$, $Aspect$, $copula$ are used in the same context as in source. For more information about these you may refer to the thesis[8]. When a term like $N_{animate}$ is encountered, script indicates syntactic category or parts of speech and subscript indicates its feature. In such indications, scripts printed in lower case directly generates words from lexicon and &,/ mark word and phrase boundaries respectively.

Group 1:This group is composed of NP rules that generates noun forms from NP's.

using 27: $N_{mobile} \rightarrow \left\{ \begin{array}{l} N_{animate} \\ n_{inanimate} \end{array} \right\}$

using 28: $N_{animate} \rightarrow \left\{ \begin{array}{l} n_{\text{ animate transport}} \\ n_{\text{ animate (proper) (human)}} \end{array} \right\}$

$$\text{generated: } N_{mobile} \rightarrow \left\{ \begin{array}{l} \left\{ \begin{array}{l} n \text{ animate transport} \\ n \text{ animate (proper) (human)} \end{array} \right\} \\ n_{inanimate} \end{array} \right\} \text{(I)}$$

$$\text{using 25:Nom} \rightarrow \left\{ \begin{array}{l} N_{immobile} \\ N_{mobile} \end{array} \right\}$$

$$\text{using 26:} N_{immobile} \rightarrow \left\{ \begin{array}{l} n_{item} \\ n_{locale(proper)} \\ n_{number} \\ n_{substance} \end{array} \right\}$$

using (I)

$$\text{generated:Nom} \rightarrow \left\{ \begin{array}{l} \left\{ \begin{array}{l} n_{item} \\ n_{locale(proper)} \\ n_{number} \\ n_{substance} \end{array} \right\} \\ \left\{ \begin{array}{l} \left\{ \begin{array}{l} n \text{ animate transport} \\ n \text{ animate (proper) (human)} \end{array} \right\} \\ n_{inanimate} \end{array} \right\} \end{array} \right\} \text{(II)}$$

$$\text{using 24: Nominal} \rightarrow \left\{ \begin{array}{l} (demonstarative\&) \left\{ \begin{array}{l} (n_{number}\&)Nom \\ n_{abstract} \end{array} \right\} \\ N_{time} \end{array} \right\}$$

$$\text{using 29:} N_{time} \rightarrow \left\{ \begin{array}{l} (demonstrative\&) \left\{ \begin{array}{l} (n_{number}\&)n \text{ time a} \\ (saat)n \text{ time b} \end{array} \right\} \\ n \text{ time c} \end{array} \right\}$$

using (II)
generated:

$$\text{Nominal}\rightarrow \left\{ \begin{array}{l} (demonstarative\&) \left\{ \begin{array}{l} (n_{number}\&) \left\{ \begin{array}{l} \left\{ \begin{array}{l} n_{item} \\ n_{locale(proper)} \\ n_{number} \\ n_{substance} \end{array} \right\} \\ \left\{ \left\{ \begin{array}{l} n \text{ animate transport} \\ n \text{ animate (proper) (human)} \end{array} \right\} \right\} \\ n_{inanimate} \end{array} \right\} \\ n_{abstract} \end{array} \right\} \\ \left\{ \left\{ \begin{array}{l} n \text{ time a} \\ n \text{ time b} \end{array} \right\} \right\} \end{array} \right\} \text{(III)}$$

Note:$n_{time\,c}$,(demonstrative&),(n_{number}&),(saat) are excluded to avoid comlexity

$$\text{using 23:NP} \rightarrow \left\{ \begin{array}{l} Nominal \\ pronoun \\ bura \\ şura \\ ora \\ nere \end{array} \right\}$$

using (III)
generated:

$$\text{NP} \rightarrow \left\{ \begin{array}{l} \left\{ \begin{array}{l} (demonstarative\&) \left\{ \begin{array}{l} (n_{number}\&) \left\{ \begin{array}{l} \left\{ \begin{array}{l} n_{item} \\ n_{locale(proper)} \\ n_{number} \\ n_{substance} \end{array} \right\} \\ \left\{ \left\{ \begin{array}{l} n \text{ animate transport} \\ n \text{ animate (proper) (human)} \end{array} \right\} \right\} \\ n_{inanimate} \end{array} \right\} \\ n_{abstract} \end{array} \right\} \\ \left\{ \left\{ \begin{array}{l} n \text{ time a} \\ n \text{ time b} \end{array} \right\} \right\} \end{array} \right\} \\ pronoun \\ bura \\ şura \\ ora \\ nere \end{array} \right\} \text{(IV)}$$

expansion of (IV):

- $NP \rightarrow pronoun$
- $NP \rightarrow bura$
- $NP \rightarrow şura$
- $NP \rightarrow ora$
- $NP \rightarrow nere$
- $NP \rightarrow n_{time\ a}$
- $NP \rightarrow n_{time\ b}$
- $NP \rightarrow n_{abstract}$
- $NP \rightarrow demonstrative\ \&\ n_{abstract}$
- $NP \rightarrow n_{animate}$
- $NP \rightarrow n_{animate\ human}$
- $NP \rightarrow n_{animate\ proper}$
- $NP \rightarrow n_{animate\ proper\ human}$
- $NP \rightarrow n_{animate\ transport}$
- $NP \rightarrow n_{inanimate}$
- $NP \rightarrow n_{item}$
- $NP \rightarrow n_{locale}$
- $NP \rightarrow n_{locale\ proper}$
- $NP \rightarrow n_{number}$
- $NP \rightarrow n_{substance}$
- $NP \rightarrow n_{number}\ \&\ n_{animate}$
- $NP \rightarrow n_{number}\ \&\ n_{animate\ human}$
- $NP \rightarrow n_{number}\ \&\ n_{animate\ proper}$
- $NP \rightarrow n_{number}\ \&\ n_{animate\ proper\ human}$
- $NP \rightarrow n_{number}\ \&\ n_{animate\ transport}$

- $NP \rightarrow n_{number}$ & $n_{inanimate}$
- $NP \rightarrow n_{number}$ & n_{item}
- $NP \rightarrow n_{number}$ & n_{locale}
- $NP \rightarrow n_{number}$ & $n_{locale\ proper}$
- $NP \rightarrow n_{number}$ & n_{number}
- $NP \rightarrow n_{number}$ & $n_{substance}$
- $NP \rightarrow demonstrative$ & $n_{animate}$
- $NP \rightarrow demonstrative$ & $n_{animate\ human}$
- $NP \rightarrow demonstrative$ & $n_{animate\ proper}$
- $NP \rightarrow demonstrative$ & $n_{animate\ proper\ human}$
- $NP \rightarrow demonstrative$ & $n_{animate\ transport}$
- $NP \rightarrow demonstrative$ & $n_{inanimate}$
- $NP \rightarrow demonstrative$ & n_{item}
- $NP \rightarrow demonstrative$ & n_{locale}
- $NP \rightarrow demonstrative$ & $n_{locale\ proper}$
- $NP \rightarrow demonstrative$ & n_{number}
- $NP \rightarrow demonstrative$ & $n_{substance}$
- $NP \rightarrow demonstrative$ & n_{number} & $n_{animate}$
- $NP \rightarrow demonstrative$ & n_{number} & $n_{animate\ human}$
- $NP \rightarrow demonstrative$ & n_{number} & $n_{animate\ proper}$
- $NP \rightarrow demonstrative$ & n_{number} & $n_{animate\ proper\ human}$
- $NP \rightarrow demonstrative$ & n_{number} & $n_{animate\ transport}$
- $NP \rightarrow demonstrative$ & n_{number} & $n_{inanimate}$
- $NP \rightarrow demonstrative$ & n_{number} & n_{item}
- $NP \rightarrow demonstrative$ & n_{number} & n_{locale}

- $NP \rightarrow demonstrative \,\&\, n_{number} \,\&\, n_{locale\,proper}$
- $NP \rightarrow demonstrative \,\&\, n_{number} \,\&\, n_{number}$
- $NP \rightarrow demonstrative \,\&\, n_{number} \,\&\, n_{substance}$

Group 2: This group is composed of instrumental noun phrases which is a special type of regular NP whose usage is restricted in some contexts only with instrumental case –le.

using 18: NP+ le $\rightarrow (demostrative\&)(n_{number}) \left\{ \begin{array}{l} N_{immobile} \\ n_{inanimate} \\ n_{\text{animate transport}} \end{array} \right\} le$

using 26: $N_{immobile} \rightarrow \left\{ \begin{array}{l} n_{item} \\ n_{locale(proper)} \\ n_{number} \\ n_{substance} \end{array} \right\}$

generated: $NP_{ins} \rightarrow (demostrative\&)(n_{number}) \left\{ \begin{array}{l} \left\{ \begin{array}{l} n_{item} \\ n_{locale(proper)} \\ n_{number} \\ n_{substance} \end{array} \right\} \\ n_{inanimate} \\ n_{\text{animate transport}} \end{array} \right\}$(I)

expansion of (I):

- $NP_{ins} \rightarrow n_{item}$
- $NP_{ins} \rightarrow n_{local}$
- $NP_{ins} \rightarrow n_{local\,proper}$
- $NP_{ins} \rightarrow n_{number}$
- $NP_{ins} \rightarrow n_{substance}$
- $NP_{ins} \rightarrow n_{inanimate}$
- $NP_{ins} \rightarrow n_{animate\,transport}$
- $NP_{ins} \rightarrow n_{number} \,\&\, n_{item}$
- $NP_{ins} \rightarrow n_{number} \,\&\, n_{local}$
- $NP_{ins} \rightarrow n_{number} \,\&\, n_{local\,proper}$

- $NP_{ins} \rightarrow n_{number}$ & n_{number}
- $NP_{ins} \rightarrow n_{number}$ & $n_{substance}$
- $NP_{ins} \rightarrow n_{number}$ & $n_{inanimate}$
- $NP_{ins} \rightarrow n_{number}$ & $n_{animate\,transport}$
- $NP_{ins} \rightarrow demonstrative$ & n_{item}
- $NP_{ins} \rightarrow demonstrative$ & n_{local}
- $NP_{ins} \rightarrow demonstrative$ & $n_{local\,proper}$
- $NP_{ins} \rightarrow demonstrative$ & n_{number}
- $NP_{ins} \rightarrow demonstrative$ & $n_{substance}$
- $NP_{ins} \rightarrow demonstrative$ & $n_{inanimate}$
- $NP_{ins} \rightarrow demonstrative$ & $n_{animate\,transport}$
- $NP_{ins} \rightarrow demonstrative$ & n_{number} & n_{item}
- $NP_{ins} \rightarrow demonstrative$ & n_{number} & n_{local}
- $NP_{ins} \rightarrow demonstrative$ & n_{number} & $n_{local\,proper}$
- $NP_{ins} \rightarrow demonstrative$ & n_{number} & n_{number}
- $NP_{ins} \rightarrow demonstrative$ & n_{number} & $n_{substance}$
- $NP_{ins} \rightarrow demonstrative$ & n_{number} & $n_{inanimate}$
- $NP_{ins} \rightarrow demonstrative$ & n_{number} & $n_{animate\,transport}$

Group 3: These rules are generated to reduce complexity in generation of NP rules and time-related noun pharases,abstract noun forms are generated from this group.

using 17: NP + le + X $\begin{bmatrix} verb_{impersonal} \\ Verb \end{bmatrix} \rightarrow$ (demonstrative&) $n_{abstract} + le +$ $X \begin{bmatrix} verb_{impersonal} \\ Verb \end{bmatrix}$

setting: $NP_{abstract} \rightarrow (demonstrative\&)n_{abstract}$

expansion of setting:

- $NP_{abstract} \rightarrow n_{abstract}$
- $NP_{abstract} \rightarrow demonstrative\,\&\,n_{abstract}$

using 29: $N_{time} \rightarrow \left\{ \begin{array}{l} (demonstrative\&) \left\{ \begin{array}{l} (n_{number}\&)n \text{ time a} \\ (saat)n \text{ time b} \end{array} \right\} \\ n \text{ time c} \end{array} \right\}$

setting: $NP_{time} \rightarrow \left\{ \left\{ \begin{array}{l} n \text{ time a} \\ n \text{ time b} \end{array} \right\} \right\}$

Note:$n_{time\,c}$,(demonstrative&),(n_{number}&),(saat) are excluded to avoid comlexity expansion of setting:

- $NP_{time} \rightarrow n_{time\,a}$
- $NP_{time} \rightarrow n_{time\,b}$

Group 4: These rules are generated to reduce complexity in generation of adverb rules.

- $PostPosPhrase_a \rightarrow postposition_a$
- $PostPosPhrase_a \rightarrow NP + den\,\&\,postposition_a$
- $PostPosPhrase_b \rightarrow NP + den\,\&\,postposition_b$
- $PostPosPhrase_c \rightarrow NP + den\,\&\,postposition_c$
- $PostPosPhrase_d \rightarrow postposition_d$
- $PostPosPhrase_d \rightarrow NP + e\,\&\,postposition_d$
- $PostPosPhrase_e \rightarrow NP + e\,\&\,postposition_e$
- $PostPosPhrase_f \rightarrow NP\,\&\,postposition_f$

Group 5: This group generates adverbial phrases in postpositional froms.

using 15: Adv $\rightarrow (particle \left\{ \begin{array}{l} a \\ b \end{array} \right\} /)(\left\{ \begin{array}{l} (NP + den\&)posposition_a \\ NP + den\&posposition \left\{ \begin{array}{l} b \\ c \end{array} \right\} \end{array} \right\} /)$

$(\left\{ \begin{array}{l} (NP + e\&)postposition_d \\ NP + e\&postposition_e \end{array} \right\} /)(NP\&postposition_f)$

using Group 4

generated:Adv $\rightarrow$ ($particle \left\{ \begin{array}{c} a \\ b \end{array} \right\}$ /)($\left\{ \begin{array}{l} PostPosPhrase_a \\ PostPosPhrase_b \\ PostPosPhrase_c \end{array} \right\}$ /)
($\left\{ \begin{array}{l} PostPosPhrase_d \\ PostPosPhrase_e \end{array} \right\}$ /)($PostPosPhrase_f$)(I)

expansion of (I):

- $Adv \rightarrow PostPosPhrase_f$
- $Adv \rightarrow PostPosPhrase_d$
- $Adv \rightarrow PostPosPhrase_e$
- $Adv \rightarrow PostPosPhrase_d \ / \ PostPosPhrase_f$
- $Adv \rightarrow PostPosPhrase_e \ / \ PostPosPhrase_f$
- $Adv \rightarrow PostPosPhrase_a$
- $Adv \rightarrow PostPosPhrase_a \ / \ PostPosPhrase_f$
- $Adv \rightarrow PostPosPhrase_a \ / \ PostPosPhrase_d$
- $Adv \rightarrow PostPosPhrase_a \ / \ PostPosPhrase_e$
- $Adv \rightarrow PostPosPhrase_a \ / \ PostPosPhrase_d \ / \ PostPosPhrase_f$
- $Adv \rightarrow PostPosPhrase_a \ / \ PostPosPhrase_e \ / \ PostPosPhrase_f$
- $Adv \rightarrow PostPosPhrase_b$
- $Adv \rightarrow PostPosPhrase_b \ / \ PostPosPhrase_f$
- $Adv \rightarrow PostPosPhrase_b \ / \ PostPosPhrase_d$
- $Adv \rightarrow PostPosPhrase_b \ / \ PostPosPhrase_e$
- $Adv \rightarrow PostPosPhrase_b \ / \ PostPosPhrase_d \ / \ PostPosPhrase_f$
- $Adv \rightarrow PostPosPhrase_b \ / \ PostPosPhrase_e \ / \ PostPosPhrase_f$
- $Adv \rightarrow PostPosPhrase_c$
- $Adv \rightarrow PostPosPhrase_c \ / \ PostPosPhrase_f$
- $Adv \rightarrow PostPosPhrase_c \ / \ PostPosPhrase_d$

- $Adv \rightarrow PostPosPhrase_c \ / \ PostPosPhrase_e$
- $Adv \rightarrow PostPosPhrase_c \ / \ PostPosPhrase_d \ / \ PostPosPhrase_f$
- $Adv \rightarrow PostPosPhrase_c \ / \ PostPosPhrase_e \ / \ PostPosPhrase_f$
- $Adv \rightarrow particle_a$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_f$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_d$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_e$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_d \ / \ PostPosPhrase_f$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_e \ / \ PostPosPhrase_f$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_a$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_a \ / \ PostPosPhrase_f$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_a \ / \ PostPosPhrase_d$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_a \ / \ PostPosPhrase_e$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_a \ / \ PostPosPhrase_d \ / \ PostPosPhrase_f$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_a \ / \ PostPosPhrase_e \ / \ PostPosPhrase_f$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_b$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_b \ / \ PostPosPhrase_f$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_b \ / \ PostPosPhrase_d$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_b \ / \ PostPosPhrase_e$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_b \ / \ PostPosPhrase_d \ / \ PostPosPhrase_f$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_b \ / \ PostPosPhrase_e \ / \ PostPosPhrase_f$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_c$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_c \ / \ PostPosPhrase_f$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_c \ / \ PostPosPhrase_d$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_a \ / \ PostPosPhrase_e$

- $Adv \rightarrow particle_a$ / $PostPosPhrase_c$ / $PostPosPhrase_d$ / $PostPosPhrase_f$
- $Adv \rightarrow particle_a$ / $PostPosPhrase_c$ / $PostPosPhrase_e$ / $PostPosPhrase_f$
- $Adv \rightarrow particle_b$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_f$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_d$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_e$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_d$ / $PostPosPhrase_f$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_e$ / $PostPosPhrase_f$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_a$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_a$ / $PostPosPhrase_f$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_a$ / $PostPosPhrase_d$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_a$ / $PostPosPhrase_e$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_a$ / $PostPosPhrase_d$ / $PostPosPhrase_f$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_a$ / $PostPosPhrase_e$ / $PostPosPhrase_f$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_b$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_b$ / $PostPosPhrase_f$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_b$ / $PostPosPhrase_d$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_b$ / $PostPosPhrase_e$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_b$ / $PostPosPhrase_d$ / $PostPosPhrase_f$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_b$ / $PostPosPhrase_e$ / $PostPosPhrase_f$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_c$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_c$ / $PostPosPhrase_f$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_c$ / $PostPosPhrase_d$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_c$ / $PostPosPhrase_e$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_c$ / $PostPosPhrase_d$ / $PostPosPhrase_f$

- $Adv \rightarrow particle_b$ / $PostPosPhrase_c$ / $PostPosPhrase_e$ / $PostPosPhrase_f$

Group 6: This group generates NP's as main phrases.
using 10: $Adjunct + NP + copula \rightarrow (NP + de/)(NP + e/)NP + copula$
generated: $AdjunctiveNP \rightarrow (NP + de/)(NP + e/)NP + copula$(I)
expansion of (I):

- $AdjunctiveNP \rightarrow NP + copula$
- $AdjunctiveNP \rightarrow NP + e \ / \ NP + copula$
- $AdjunctiveNP \rightarrow NP + de \ / \ NP + copula$
- $AdjunctiveNP \rightarrow NP + de \ / \ NP + e \ / \ NP + copula$

Group 7: This group generates adjectives as main phrases.
using 10: $Adjunct + adj_a + copula \rightarrow (NP + de/)(NP + e/)adj_a + copula$
generated: $AdjunctiveAdj_a \rightarrow (NP + de/)(NP + e/)adj_a + copula$(I)
expansion of (I):

- $AdjunctiveAdj_a \rightarrow adj_a + copula$
- $AdjunctiveAdj_a \rightarrow NP + e \ / \ adj_a + copula$
- $AdjunctiveAdj_a \rightarrow NP + de \ / \ adj_a + copula$
- $AdjunctiveAdj_a \rightarrow NP + de \ / \ NP + e \ / \ adj_a + copula$

Group 8: This group generates adjectives as main phrases.
using 10: $Adjunct + adj_b + copula \rightarrow (NP + de/)(NP + e/)adj_b + copula$
generated: $AdjunctiveAdj_b \rightarrow (NP + de/)(NP + e/)adj_b + copula$(I)
expansion of (I):

- $AdjunctiveAdj_b \rightarrow adj_b + copula$
- $AdjunctiveAdj_b \rightarrow NP + e \ / \ adj_b + copula$
- $AdjunctiveAdj_b \rightarrow NP + de \ / \ adj_b + copula$

- $AdjunctiveAdj_b \rightarrow NP + de \ / \ NP + e \ / \ adj_b + copula$

Group 9: This group generates adjectives as main phrases.
using 10: $Adjunct + adj_c + copula \rightarrow (NP + de/)(NP + \left\{ \begin{array}{l} e \\ den \end{array} \right\} /)adj_c + copula$
generated:$AdjunctiveAdj_c \rightarrow (NP + de/)(NP + \left\{ \begin{array}{l} e \\ den \end{array} \right\} /)adj_c + copula$(I)
expansion of (I):

- $AdjunctiveAdj_c \rightarrow adj_c + copula$
- $AdjunctiveAdj_c \rightarrow NP + e \ / \ adj_c + copula$
- $AdjunctiveAdj_c \rightarrow NP + de \ / \ adj_c + copula$
- $AdjunctiveAdj_c \rightarrow NP + de \ / \ NP + e \ / \ adj_c + copula$
- $AdjunctiveAdj_c \rightarrow NP + den \ / \ adj_c + copula$
- $AdjunctiveAdj_c \rightarrow NP + de \ / \ NP + den \ / \ adj_c + copula$

Group 10: This group generates adjectives as main phrases.
using 10:$Adjunct + adj_d + copula \rightarrow (NP + de/)(NP + le/)adj_d + copula$
generated:$AdjunctiveAdj_d \rightarrow (NP + de/)(NP + le/)adj_d + copula$(I)
using Group 2
generated:$AdjunctiveAdj_d \rightarrow (NP + de/)(NP_{ins} + le/)adj_d + copula$(II)
expansion of (II):

- $AdjunctiveAdj_d \rightarrow adj_d + copula$
- $AdjunctiveAdj_d \rightarrow NP_{ins} + le \ / \ adj_d + copula$
- $AdjunctiveAdj_d \rightarrow NP + de \ / \ adj_d + copula$
- $AdjunctiveAdj_d \rightarrow NP + de \ / \ NP_{ins} + le \ / \ adj_d + copula$

Group 11: This group includes rules that generates VP's where verb is classified as impersonal.

using 17: NP + le + X $\begin{bmatrix} verb_{impersonal} \\ Verb \end{bmatrix} \rightarrow$ (demonstrative&) $n_{abstract} + le +$
$X \begin{bmatrix} verb_{impersonal} \\ Verb \end{bmatrix}$
generated: NP + le + X $\begin{bmatrix} verb_{impersonal} \\ Verb \end{bmatrix} \rightarrow NP_{abstract}+le+X \begin{bmatrix} verb_{impersonal} \\ Verb \end{bmatrix}$(I)
using 11: $Adjunct + verb_{impersonal} \rightarrow (NP + de/)(NP + le/)(NP + den/)$
$(NP + e/)verb_{impersonal}$
using (I)
generated: $Adjunct + verb_{impersonal} \rightarrow (NP + de/)(NP_{abstract} + le/)(NP + den/)$
$(NP + e/)verb_{impersonal}$(II)
using (II)
generated: $VP_{impersonal} \rightarrow (NP + de/)(NP_{abstract} + le/)(NP + den/)$
$(NP + e/)verb_{impersonal}$(III)
expansion of (III):

- $VP_{impersonal} \rightarrow verb_{impersonal}$
- $VP_{impersonal} \rightarrow NP + e \ / \ verb_{impersonal}$
- $VP_{impersonal} \rightarrow NP + den \ / \ verb_{impersonal}$
- $VP_{impersonal} \rightarrow NP + den \ / \ NP + e \ / \ verb_{impersonal}$
- $VP_{impersonal} \rightarrow NP_{abstract} + le \ / \ verb_{impersonal}$
- $VP_{impersonal} \rightarrow NP_{abstract} + le \ / \ NP + e \ / \ verb_{impersonal}$
- $VP_{impersonal} \rightarrow NP_{abstract} + le \ / \ NP + den \ / \ verb_{impersonal}$
- $VP_{impersonal} \rightarrow NP_{abstract} + le \ / \ NP + den \ / \ NP + e \ / \ verb_{impersonal}$
- $VP_{impersonal} \rightarrow NP + de \ / \ verb_{impersonal}$
- $VP_{impersonal} \rightarrow NP + de \ / \ NP + e \ / \ verb_{impersonal}$
- $VP_{impersonal} \rightarrow NP + de \ / \ NP + den \ / \ verb_{impersonal}$
- $VP_{impersonal} \rightarrow NP + de \ / \ NP + den \ / \ NP + e \ / \ verb_{impersonal}$
- $VP_{impersonal} \rightarrow NP + de \ / \ NP_{abstract} + le \ / \ verb_{impersonal}$
- $VP_{impersonal} \rightarrow NP + de \ / \ NP_{abstract} + le \ / \ NP + e \ / \ verb_{impersonal}$
- $VP_{impersonal} \rightarrow NP + de \ / \ NP_{abstract} + le \ / \ NP + den \ / \ verb_{impersonal}$

- $VP_{impersonal} \rightarrow NP{+}de \,/\, NP_{abstract}{+}le \,/\, NP{+}den \,/\, NP{+}e \,/\, verb_{impersonal}$

Group 12: This group includes rules that generates VP's where verb is classified as mobile, mobile a, mobile b.
using 11:
$Adjunct + verb_{mobile(\begin{bmatrix} a \\ b \end{bmatrix})} \rightarrow$ (NP+de/)(NP+le/)(NP+den/)
(NP+e/)$verb_{mobile(\begin{bmatrix} a \\ b \end{bmatrix})}$
using Group 2
generated: $Adjunct + verb_{mobile(\begin{bmatrix} a \\ b \end{bmatrix})} \rightarrow (NP + de/)(NP_{ins} + le/)(NP + den/)$
$(NP + e/)verb_{mobile(\begin{bmatrix} a \\ b \end{bmatrix})}$ (I)
setting: $VP_{mobile} \rightarrow Adjunct + verb_{mobile}$
setting: $VP_{mobile\,a} \rightarrow Adjunct + verb_{mobile\,a}$
setting: $VP_{mobile\,b} \rightarrow Adjunct + verb_{mobile\,b}$
expansion of (I) according to settings:

- $VP_{mobile} \rightarrow verb_{mobile}$
- $VP_{mobile} \rightarrow NP + e \;/\; verb_{mobile}$
- $VP_{mobile} \rightarrow NP + den \;/\; verb_{mobile}$
- $VP_{mobile} \rightarrow NP + den \;/\; NP + e \;/\; verb_{mobile}$
- $VP_{mobile} \rightarrow NP_{ins} + le \;/\; verb_{mobile}$
- $VP_{mobile} \rightarrow NP_{ins} + le \;/\; NP + e \;/\; verb_{mobile}$
- $VP_{mobile} \rightarrow NP_{ins} + le \;/\; NP + den \;/\; verb_{mobile}$
- $VP_{mobile} \rightarrow NP_{ins} + le \;/\; NP + den \;/\; NP + e \;/\; verb_{mobile}$
- $VP_{mobile} \rightarrow NP_{time} + de \;/\; verb_{mobile}$
- $VP_{mobile} \rightarrow NP_{time} + de \;/\; NP + e \;/\; verb_{mobile}$
- $VP_{mobile} \rightarrow NP_{time} + de \;/\; NP + den \;/\; verb_{mobile}$

- $VP_{mobile} \rightarrow NP_{time} + de \ / \ NP + den \ / \ NP + e \ / \ verb_{mobile}$
- $VP_{mobile} \rightarrow NP_{time} + de \ / \ NP_{ins} + le \ / \ verb_{mobile}$
- $VP_{mobile} \rightarrow NP_{time} + de \ / \ NP_{ins} + le \ / \ NP + e \ / \ verb_{mobile}$
- $VP_{mobile} \rightarrow NP_{time} + de \ / \ NP_{ins} + le \ / \ NP + den \ / \ verb_{mobile}$
- $VP_{mobile} \rightarrow NP_{time} + de \ / \ NP_{ins} + le \ / \ NP + den \ / \ NP + e \ / \ verb_{mobile}$
- $VP_{mobile\ a} \rightarrow verb_{mobile\ a}$
- $VP_{mobile\ a} \rightarrow NP + e \ / \ verb_{mobile\ a}$
- $VP_{mobile\ a} \rightarrow NP + den \ / \ verb_{mobile\ a}$
- $VP_{mobile\ a} \rightarrow NP + den \ / \ NP + e \ / \ verb_{mobile\ a}$
- $VP_{mobile\ a} \rightarrow NP_{ins} + le \ / \ verb_{mobile\ a} + Aspect$
- $VP_{mobile\ a} \rightarrow NP_{ins} + le \ / \ NP + e \ / \ verb_{mobile\ a}$
- $VP_{mobile\ a} \rightarrow NP_{ins} + le \ / \ NP + den \ / \ verb_{mobile\ a}$
- $VP_{mobile\ a} \rightarrow NP_{ins} + le \ / \ NP + den \ / \ NP + e \ / \ verb_{mobile\ a}$
- $VP_{mobile\ a} \rightarrow NP_{time} + de \ / \ verb_{mobile\ a}$
- $VP_{mobile\ a} \rightarrow NP_{time} + de \ / \ NP + e \ / \ verb_{mobile\ a}$
- $VP_{mobile\ a} \rightarrow NP_{time} + de \ / \ NP + den \ / \ verb_{mobile\ a}$
- $VP_{mobile\ a} \rightarrow NP_{time} + de \ / \ NP + den \ / \ NP + e \ / \ verb_{mobile\ a}$
- $VP_{mobile\ a} \rightarrow NP_{time} + de \ / \ NP_{ins} + le \ / \ verb_{mobile\ a}$
- $VP_{mobile\ a} \rightarrow NP_{time} + de \ / \ NP_{ins} + le \ / \ NP + e \ / \ verb_{mobile\ a}$
- $VP_{mobile\ a} \rightarrow NP_{time} + de \ / \ NP_{ins} + le \ / \ NP + den \ / \ verb_{mobile\ a}$
- $VP_{mobile\ a} \rightarrow NP_{time} + de \ / \ NP_{ins} + le \ / \ NP + den \ / \ NP + e \ / \ verb_{mobile\ a}$
- $VP_{mobile\ b} \rightarrow verb_{mobile\ b}$
- $VP_{mobile\ b} \rightarrow NP + e \ / \ verb_{mobile\ b}$
- $VP_{mobile\ b} \rightarrow NP + den \ / \ verb_{mobile\ b}$
- $VP_{mobile\ b} \rightarrow NP + den \ / \ NP + e \ / \ verb_{mobile\ b}$

- $VP_{mobile\,b} \rightarrow NP_{ins} + le \ / \ verb_{mobile\,b}$
- $VP_{mobile\,b} \rightarrow NP_{ins} + le \ / \ NP + e \ / \ verb_{mobile\,b}$
- $VP_{mobile\,b} \rightarrow NP_{ins} + le \ / \ NP + den \ / \ verb_{mobile\,b}$
- $VP_{mobile\,b} \rightarrow NP_{ins} + le \ / \ NP + den \ / \ NP + e \ / \ verb_{mobile\,b}$
- $VP_{mobile\,b} \rightarrow NP_{time} + de \ / \ verb_{mobile\,b}$
- $VP_{mobile\,b} \rightarrow NP_{time} + de \ / \ NP + e \ / \ verb_{mobile\,b}$
- $VP_{mobile\,b} \rightarrow NP_{time} + de \ / \ NP + den \ / \ verb_{mobile\,b}$
- $VP_{mobile\,b} \rightarrow NP_{time} + de \ / \ NP + den \ / \ NP + e \ / \ verb_{mobile\,b}$
- $VP_{mobile\,b} \rightarrow NP_{time} + de \ / \ NP_{ins} + le \ / \ verb_{mobile\,b}$
- $VP_{mobile\,b} \rightarrow NP_{time} + de \ / \ NP_{ins} + le \ / \ NP + e \ / \ verb_{mobile\,b}$
- $VP_{mobile\,b} \rightarrow NP_{time} + de \ / \ NP_{ins} + le \ / \ NP + den \ / \ verb_{mobile\,b}$
- $VP_{mobile\,b} \rightarrow NP_{time} + de \ / \ NP_{ins} + le \ / \ NP + den \ / \ NP + e \ / \ verb_{mobile\,b}$

Group 13: This group includes rules that generates VP's where verb is classified as only Verb.
using 12: Adjunct + Verb → (NP+de/)(NP+le/)(NP+den) Verb
using 17: NP + le + X $\begin{bmatrix} verb_{impersonal} \\ Verb \end{bmatrix}$ → (demonstrative&) $n_{abstract} + le +$ $X \begin{bmatrix} verb_{impersonal} \\ Verb \end{bmatrix}$
generated: Adjunct + Verb → (NP+de/)($NP_{abstract} + le$/)(NP+den) Verb (I)
setting: $VP_{non} \rightarrow Adjunct + Verb$
generated: $VP_{non} \rightarrow (NP + de/)(NP_{abstract} + le/)(NP + den)Verb$ (II)
expansion of (II):

- $VP_{non} \rightarrow Verb$
- $VP_{non} \rightarrow NP + den \ / \ Verb$
- $VP_{non} \rightarrow NP_{abstract} + le \ / \ Verb$
- $VP_{non} \rightarrow NP_{abstract} + le \ / \ NP + den \ / \ Verb$

- $VP_{non} \rightarrow NP + de \ / \ Verb$
- $VP_{non} \rightarrow NP + de \ / \ NP + den \ / \ Verb$
- $VP_{non} \rightarrow NP + de \ / \ NP_{abstract} + le \ / \ Verb$
- $VP_{non} \rightarrow NP + de \ / \ NP_{abstract} + le \ / \ NP + den \ / \ Verb$

Group 14: This group includes rules that generates VP's where verb is classified as transitive and transitive human.
using 13: $Adjunct + Verb_{transitive(human)} \rightarrow$
$(NP + de/)(NP + le/)(NP + den/)(NP + i)Verb_{transitive(human)}$
using Group 2
generated: $Adjunct + Verb_{transitive(human)} \rightarrow$
$(NP + de/)(NP_{ins} + le/)(NP + den/)(NP + i)Verb_{transitive(human)}$(I)
setting: $VP_{transitive} \rightarrow Adjunct + Verb_{transitive}$
setting: $VP_{transitive\,human} \rightarrow Adjunct + Verb_{transitive\,human}$
expansion of (I) and settings:

- $VP_{transitive} \rightarrow Verb_{transitive}$
- $VP_{transitive} \rightarrow NP + i \ / \ Verb_{transitive}$
- $VP_{transitive} \rightarrow NP + den \ / \ Verb_{transitive}$
- $VP_{transitive} \rightarrow NP + den \ / \ NP + i \ / \ Verb_{transitive}$
- $VP_{transitive} \rightarrow NP_{ins} + le \ / \ Verb_{transitive}$
- $VP_{transitive} \rightarrow NP_{ins} + le \ / \ NP + i \ / \ Verb_{transitive}$
- $VP_{transitive} \rightarrow NP_{ins} + le \ / \ NP + den \ / \ Verb_{transitive}$
- $VP_{transitive} \rightarrow NP_{ins} + le \ / \ NP + den \ / \ NP + i \ / \ Verb_{transitive}$
- $VP_{transitive} \rightarrow NP + de \ / \ Verb_{transitive}$
- $VP_{transitive} \rightarrow NP + de \ / \ NP + i \ / \ Verb_{transitive}$
- $VP_{transitive} \rightarrow NP + de \ / \ NP + den \ / \ Verb_{transitive}$
- $VP_{transitive} \rightarrow NP + de \ / \ NP + den \ / \ NP + i \ / \ Verb_{transitive}$
- $VP_{transitive} \rightarrow NP + de \ / \ NP_{ins} + le \ / \ Verb_{transitive}$

- $VP_{transitive} \rightarrow NP + de \ / \ NP_{ins} + le \ / \ NP + i \ / \ Verb_{transitive}$
- $VP_{transitive} \rightarrow NP + de \ / \ NP_{ins} + le \ / \ NP + den \ / \ Verb_{transitive}$
- $VP_{transitive} \rightarrow NP + de \ / \ NP_{ins} + le \ / \ NP + den \ / \ NP + i \ / \ Verb_{transitive}$
- $VP_{transitive\ human} \rightarrow Verb_{transitive\ human}$
- $VP_{transitive\ human} \rightarrow NP + i \ / \ Verb_{transitive\ human}$
- $VP_{transitive\ human} \rightarrow NP + den \ / \ Verb_{transitive\ human}$
- $VP_{transitive\ human} \rightarrow NP + den \ / \ NP + i \ / \ Verb_{transitive\ human}$
- $VP_{transitive\ human} \rightarrow NP_{ins} + le \ / \ Verb_{transitive\ human}$
- $VP_{transitive\ human} \rightarrow NP_{ins} + le \ / \ NP + i \ / \ Verb_{transitive\ human}$
- $VP_{transitive\ human} \rightarrow NP_{ins} + le \ / \ NP + den \ / \ Verb_{transitive\ human}$
- $VP_{transitive\ human} \rightarrow NP_{ins} + le \ / \ NP + den \ / \ NP + i \ / \ Verb_{transitive\ human}$
- $VP_{transitive\ human} \rightarrow NP + de \ / \ Verb_{transitive\ human}$
- $VP_{transitive\ human} \rightarrow NP + de \ / \ NP + i \ / \ Verb_{transitive\ human}$
- $VP_{transitive\ human} \rightarrow NP + de \ / \ NP + den \ / \ Verb_{transitive\ human}$
- $VP_{transitive\ human} \rightarrow NP + de \ / \ NP + den \ / \ NP + i \ / \ Verb_{transitive\ human}$
- $VP_{transitive\ human} \rightarrow NP + de \ / \ NP_{ins} + le \ / \ Verb_{transitive\ human}$
- $VP_{transitive\ human} \rightarrow NP + de \ / \ NP_{ins} + le \ / \ NP + i \ / \ Verb_{transitive\ human}$
- $VP_{transitive\ human} \rightarrow NP + de \ / \ NP_{ins} + le \ / \ NP + den \ / \ Verb_{transitive\ human}$
- $VP_{transitive\ human} \rightarrow NP + de / NP_{ins} + le / NP + den / NP + i / Verb_{transitive\ human}$

Group 15: This group includes rules that generates VP's where verb is classified as transitive mobile 1, transitive human mobile 1.
using 14: $Adjunct + verb_{transitive(human)\ mobile\ 1} \rightarrow$
$(NP + de/)(NP + le/)(NP + den/)(NP + e/)(NP + i)Verb_{transitive(human)\ mobile\ 1}$
using 16: $NP + de + X + verb_{Y\ mobile Z} \rightarrow N_{time} + de + X + verb_{Y\ mobile Z}$

using 29: $N_{time} \rightarrow \left\{ \begin{array}{l} (demonstrative\&) \left\{ \begin{array}{l} (n_{number}\&)n \text{ time a} \\ (saat)n \text{ time b} \end{array} \right\} \\ n \text{ time c} \end{array} \right\}$
using Group 3
using Group 2
generated: $Adjunct + verb_{transitive(human)\ mobile\ 1} \rightarrow$
$(NP_{time}+de/)(NP+le/)(NP+den/)(NP+e/)(NP+i)Verb_{transitive(human)\ mobile\ 1}$(I)
setting: $VP_{transitive\ mobile\ 1} \rightarrow Adjunct + verb_{transitive\ mobile\ 1}$
setting: $VP_{transitive\ human\ mobile\ 1} \rightarrow Adjunct + verb_{transitive\ human\ mobile\ 1}$
expansion of (I) and settings:

- $VP_{transitive\ mobile\ 1} \rightarrow Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP + i\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP + e\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP + e\ /\ NP + i\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP + den\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP + den\ /\ NP + i\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP + den\ /\ NP + e\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP + den\ /\ NP + e\ /\ NP + i\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{ins} + le\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{ins} + le\ /\ NP + i\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{ins} + le\ /\ NP + e\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{ins} + le\ /\ NP + e\ /\ NP + i\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{ins} + le\ /\ NP + den\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{ins} + le\ /\ NP + den\ /\ NP + i\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{ins} + le/NP + den/NP + e/NP + i/Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP + i\ /\ Verb_{transitive\ mobile\ 1}$

- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP + e\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP + e\ /\ NP + i\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP + den\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP + den\ /\ NP + i\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP + den\ /\ NP + e\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP + den\ /\ NP + e\ /\ NP + i\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + i\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + e\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + e\ /\ NP + i\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de/NP_{ins} + le/NP + den/Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + den\ /\ NP + i\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ NP + i\ /\ Verb_{transitive\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP + i\ /\ Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP + e\ /\ Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP + e\ /\ NP + i\ /\ Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP + den\ /\ Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP + den\ /\ NP + i\ /\ Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP + den\ /\ NP + e\ /\ Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP + den/NP + e/NP + i/Verb_{transitive\ human\ mobile\ 1}$

- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{ins} + le\ /\ Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{ins} + le\ /\ NP + i\ /\ Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{ins} + le\ /\ NP + e\ /\ Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{ins} + le/NP + e/NP + i/Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{ins} + le/NP + den/Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{ins} + le/NP + den/NP + i/Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{ins} + le/NP + den/NP + e/Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ NP + i\ /\ Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time} + de\ /\ Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time} + de/NP + i/Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time} + de/NP + e/Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time} + de/NP + e/NP + i/Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time} + de/NP + den/Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time} + de/NP + den/NP + i/Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time} + de/NP + den/NP + e/Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP + den\ /\ NP + e\ /\ NP + i\ /\ Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time} + de/NP_{ins} + le/Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time} + de/NP_{ins} + le/NP + i/Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time} + de/NP_{ins} + le/NP + e/Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + e\ /\ NP + i\ /\ Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time} + de/NP_{ins} + le/NP + den/Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + den\ /\ NP + i\ /\ Verb_{transitive\ human\ mobile\ 1}$

- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ Verb_{transitive\ human\ mobile\ 1}$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ NP + i\ /\ Verb_{transitive\ human\ mobile\ 1}$

Group 16: This group includes rules that generates NP's that appear as the object of verb that are classifed as mobile 2.
using 20: $NP + i/\ verb_{transitive(human)mobile2} \rightarrow$
(demostrative&)(n_{number}&)$\left\{ \begin{array}{l} N_{immobile} \\ n_{inanimate} \end{array} \right\} i/\ verb_{transitive(human)mobile2}$
setting: $NP_{mob2} \rightarrow$ (demostrative&)(n_{number}&)$\left\{ \begin{array}{l} N_{immobile} \\ n_{inanimate} \end{array} \right\}$/
using 26: $N_{immobile} \rightarrow \left\{ \begin{array}{l} n_{item} \\ n_{locale(proper)} \\ n_{number} \\ n_{substance} \end{array} \right\}$
generated: $NP_{mob2} \rightarrow$ (demostrative&)(n_{number}&)$\left\{ \begin{array}{l} \left\{ \begin{array}{l} n_{item} \\ n_{locale(proper)} \\ n_{number} \\ n_{substance} \end{array} \right\} \\ n_{inanimate} \end{array} \right\}$/(I)
expansion of (I):

- $NP_{mob2} \rightarrow n_{item}$
- $NP_{mob2} \rightarrow n_{local}$
- $NP_{mob2} \rightarrow n_{local\ proper}$
- $NP_{mob2} \rightarrow n_{number}$
- $NP_{mob2} \rightarrow n_{substance}$
- $NP_{mob2} \rightarrow n_{inanimate}$
- $NP_{mob2} \rightarrow n_{number}\ \&\ n_{item}$
- $NP_{mob2} \rightarrow n_{number}\ \&\ n_{local}$
- $NP_{mob2} \rightarrow n_{number}\ \&\ n_{local\ proper}$

- $NP_{mob2} \rightarrow n_{number}$ & n_{number}
- $NP_{mob2} \rightarrow n_{number}$ & $n_{substance}$
- $NP_{mob2} \rightarrow n_{number}$ & $n_{inanimate}$
- $NP_{mob2} \rightarrow demonstrative$ & n_{item}
- $NP_{mob2} \rightarrow demonstrative$ & n_{local}
- $NP_{mob2} \rightarrow demonstrative$ & $n_{local\ proper}$
- $NP_{mob2} \rightarrow demonstrative$ & n_{number}
- $NP_{mob2} \rightarrow demonstrative$ & $n_{substance}$
- $NP_{mob2} \rightarrow demonstrative$ & $n_{inanimate}$
- $NP_{mob2} \rightarrow demonstrative$ & n_{number} & n_{item}
- $NP_{mob2} \rightarrow demonstrative$ & n_{number} & n_{local}
- $NP_{mob2} \rightarrow demonstrative$ & n_{number} & $n_{local\ proper}$
- $NP_{mob2} \rightarrow demonstrative$ & n_{number} & n_{number}
- $NP_{mob2} \rightarrow demonstrative$ & n_{number} & $n_{substance}$
- $NP_{mob2} \rightarrow demonstrative$ & n_{number} & $n_{inanimate}$

Group 17: This group includes rules that generates VP's where verb is classified as transitive mobile 2 and transitive human mobile 2.
using 14: $Adjunct + verb_{transitive(human)\ mobile\ 2} \rightarrow$
$(NP+de/)(NP+le/)(NP+den/)(NP+e/)(NP+i)Verb_{transitive(human)\ mobile\ 2}$
using 16: $NP + de + X + verb_{Y\ mobile Z} \rightarrow N_{time} + de + X + verb_{Y\ mobile Z}$
using 29: $N_{time} \rightarrow \left\{ \begin{array}{l} (demonstrative\&) \left\{ \begin{array}{l} (n_{number}\&)n \text{ time a} \\ (saat)n \text{ time b} \end{array} \right\} \\ n \text{ time c} \end{array} \right\}$
using Group 3
using Group 2
using Group 16
generated: $Adjunct + verb_{transitive(human)\ mobile\ 1} \rightarrow$
$(NP_{time}+de/)(NP+le/)(NP+den/)(NP+e/)(NP_{mob2}+i)Verb_{transitive(human)\ mobile\ 2}$(I)

setting: $VP_{transitive\ mobile\ 2} \rightarrow Adjunct + verb_{transitive\ mobile\ 2}$
setting: $VP_{transitive\ human\ mobile\ 2} \rightarrow Adjunct + verb_{transitive\ human\ mobile\ 2}$
expansion of (I) and settings:

- $VP_{transitive\ mobile\ 2} \rightarrow Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{mob2} + i\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP + e\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP + e\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP + den\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP + den\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP + den\ /\ NP + e\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP + den\ /\ NP + e\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{ins} + le\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{ins} + le\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{ins} + le\ /\ NP + e\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{ins} + le\ /\ NP + e\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{ins} + le\ /\ NP + den\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{ins} + le\ /\ NP + den\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP + e\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP + e\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP + den\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP + den\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ mobile\ 2}$

- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP + den\ /\ NP + e\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP + den\ /\ NP + e\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de / NP_{ins} + le / NP_{mob2} + i / Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de / NP_{ins} + le / NP + e / Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + e\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de / NP_{ins} + le / NP + den / Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + den\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{mob2} + i\ /\ Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP + e\ /\ Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP + e / NP_{mob2} + i / Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP + den\ /\ Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP + den / NP_{mob2} + i / Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP + den\ /\ NP + e\ /\ Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP + den / NP + e / NP_{mob2} + i / Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{ins} + le\ /\ Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{ins} + le / NP_{mob2} + i / Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{ins} + le\ /\ NP + e\ /\ Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{ins} + le / NP + e / NP_{mob2} + i / Verb_{transitive\ human\ mobile\ 2}$

- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{ins}+le/NP+den/Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{ins}+le/NP+den/NP_{mob2}+i/Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{ins}+le/NP+den/NP+e/Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time} + de\ /\ Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time}+de/NP_{mob2}+i/Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time}+de/NP+e/Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time}+de/NP+e/NP_{mob2}+i/Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time}+de/NP+den/Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time}+de/NP+den/NP_{mob2}+i/Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time}+de/NP+den/NP+e/Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP + den\ /\ NP + e\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time}+de/NP_{ins}+le/Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time}+de/NP_{ins}+le/NP_{mob2}+i/Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time}+de/NP_{ins}+le/NP+e/Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + e\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time}+de/NP_{ins}+le/NP+den/Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time}+de\ /\ NP_{ins}+le\ /\ NP+den\ /\ NP_{mob2}+ i\ /\ Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ Verb_{transitive\ human\ mobile\ 2}$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ human\ mobile\ 2}$

Group 18: This group generates aspects of verbs.

using 4: Aspect $\rightarrow \left\{ \begin{array}{l} er \\ ecek \\ iyor \\ meli \\ miş \end{array} \right\}$

expansion of 4:

- Aspect → er
- Aspect → ecek
- Aspect → iyor
- Aspect → meli
- Aspect → miş

Group 19: This group classifies VP's.

- $VP \rightarrow VP_n$
- $VP \rightarrow VP_{human}$
- $VP \rightarrow VP_{mob}$
- $VP_n \rightarrow AdjunctiveAdj_a$
- $VP_n \rightarrow AdjunctiveAdj_b$
- $VP_n \rightarrow AdjunctiveAdj_c$
- $VP_n \rightarrow AdjunctiveAdj_d$
- $VP_n \rightarrow AdjunctiveNP$
- $VP_n \rightarrow VP_{impersonal}$
- $VP_n \rightarrow VP_{non}$
- $VP_n \rightarrow VP_{transitive}$

- $VP_{human} \rightarrow VP_{transitive\ human}$
- $VP_{human} \rightarrow VP_{transitive\ human\ mobile\ 1}$
- $VP_{human} \rightarrow VP_{transitive\ human\ mobile\ 2}$
- $VP_{mob} \rightarrow VP_{mobile\ a}$
- $VP_{mob} \rightarrow VP_{mobile\ b}$
- $VP_{mob} \rightarrow VP_{mobile}$
- $VP_{mob} \rightarrow VP_{transitive\ mobile\ 1}$
- $VP_{mob} \rightarrow VP_{transitive\ mobile\ 2}$

Group 20: This group generates NP's that are subjects of verb classified as mobile.
using 22: NP/X + $verb$ Y mobile Z $\rightarrow$
$\left\{ \begin{array}{l} (demonstrative\&)(n_{number}\&)N_{mobile} \\ pronoun \end{array} \right\}$/X + $verb$ Y mobile Z
generated: $NP_{mob} \rightarrow \left\{ \begin{array}{l} (demonstrative\&)(n_{number}\&)N_{mobile} \\ pronoun \end{array} \right\}$(I)
expansion of (I):

- $NP_{mob} \rightarrow n_{animate}$
- $NP_{mob} \rightarrow n_{animate\ proper}$
- $NP_{mob} \rightarrow n_{animate\ transport}$
- $NP_{mob} \rightarrow n_{inanimate}$
- $NP_{mob} \rightarrow n_{number}\ \&\ n_{animate}$
- $NP_{mob} \rightarrow n_{number}\ \&\ n_{animate\ proper}$
- $NP_{mob} \rightarrow n_{number}\ \&\ n_{animate\ transport}$
- $NP_{mob} \rightarrow n_{number}\ \&\ n_{inanimate}$
- $NP_{mob} \rightarrow demonstrative\ \&\ n_{animate}$
- $NP_{mob} \rightarrow demonstrative\ \&\ n_{animate\ proper}$

- $NP_{mob} \rightarrow demonstrative \,\&\, n_{animate\ transport}$
- $NP_{mob} \rightarrow demonstrative \,\&\, n_{inanimate}$
- $NP_{mob} \rightarrow demonstrative \,\&\, n_{number} \,\&\, n_{animate}$
- $NP_{mob} \rightarrow demonstrative \,\&\, n_{number} \,\&\, n_{animate\ proper}$
- $NP_{mob} \rightarrow demonstrative \,\&\, n_{number} \,\&\, n_{animate\ transport}$
- $NP_{mob} \rightarrow demonstrative \,\&\, n_{number} \,\&\, n_{inanimate}$
- $NP_{mob} \rightarrow pronoun$

Group 21: This group generates NP's that are subjects of verb classified as human.
using 21: NP/X + $verb$ Y human Z $\rightarrow$
$$\left\{ \begin{array}{l} (demonstrative\&)(n_{number}\&)n \text{ animate (proper) human} \\ ben \\ sen \\ o \end{array} \right\} /\text{X} + verb \text{ Y human Z}$$
generated: $NP_{human} \rightarrow \left\{ \begin{array}{l} (demonstrative\&)(n_{number}\&)n \text{ animate (proper) human} \\ ben \\ sen \\ o \end{array} \right\}$(I)
expansion of (I):

- $NP_{human} \rightarrow n_{animate\ human}$
- $NP_{human} \rightarrow n_{animate\ proper\ human}$
- $NP_{human} \rightarrow n_{number} \,\&\, n_{animate\ human}$
- $NP_{human} \rightarrow n_{number} \,\&\, n_{animate\ proper\ human}$
- $NP_{human} \rightarrow demonstrative \,\&\, n_{animate\ human}$
- $NP_{human} \rightarrow demonstrative \,\&\, n_{animate\ proper\ human}$
- $NP_{human} \rightarrow demonstrative \,\&\, n_{number} \,\&\, n_{animate\ human}$
- $NP_{human} \rightarrow demonstrative \,\&\, n_{number} \,\&\, n_{animate\ proper\ human}$
- $NP_{human} \rightarrow ben$

- $NP_{human} \rightarrow sen$
- $NP_{human} \rightarrow o$

Group 22: This group generates S's that have subjects and mobile verbs.
using 1: S → (NP/) (Adv/) VP (mi)/
using 22: NP/X + $verb$ $_{\text{Y mobile Z}}$ →
$\left\{ \begin{array}{l} (demonstrative\&)(n_{number}\&)N_{mobile} \\ pronoun \end{array} \right\}$/X + $verb$ $_{\text{Y mobile Z}}$
using Group 20
using Group19
generated: $S \rightarrow NP_{mob}/(Adv/)VP(mi)/(\text{I})$
expansion of (I):

- S → NP_{mob}/VP_{mob}
- S → $NP_{mob}/VP_{mob}mi$
- S → $NP_{mob}/Adv/VP_{mob}$
- S → $NP_{mob}/Adv/VP_{mob}mi$

Group 23: This group generates S's that have subjects and human verbs.
using 1: S → (NP/) (Adv/) VP (mi)/
using 21: NP/X + $verb$ $_{\text{Y human Z}}$ →
$\left\{ \begin{array}{l} (demonstrative\&)(n_{number}\&)n_{\text{ animate (proper) human}} \\ ben \\ sen \\ o \end{array} \right\}$/X + $verb$ $_{\text{Y human Z}}$
using Group 21
using Group 19
generated: $S \rightarrow NP_{human}/(Adv/)VP(mi)/(\text{I})$
expansion of (I):

- S → NP_{human}/VP_{human}
- S → $NP_{human}/VP_{human}mi$
- S → $NP_{human}/Adv/VP_{human}$

- S $\rightarrow NP_{human}/Adv/VP_{human}mi$

Group 24: This group generates S's different from the structures above.
using 1: S $\rightarrow$ (NP/) (Adv/) VP (mi)/
using Group 19
generated: $S \rightarrow (NP/)(Adv/)VP_n(mi)/$(I)
expansion of (I) and 1:

- S $\rightarrow VP$
- S $\rightarrow VPmi$
- S $\rightarrow Adv/VP$
- S $\rightarrow Adv/VPmi$
- S $\rightarrow NP/VP_n$
- S $\rightarrow NP/VP_nmi$
- S $\rightarrow NP/Adv/VP_n$
- S $\rightarrow NP/Adv/VP_nmi$

4 Practice and Results

4.1 Dependency Parsing

Treebank is composed of 5635 sentences and all of these sentences are given as input for conversion to proposed representation to written program in order to measure the success ratio of the model. Output sentences are checked by hand if they are successfully converted. Success criteria are defined as the following:

1. Sentence is converted into proposed tree model
2. Word order information is not lost

Out of 5635 sentences, 42 sentences cannot be converted to the desired format. These situations existed because of malformed sentences in Treebank. These sentences are annotated to have cyclic relations, but after careful investigations on sentences, these are seen to be annotated not correctly.

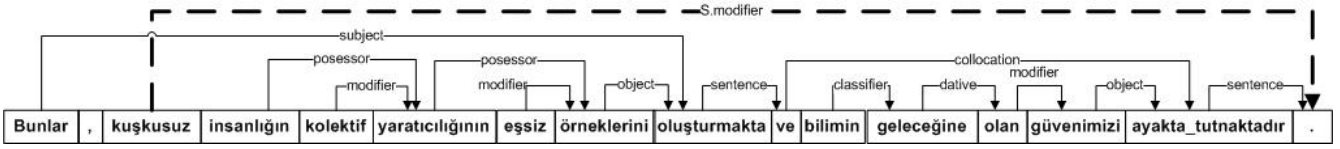

Figure 11: Dependency chart for an example failure

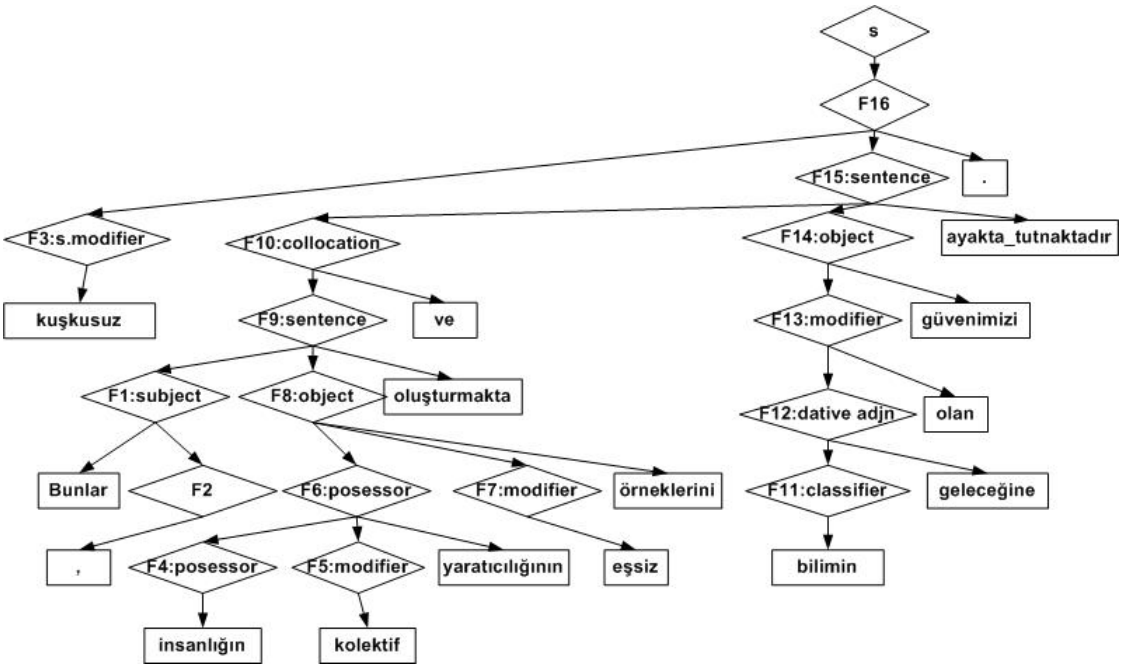

Figure 12: Transformation for an example failure

All of the rest are converted correctly according to the given algorithm into tree structure. After investigations on these sentences, non-projective constructs, such as an adverbial modifying a verb, cutting in between a modifier or the head noun making up the subject NP cause violation of second one of criteria. These are generally annotated to be sentential modifier, etc. linked to the last punctuation from where they lie in the sentence not mattering cutting the phrases. *(Refer to Figure 11)* This situation comes from the free-order constituent property of Turkish. "Bunlar, kuşkusuz insanlǵın kolektif yaratıcılǵının eşsiz örneklerini oluşturmakta ve bilimin geleceǵine olan güvenimizi ayakta tutmaktadır." sentence is a typical example of the situation above *(Refer to Figure 12)*. "kuşkusuz" is connected to "." punctuation as sentential modifier.

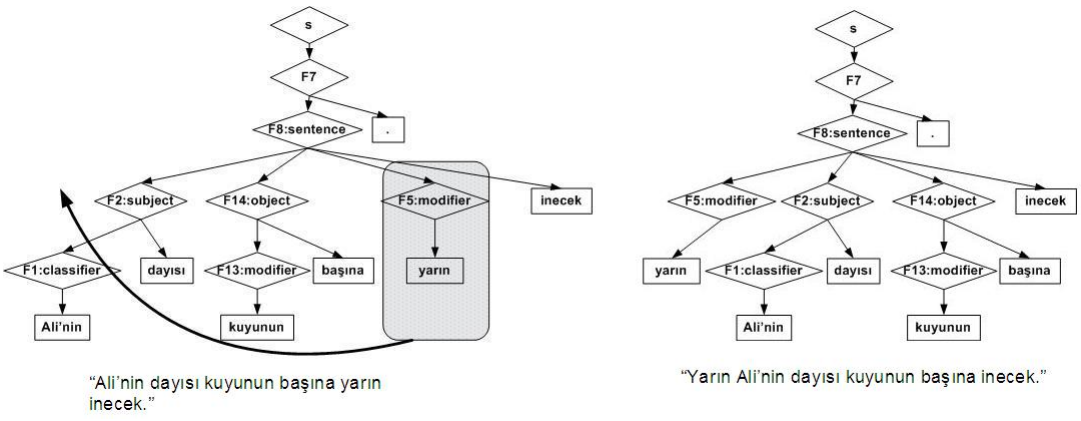

Figure 13: Application of "adverb displacement" tool

These constructs violate the model and diverge from success criteria. These, however, are very rare, summing up to 387 sentences forming % 6.9 of whole Treebank. Although example given here does not matter so much (in fact this is considered to be a new tool, moving sentential modifiers to the beginning of the sentence), this is not applicable to all cross-linked sentences.

As a result, proposed model is able to fully represent % 92.3 of the Treebank and helps "conjunct order change" and "adverb displacement" tools work. These tools need deeper unserstanding of syntax, organization of phrases, clauses functioning as complex subjects and complement clauses. Both of these tools deal with displacement of functional units and that's why proposed model does well because model's basic concern is to identify sentence order in the structure. In adverb displacement tool, we simply change the place of the main adverb that affects the sentence *(Refer to Figure 13)*. In treebank format we have to follow all the words' links backward to capture the units at the same level with the adverb but with the proposed model these level information is already included. The same argument is valid for second tool mentioned. We change the order of conjuncts around "ve" or punctuation "," which are operational units not only for simple words but also big functional units *(Refer to Figure 14)*.

Two syntactic tools mentioned above are not all we created, just the ones that are handled by means of proposed model. As a result of our studies, we were able to discover twenty morpho-syntactic watermarking tools which were defined and implemented for "Syntactic Tools for Text Watermarking".[1]

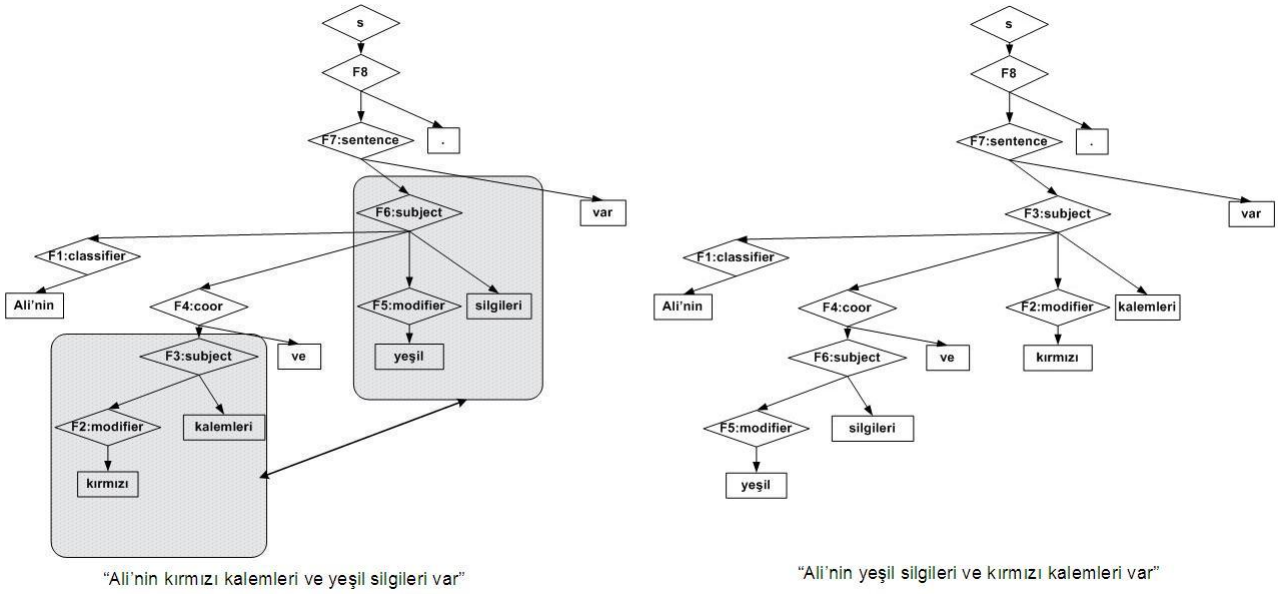

Figure 14: Application of "conjunct order change" tool

4.2 Phrase Structure Parsing

Turkish parser is implemented in Prolog, a logic language, which seemed the most suitable language for my aim. Implementation is not done by means of standard predicates but using definite clause grammar rules. Complete listing of the DCG rules of the parser can be found in Appendix D.
Since DCGs are designed for Indo-European languages in which all functionalties are addressed by means of words, it is not so appropriate for Turkish in which functionalities are addressed by suffixes. In order to distinguish the relations, deep structure of the words should be analyzed but it is impossible to do that by using DCG in standard ways. To overcome this problem, words are assumed to be morphologically analyzed beforehand, each morphological part of a word is assigned to a list element and word boundaries are defined as list elements 'x'. There are some example sentences and how they are represented:

- `"O adam hemen bana yazyor."`
  `[o,x,adam,x,hemen,x,ben,e,x,yaz,iyor]`
- `"Gidecek."`
  `[git,ecek]`
- `"Üç kalemden memnun(dur)."`

[üç,x,kalem,den,x,memnun,dir]

- "Bu üç kalemden memnun(dur)."
  [bu,x,üç,x,kalem,den,x,memnun,dir]
- "İki araba bana fazla(dır)."
  [iki,x,araba,x,ben,e,x,fazla,dir]
- "Mehmet arabayla meşgul(dür)."
  [mehmet,x,araba,le,x,megul,dir]
- "Üç köpeǵe raǵmen evden kaçmış."
  [üç,x,köpek,e,x,raǵmen,x,ev,den,x,kaç,miş]
- "O üç köpeǵe raǵmen evden kaçmış."
  [o,üç,x,köpek,e,x,raǵmen,x,ev,den,x,kaç,miş]
- "Ahmet gene Gülen için Şermin'i bekliyor."
  [ahmet,x,gene,x,gülen,x,için,x,şermin,i,x,bekle,iyor]
- "Dörtte otobüsle geliyor."
  [dört,de,x,otobüs,le,x,gel,iyor]

Execution is done by asking Prolog interperter if a list can be genarated by the rule specified. If we want to know that a list is a sentence or not we simply enter:
`s(X,[bu,x,üç,x,kalem,den,x,memnun,dir],[])`

or check a list if it constructs a verb phrase or not:

`vp(X,[otobüs,le,x,gel,iyor],[]).`

In the above predicates, X declares the resulting tree of the phrase structures if the list can be generated by the specified rule. According to recursive iteration of the garammar rules applied by Prolog, list is checked against each rule if it satisfies or not, by means of difference lists. Trees are represented by means of consisting sets using parantheses. Each set is labeled as governing rule of that structure and all of them are covered by the rule specified for the execution. Some examples are listed below for execution and output trees of them.

- `s(X,[o,x,adam,x,hemen,x,ben,e,x,yaz,iyor],[]).`
  `X=`
  `s(`

```
np_human(
 demonstrative(
  o),
 n_animate_human(
  adam)),
adv(
 particle_a(
  hemen)),
vp_human(
 vp_transitive_human_mobile_2(
  np(
   pronoun(
    ben)),
  verb_transitive_human_mobile_2(
   yaz),
  aspect(
   iyor))))
```

refer to figure 15

- ```
  s(X,[git,ecek],[]).
  X=
  s(
   vp(
    vp_mob(
     vp_mobile_a(
      verb_mobile_a(
       git),
      aspect(
       ecek)))))
  ```

refer to figure 16

- ```
  s(X,[üç,x,kalem,den,x,memnun,dir],[]).
  X=
  s(
   vp(
    vp_n(
     adjunctadj_c(
      np(
       n_number(
        üç),
       n_item(
  ```

```
      kalem)),
    adj_c(
     memnun),
    copula(
     dir)))))
.
X=
s(
 np(
  n_time_b(
   üç)),
 vp_n(
  adjunctadj_c(
   np(
    n_item(
     kalem)),
   adj_c(
    memnun),
   copula(
    dir))))
.
X=
s(
 np(
  n_number(
   üç)),
 vp_n(
  adjunctadj_c(
   np(
    n_item(
     kalem)),
   adj_c(
    memnun),
   copula(
    dir))))
```

- `s(X,[bu,x,üç,x,kalem,den,x,memnun,dir],[]).`

```
X=
s(
```

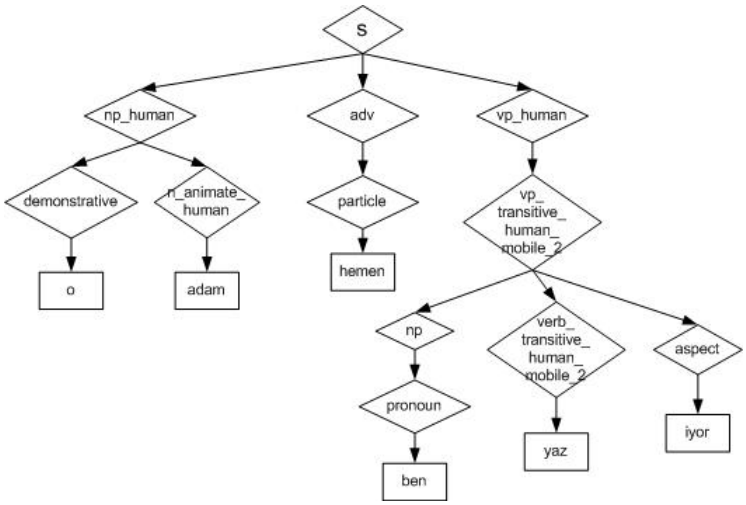

Figure 15: Tree for `s(X,[o,x,adam,x,hemen,x,ben,e,x,yaz,iyor],[])`.

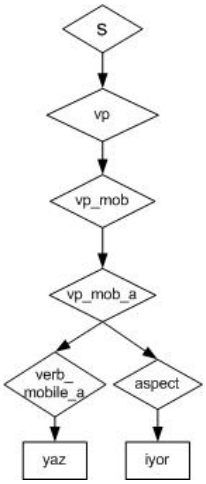

Figure 16: Tree for `s(X,[git,ecek],[])`.

```
 vp(
  vp_n(
   adjunctadj_c(
    np(
     demonstrative(
      bu),
     n_number(
      üç),
     n_item(
      kalem)),
    adj_c(
     memnun),
    copula(
     dir)))))
.
X=
s(
 np(
  pronoun(
   bu)),
 vp_n(
  adjunctadj_c(
   np(
    n_number(
     üç),
    n_item(
     kalem)),
   adj_c(
    memnun),
   copula(


    dir))))
.
X=
s(
 np(
  demonstrative(
   bu),
  n_number(
```

```
    üç)),
   vp_n(
    adjunctadj_c(
     np(
      n_item(
       kalem)),
     adj_c(
      memnun),
     copula(
      dir))))
```

- ```
  s(X,[iki,x,araba,x,ben,e,x,fazla,dir],[]).
  X=
  s(
   np(
    n_number(
     iki),
    n_inanimate(
     araba)),
   vp_n(
    adjunctadj_b(
     np(
      pronoun(
       ben)),
     adj_b(
      fazla),
     copula(
      dir))))
  ```

- ```
  s(X,[mehmet,x,araba,le,x,megul,dir],[]).
  X=
  s(
   np(
    n_animate_proper_human(
     mehmet)),
   vp_n(
    adjunctadj_d(
     np_ins(
      n_inanimate(
  ```

```
      araba)),
    adj_d(
     megul),
    copula(
     dir))))
```

- ```
  s(X,[üç,x,köpek,e,x,rağmen,x,ev,den,x,kaç,miş],[]).
  X=
  s(
   adv(
    postposphrase_e(
     np(
      n_number(
       üç),
      n_animate(
       köpek)),
     postposition_e(
      rağmen))),
   vp(
    vp_mob(
     vp_mobile(
      np(
       n_locale(
        ev)),
      verb_mobile(
       kaç),
      aspect(
       miş)))))
  ```

- ```
  s(X,[o,x,üç,x,köpek,e,x,rağmen,x,kaç,miş],[]).
  X=
  s(
   adv(
    postposphrase_e(
     np(
      demonstrative(
       o),
      n_number(
       üç),
  ```

```
    n_animate(
     köpek)),
   postposition_e(
    rağmen))),
 vp(
  vp_mob(
   vp_mobile(
    verb_mobile(
     kaç),
    aspect(
     miş)))))
.
X=
s(
 np_mob(
  pronoun(
   o)),
 adv(
  postposphrase_e(
   np(
    n_number(
     üç),
    n_animate(
     köpek)),
   postposition_e(
    rağmen))),
 vp_mob(
  vp_mobile(
   verb_mobile(
    kaç),
   aspect(
    miş))))
```

- `s(X,[ahmet,x,gene,x,gülen,x,için,x,şermin,i,x,bekle,iyor],[]).`

```
X=
s(
 np(
  n_animate_proper_human(
   ahmet)),
```

```
 adv(
  particle_a(
   gene),
  postposphrase_f(
   np(
    n_animate_proper_human(
     gülen)),
   postposition_f(
    için))),
 vp_n(
  vp_transitive(
   np(
    n_animate_proper_human(
     şermin)),
   verb_transitive(
    bekle),
   aspect(
    iyor))))
```

- ```
  s(X,[dört,de,x,otobüs,le,x,gel,iyor],[]).
  X=
  s(
   vp(
    vp_mob(
     vp_mobile_b(
      np_time(
       n_time_b(
        dört)),
      np_ins(
       n_inanimate(
        otobüs)),
      verb_mobile_b(
       gel),
      aspect(
       iyor)))))
  ```

As it is seen in sentence "O üç köpeǵe rağmen evden kaçmış." there are two different parse trees generated after execution. Parser reflects the ambiguity of Turkish in this sentence as it is in natural language. Although such ambiguities
  ```

are eliminated by using commas after subject in natural language, parser does not recognize this solution since it does not consider any punctuation.
After all conversion process there appeared 249 rules, which can be classified as such:

- 120 NP rules (subject or object function)
- 79 ADV rules
- 277 VP and related rules (classification)
- 16 S rules

These rules are implemented from higher level rules to lower level ones.

5 Conclusion

Wordnode-Relationnode model is needed to implement some of the tools of Text Watermarking for Turkish. Although it does not have full capability to represent the whole Treebank now, it is an efficient output for natural language processing. Surely first action to be taken will be to eliminate the minor percentage of failure and when these constructs are modeled in a favorable manner, this will be major candidate for the representation of Treebank.
As it is known by us, dependency representation is not the only way to handle sentences in a language. Phrase structures are widely used and there are several treebanks constructed in this manner for other languages. Turkish is known not to have such a constitutional treebank. My study in phrase structure parsing is a starting phase for such treebanks. Parsing for simple sentences can be achieved and parse trees for these sentences can be collected in a repository.
Although parser is very primitive now, it is available for some further development with little effort. At this stage free-order property of Turkish language is not implemented for the parser but this can be implemented by adding rules generated as all phrase permutations of existing rules. Also little further development will enable parser to recognize sentences where question suffix **mi** is not in the verb phrase but in any other one. This can be achieved by creating new that propagates mi suffix to lower level rules.
Some researches suggest that combined models, where structural pharases and dependency relations are both represented in a tree model, are fully utilized models for natural language processing. After building process of phrase structural treebanks, combined models can be created conveniently by means of using these two models.

6 Appendix A

6.1 Transform.java

This appendix is the listing of conversion utility that is mentioned in section 2.3. This utility is coded in Java platform because of its rich set of XML manipulaton libraries. For implementation Java Software Development Kit 5 is used with NetBeans integrated development environment. You can find additional information about code in comments in code.

```
/* Main.java
*
* Created on October 18, 2006, 8:25 PM
*
* ERSIN IHSAN UNKAR
* CMPE 491
*
*/

package xmlparsing;
//io packages
import java.io.*;
// DOM classes.
import com.sun.org.apache.xerces.internal.parsers.DOMParser;
import org.w3c.dom.*;
//JAXP 1.1
import javax.xml.parsers.*;
import javax.xml.transform.OutputKeys;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;

public class Main {
  public Main() {
  }
  public static void main(String[] args) {
    try{
      DOMParser inParser = new DOMParser();
      int ss=1;
```

```
      inParser.parse("tree ("+ss+").xml");//input file name
      File outXml=new File("outtree"+ss+".xml");//output filename
      Document inDocument = inParser.getDocument();
      NodeList inNodes = inDocument.getElementsByTagName("W"); //get
word nodes
      Document xmldoc = null;
      DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
      DocumentBuilder builder = factory.newDocumentBuilder();
      DOMImplementation impl = builder.getDOMImplementation();
      Element e = null;
      Element f =null;
      Node n = null;
      xmldoc = impl.createDocument(null, "SENTENCE", null);
      Element root = xmldoc.getDocumentElement();
      String REL=null;
      String IX=null;
      for(int i = 0; i < inNodes.getLength(); i ++) {//traverse all nodes
        Element wordElement =(Element)inNodes.item(i);
        REL=wordElement.getAttribute("REL");
        REL=REL.substring(REL.indexOf("(")+1,REL.indexOf(")")).trim();
        IX=wordElement.getAttribute("IX");
        f=xmldoc.createElementNS(null,"F"+IX);//create functional node
        f.setAttribute("ID",IX);
        f.setAttribute("IX",IX);
        f.setAttribute("Function",REL);
        e = xmldoc.createElementNS(null, "W");//create word node
        e.setAttributeNS(null, "IX", IX);
        e.setAttributeNS(null, "LEM", wordElement.getAttribute("LEM"));
        e.setAttributeNS(null, "MORPH", wordElement.getAttribute("MORPH"));
        e.setAttributeNS(null, "REL",wordElement.getAttribute("REL"));
        if (REL.equals("")){
          if(wordElement.getFirstChild().getTextContent().equals(".")){
            f.setAttribute("Function","FULL STOP");
          }
        }
        n=xmldoc.createTextNode(wordElement.getFirstChild().getNodeValue());
        e.appendChild(n);
        f.appendChild(e);// append word node to functional node as its
child
        root.appendChild(f);
```

```
      }
      NodeList tempNodes1=null;
      NodeList insertedNode=null;
      NodeList targetNode=null;
      for(int i =0 ; i <inNodes.getLength(); i++) {
        Element wordElement =(Element)inNodes.item(i);
        IX=wordElement.getAttribute("IX");
        REL=wordElement.getAttribute("REL");//get relation
        REL=REL.substring(REL.indexOf("(")+1,REL.indexOf(")")).trim();
        e=xmldoc.getElementById(String.valueOf(i));
        if(REL.equals("")==true){
          insertedNode=xmldoc.getElementsByTagName("F"+IX);//if relation
is null
          root.appendChild(insertedNode.item(0));
        }
        else{
          REL=wordElement.getAttribute("REL");
          REL=REL.split(",",2)[0];
          REL=REL.substring(1);
          insertedNode=xmldoc.getElementsByTagName("F"+IX);
          targetNode=xmldoc.getElementsByTagName("F"+REL);
          NodeList childrenOfTarget=targetNode.item(0).getChildNodes();
          for(int j = 0 ;j <childrenOfTarget.getLength(); j++) {//find
correct place as a child
            String IXofchild=((Element)childrenOfTarget.item(j)).getAttribute("IX");
            if(Integer.parseInt(IX)<Integer.parseInt(IXofchild)){
              targetNode.item(0).insertBefore( insertedNode.item(0),
childrenOfTarget.item(j));
              break;
            }
            targetNode.item(0).appendChild(insertedNode.item(0));
          }
        }
      }
      BufferedWriter out=new BufferedWriter(new FileWriter(outXml));
      DOMSource domSource = new DOMSource(xmldoc);
      StreamResult streamResult = new StreamResult(out);
      TransformerFactory tf = TransformerFactory.newInstance();
      Transformer serializer = tf.newTransformer();
      serializer.setOutputProperty(OutputKeys.ENCODING,"ISO-8859-9");
```

```
      serializer.setOutputProperty(OutputKeys.INDENT,"yes");
      serializer.transform(domSource, streamResult);
      out.close();
    }
    catch(Exception e){
      e.printStackTrace();
    }
  }
}
```

7 Appendix B

7.1 Meskill's Transformational Grammar

This study is first section in Meskill's Ph.D Thesis, titled as *A Transformational Analysis Of Turkish Syntax*. Section, titled as *Immediate Constituent Expansion Rules*, consists of tranformational rules which construct a base for complex sentence forms that can be generated by applying optional transformations.
This study is the source for rule conversion sessions explained in section 3.3. Resulting rules after conversion sessions are listed in Appendix C.

1. S → (NP/) (Adv/) VP (mi)/
2. S → (Adjunct) $\left\{ \begin{array}{l} V_l \\ V \end{array} \right\}$
3. $V_c \rightarrow \left\{ \begin{array}{l} V + Aspect \\ Adj \\ NP \end{array} \right\}$ copula
4. Aspect → $\left\{ \begin{array}{l} er \\ ecek \\ iyor \\ meli \\ miş \end{array} \right\}$
5. Adj → $\left\{ \begin{array}{l} adj_a \\ adj_b \\ adj_c \\ adj_d \end{array} \right\}$
6. V → $\left\{ \begin{array}{l} verb_{impersonal} \\ Vb_{(mobile)} \end{array} \right\}$
7. Vb → $Verb_{(tansitive(human))}$
8. $Verb_{transitive(human)mobile} \rightarrow verb_{transitive(human)mobile\left\{ \begin{array}{l} 1 \\ 2 \end{array} \right\}}$
9. $Verb_{mobile} \rightarrow verb_{mobile(\left\{ \begin{array}{l} a \\ b \end{array} \right\})}$

10. $Adjunct \begin{bmatrix} \begin{bmatrix} NP \\ adj \begin{bmatrix} a \\ b \end{bmatrix} \end{bmatrix} \\ adj_c \\ adj_d \end{bmatrix} copula \rightarrow (NP{+}de/) \begin{bmatrix} (NP+e/) \begin{bmatrix} NP \\ adj \begin{bmatrix} a \\ b \end{bmatrix} \end{bmatrix} \\ (NP+\begin{Bmatrix} e \\ den \end{Bmatrix}/)adj_l \\ (NP+le/)adj_d \end{bmatrix} copula$

11. $Adjunct \begin{bmatrix} verb_{impersonal} \\ verb \quad {}_{mobile(}\begin{bmatrix} a \\ b \end{bmatrix}{}_{)} \end{bmatrix} \rightarrow$

(NP+de/)(NP+le/)(NP+den/)(NP+e/) $\begin{bmatrix} verb_{impersonal} \\ verb \quad {}_{mobile(}\begin{bmatrix} a \\ b \end{bmatrix}{}_{)} \end{bmatrix}$

12. Adjunct + Verb $\rightarrow$ (NP+de/)(NP+le/)(NP+den) Verb

13. $Adjunct + Verb_{transitive(human)} \rightarrow$
$(NP + de/)(NP + le/)(NP + den/)(NP + i)Verb_{transitive(human)}$

14. $Adjunct + verb \quad {}_{transitive(human)mobile}\begin{bmatrix} 1 \\ 2 \end{bmatrix} \rightarrow$
$(NP{+}de/)(NP{+}le/)(NP{+}den/)(NP{+}e/)(NP{+}i)Verb \quad {}_{transitive(human)mobile}\begin{bmatrix} 1 \\ 2 \end{bmatrix}$

15. Adv $\rightarrow (particle \begin{Bmatrix} a \\ b \end{Bmatrix})(\begin{Bmatrix} (NP + den\&)posposition_a \\ NP + den\&posposition \begin{Bmatrix} a \\ b \end{Bmatrix} \end{Bmatrix}/)$
$(\begin{Bmatrix} (NP + e\&)postposition_d \\ NP + e\&postposition_e \end{Bmatrix}/)(NP\&postposition_f)$

16. $NP + de + X + verb_{YmobileZ} \rightarrow N_{time} + de + X + verb_{YmobileZ}$

17. NP + le + X $\begin{bmatrix} verb_{impersonal} \\ Verb \end{bmatrix} \rightarrow$ (demonstrative&) $n_{abstract}{+}le{+}X \begin{bmatrix} verb_{impersonal} \\ Verb \end{bmatrix}$

18. NP+ le $\rightarrow (demostrative\&)(n_{number}) \begin{Bmatrix} N_{immobile} \\ n_{inanimate} \\ n_{\text{animate transport}} \end{Bmatrix} le$

19. $NP + den\&postposition_c \rightarrow \begin{Bmatrix} pronoun \\ Nominal \end{Bmatrix} den\&postposition_l$

20. $NP + i/\ verb_{transitive(human)mobile2} \rightarrow$
$(\text{demostrative\&})(n_{number}\&)\left\{ \begin{array}{l} N_{immobile} \\ n_{inanimate} \end{array} \right\} i/\ verb_{transitive(human)mobile2}$

21. $\text{NP/X} + verb\ _{\text{Y human Z}} \rightarrow$
$\left\{ \begin{array}{l} (demonstrative\&)(n_{number}\&)n\ _{\text{animate (proper) human}} \\ ben \\ sen \\ o \end{array} \right\} /\text{X} + verb\ _{\text{Y human Z}}$

22. $\text{NP/X} + verb\ _{\text{Y mobile Z}} \rightarrow$
$\left\{ \begin{array}{l} (demonstrative\&)(n_{number}\&)N_{mobile} \\ pronoun \end{array} \right\} /\text{X} + verb\ _{\text{Y mobile Z}}$

23. $\text{NP} \rightarrow \left\{ \begin{array}{l} Nominal \\ pronoun \\ bura \\ şura \\ ora \\ nere \end{array} \right\}$

24. $\text{Nominal} \rightarrow \left\{ \begin{array}{l} (demonstarative\&) \left\{ \begin{array}{l} (n_{number}\&)\ \text{Nom} \\ n_{abstract} \end{array} \right\} \\ N_{time} \end{array} \right\}$

25. $\text{Nom} \rightarrow \left\{ \begin{array}{l} N_{immobile} \\ N_{mobile} \end{array} \right\}$

26. $N_{immobile} \rightarrow \left\{ \begin{array}{l} n_{item} \\ n_{locale(proper)} \\ n_{number} \\ n_{substance} \end{array} \right\}$

27. $N_{mobile} \rightarrow \left\{ \begin{array}{l} N_{animate} \\ n_{inanimate} \end{array} \right\}$

28. $N_{animate} \rightarrow \left\{ \begin{array}{l} n\ _{\text{animate transport}} \\ n\ _{\text{animate (proper) (human)}} \end{array} \right\}$

29. $N_{time} \rightarrow \left\{ \begin{array}{l} (demonstrative\&) \left\{ \begin{array}{l} (n_{number}\&)n\ _{\text{time a}} \\ (saat)n\ _{\text{time b}} \end{array} \right\} \\ n\ _{\text{time c}} \end{array} \right\}$

30. $n_{\text{ time c}} + \text{X}\left[\begin{array}{l} Verb_Y \\ verb_Z \end{array}\right]\left(\left[\begin{array}{l} er \\ ecek \\ iyor \\ meli \end{array}\right]\right) \rightarrow \left\{\begin{array}{l} \text{şimdi} \\ yarn \end{array}\right\} \text{X}\left[\begin{array}{l} Verb_Y \\ verb_Z \end{array}\right]\left(\left[\begin{array}{l} er \\ ecek \\ iyor \\ meli \end{array}\right]\right)$

31. $n_{\text{ time c}} + \text{X} + \text{miş} \rightarrow \left\{\begin{array}{l} \text{şimdi} \\ \text{dün} \end{array}\right\} \text{X} + \text{miş}$

8 Appendix C

8.1 Context-Free Grammar used for parser

This appendix is a full compilation of all the rules used for Turkish parser. This compilation is not grouped according to rule generation sessions. For those sessions please refer to section 3.3. Rules are ordered in implementation order which is from higer level rules to lower ones.

- S $\rightarrow VP$
- S $\rightarrow VPmi$
- S $\rightarrow Adv/VP$
- S $\rightarrow Adv/VPmi$
- S $\rightarrow NP/VP_n$
- S $\rightarrow NP/VP_nmi$
- S $\rightarrow NP/Adv/VP_n$
- S $\rightarrow NP/Adv/VP_nmi$
- S $\rightarrow NP_{human}/VP_{human}$
- S $\rightarrow NP_{human}/VP_{human}mi$
- S $\rightarrow NP_{human}/Adv/VP_{human}$
- S $\rightarrow NP_{human}/Adv/VP_{human}mi$
- S $\rightarrow NP_{mob}/VP_{mob}$
- S $\rightarrow NP_{mob}/VP_{mob}mi$
- S $\rightarrow NP_{mob}/Adv/VP_{mob}$
- S $\rightarrow NP_{mob}/Adv/VP_{mob}mi$
- $VP_n \rightarrow AdjunctiveAdj_a$
- $VP_n \rightarrow AdjunctiveAdj_b$
- $VP_n \rightarrow AdjunctiveAdj_c$
- $VP_n \rightarrow AdjunctiveNP$

- $VP_n \rightarrow VP_{impersonal}$
- $VP_n \rightarrow VP_n$
- $VP_n \rightarrow VP_{transitive}$
- $VP_{human} \rightarrow VP_{transitive\ human}$
- $VP_{mob} \rightarrow VP_{mobile\ a}$
- $VP_{mob} \rightarrow VP_{mobile\ b}$
- $VP_{mob} \rightarrow VP_{mobile}$
- $VP_{mob} \rightarrow VP_{transitive\ mobile\ 1}$
- $VP_{mob} \rightarrow VP_{transitive\ mobile\ 2}$
- $VP_{human} \rightarrow VP_{transitive\ human\ mobile\ 1}$
- $VP_{human} \rightarrow VP_{transitive\ human\ mobile\ 2}$
- $VP \rightarrow VP_n$
- $VP \rightarrow VP_{human}$
- $VP \rightarrow VP_{mob}$
- Aspect $\rightarrow$ er
- Aspect $\rightarrow$ ecek
- Aspect $\rightarrow$ iyor
- Aspect $\rightarrow$ meli
- Aspect $\rightarrow$ miş
- $AdjunctiveNP \rightarrow NP + copula$
- $AdjunctiveNP \rightarrow NP + e\ /\ NP + copula$
- $AdjunctiveNP \rightarrow NP + de\ /\ NP + copula$
- $AdjunctiveNP \rightarrow NP + de\ /\ NP + e\ /\ NP + copula$
- $AdjunctiveAdj_a \rightarrow adj_a + copula$
- $AdjunctiveAdj_a \rightarrow NP + e\ /\ adj_a + copula$

- $AdjunctiveAdj_a \rightarrow NP + de \ / \ adj_a + copula$
- $AdjunctiveAdj_a \rightarrow NP + de \ / \ NP + e \ / \ adj_a + copula$
- $AdjunctiveAdj_b \rightarrow adj_b + copula$
- $AdjunctiveAdj_b \rightarrow NP + e \ / \ adj_b + copula$
- $AdjunctiveAdj_b \rightarrow NP + de \ / \ adj_b + copula$
- $AdjunctiveAdj_b \rightarrow NP + de \ / \ NP + e \ / \ adj_b + copula$
- $AdjunctiveAdj_c \rightarrow adj_c + copula$
- $AdjunctiveAdj_c \rightarrow NP + e \ / \ adj_c + copula$
- $AdjunctiveAdj_c \rightarrow NP + de \ / \ adj_c + copula$
- $AdjunctiveAdj_c \rightarrow NP + de \ / \ NP + e \ / \ adj_c + copula$
- $AdjunctiveAdj_c \rightarrow NP + den \ / \ adj_c + copula$
- $AdjunctiveAdj_c \rightarrow NP + de \ / \ NP + den \ / \ adj_c + copula$
- $AdjunctiveAdj_d \rightarrow adj_d + copula$
- $AdjunctiveAdj_d \rightarrow NP_{ins} + le \ / \ adj_d + copula$
- $AdjunctiveAdj_d \rightarrow NP + de \ / \ adj_d + copula$
- $AdjunctiveAdj_d \rightarrow NP + de \ / \ NP_{ins} + le \ / \ adj_d + copula$
- $VP_{impersonal} \rightarrow verb_{impersonal} + Aspect$
- $VP_{impersonal} \rightarrow NP + e \ / \ verb_{impersonal} + Aspect$
- $VP_{impersonal} \rightarrow NP + den \ / \ verb_{impersonal} + Aspect$
- $VP_{impersonal} \rightarrow NP + den \ / \ NP + e \ / \ verb_{impersonal} + Aspect$
- $VP_{impersonal} \rightarrow NP_{abstract} + le \ / \ verb_{impersonal} + Aspect$
- $VP_{impersonal} \rightarrow NP_{abstract} + le \ / \ NP + e \ / \ verb_{impersonal} + Aspect$
- $VP_{impersonal} \rightarrow NP_{abstract} + le \ / \ NP + den \ / \ verb_{impersonal} + Aspect$
- $VP_{impersonal} \rightarrow NP_{abstract} + le \,/\, NP + den \,/\, NP + e \,/\, verb_{impersonal} + Aspect$
- $VP_{impersonal} \rightarrow NP + de \ / \ verb_{impersonal} + Aspect$

- $VP_{impersonal} \rightarrow NP + de \ / \ NP + e \ / \ verb_{impersonal} + Aspect$
- $VP_{impersonal} \rightarrow NP + de \ / \ NP + den \ / \ verb_{impersonal} + Aspect$
- $VP_{impersonal} \rightarrow NP + de \ / \ NP + den \ / \ NP + e \ / \ verb_{impersonal} + Aspect$
- $VP_{impersonal} \rightarrow NP + de \ / \ NP_{abstract} + le \ / \ verb_{impersonal} + Aspect$
- $VP_{impersonal} \rightarrow NP + de \ / \ NP_{abstract} + le \ / \ NP + e \ / \ verb_{impersonal} + Aspect$
- $VP_{impersonal} \rightarrow NP + de \ / \ NP_{abstract} + le \ / \ NP + den \ / \ verb_{impersonal} + Aspect$
- $VP_{impersonal} \rightarrow NP + de \ / \ NP_{abstract} + le \ / \ NP + den \ / \ NP + e \ / \ verb_{impersonal} + Aspect$
- $VP_{mobile} \rightarrow verb_{mobile} + Aspect$
- $VP_{mobile} \rightarrow NP + e \ / \ verb_{mobile} + Aspect$
- $VP_{mobile} \rightarrow NP + den \ / \ verb_{mobile} + Aspect$
- $VP_{mobile} \rightarrow NP + den \ / \ NP + e \ / \ verb_{mobile} + Aspect$
- $VP_{mobile} \rightarrow NP_{ins} + le \ / \ verb_{mobile} + Aspect$
- $VP_{mobile} \rightarrow NP_{ins} + le \ / \ NP + e \ / \ verb_{mobile} + Aspect$
- $VP_{mobile} \rightarrow NP_{ins} + le \ / \ NP + den \ / \ verb_{mobile} + Aspect$
- $VP_{mobile} \rightarrow NP_{ins} + le \ / \ NP + den \ / \ NP + e \ / \ verb_{mobile} + Aspect$
- $VP_{mobile} \rightarrow NP_{time} + de \ / \ verb_{mobile} + Aspect$
- $VP_{mobile} \rightarrow NP_{time} + de \ / \ NP + e \ / \ verb_{mobile} + Aspect$
- $VP_{mobile} \rightarrow NP_{time} + de \ / \ NP + den \ / \ verb_{mobile} + Aspect$
- $VP_{mobile} \rightarrow NP_{time} + de \ / \ NP + den \ / \ NP + e \ / \ verb_{mobile} + Aspect$
- $VP_{mobile} \rightarrow NP_{time} + de \ / \ NP_{ins} + le \ / \ verb_{mobile} + Aspect$
- $VP_{mobile} \rightarrow NP_{time} + de \ / \ NP_{ins} + le \ / \ NP + e \ / \ verb_{mobile} + Aspect$
- $VP_{mobile} \rightarrow NP_{time} + de \ / \ NP_{ins} + le \ / \ NP + den \ / \ verb_{mobile} + Aspect$
- $VP_{mobile} \rightarrow NP_{time} + de \ / \ NP_{ins} + le \ / \ NP + den \ / \ NP + e \ / \ verb_{mobile} + Aspect$
- $VP_{mobile\,a} \rightarrow verb_{mobile\,a} + Aspect$

- $VP_{mobile\ a} \rightarrow NP + e\ /\ verb_{mobile\ a} + Aspect$
- $VP_{mobile\ a} \rightarrow NP + den\ /\ verb_{mobile\ a} + Aspect$
- $VP_{mobile\ a} \rightarrow NP + den\ /\ NP + e\ /\ verb_{mobile\ a} + Aspect$
- $VP_{mobile\ a} \rightarrow NP_{ins} + le\ /\ verb_{mobile\ a} + Aspect$
- $VP_{mobile\ a} \rightarrow NP_{ins} + le\ /\ NP + e\ /\ verb_{mobile\ a} + Aspect$
- $VP_{mobile\ a} \rightarrow NP_{ins} + le\ /\ NP + den\ /\ verb_{mobile\ a} + Aspect$
- $VP_{mobile\ a} \rightarrow NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ verb_{mobile\ a} + Aspect$
- $VP_{mobile\ a} \rightarrow NP_{time} + de\ /\ verb_{mobile\ a} + Aspect$
- $VP_{mobile\ a} \rightarrow NP_{time} + de\ /\ NP + e\ /\ verb_{mobile\ a} + Aspect$
- $VP_{mobile\ a} \rightarrow NP_{time} + de\ /\ NP + den\ /\ verb_{mobile\ a} + Aspect$
- $VP_{mobile\ a} \rightarrow NP_{time} + de\ /\ NP + den\ /\ NP + e\ /\ verb_{mobile\ a} + Aspect$
- $VP_{mobile\ a} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ verb_{mobile\ a} + Aspect$
- $VP_{mobile\ a} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + e\ /\ verb_{mobile\ a} + Aspect$
- $VP_{mobile\ a} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + den\ /\ verb_{mobile\ a} + Aspect$
- $VP_{mobile\ a} \rightarrow NP_{time} + de/NP_{ins} + le/NP + den/NP + e/verb_{mobile\ a} + Aspect$
- $VP_{mobile\ b} \rightarrow verb_{mobile\ b} + Aspect$
- $VP_{mobile\ b} \rightarrow NP + e\ /\ verb_{mobile\ b} + Aspect$
- $VP_{mobile\ b} \rightarrow NP + den\ /\ verb_{mobile\ b} + Aspect$
- $VP_{mobile\ b} \rightarrow NP + den\ /\ NP + e\ /\ verb_{mobile\ b} + Aspect$
- $VP_{mobile\ b} \rightarrow NP_{ins} + le\ /\ verb_{mobile\ b} + Aspect$
- $VP_{mobile\ b} \rightarrow NP_{ins} + le\ /\ NP + e\ /\ verb_{mobile\ b} + Aspect$
- $VP_{mobile\ b} \rightarrow NP_{ins} + le\ /\ NP + den\ /\ verb_{mobile\ b} + Aspect$
- $VP_{mobile\ b} \rightarrow NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ verb_{mobile\ b} + Aspect$
- $VP_{mobile\ b} \rightarrow NP_{time} + de\ /\ verb_{mobile\ b} + Aspect$
- $VP_{mobile\ b} \rightarrow NP_{time} + de\ /\ NP + e\ /\ verb_{mobile\ b} + Aspect$

- $VP_{mobile\,b} \rightarrow NP_{time} + de \ / \ NP + den \ / \ verb_{mobile\,b} + Aspect$
- $VP_{mobile\,b} \rightarrow NP_{time} + de \ / \ NP + den \ / \ NP + e \ / \ verb_{mobile\,b} + Aspect$
- $VP_{mobile\,b} \rightarrow NP_{time} + de \ / \ NP_{ins} + le \ / \ verb_{mobile\,b} + Aspect$
- $VP_{mobile\,b} \rightarrow NP_{time} + de \ / \ NP_{ins} + le \ / \ NP + e \ / \ verb_{mobile\,b} + Aspect$
- $VP_{mobile\,b} \rightarrow NP_{time} + de \ / \ NP_{ins} + le \ / \ NP + den \ / \ verb_{mobile\,b} + Aspect$
- $VP_{mobile\,b} \rightarrow NP_{time}{+}de/NP_{ins}{+}le/NP{+}den/NP{+}e/verb_{mobile\,b}{+}Aspect$
- $VP_{non} \rightarrow Verb + Aspect$
- $VP_{non} \rightarrow NP + den \ / \ Verb + Aspect$
- $VP_{non} \rightarrow NP_{abstract} + le \ / \ Verb + Aspect$
- $VP_{non} \rightarrow NP_{abstract} + le \ / \ NP + den \ / \ Verb + Aspect$
- $VP_{non} \rightarrow NP + de \ / \ Verb + Aspect$
- $VP_{non} \rightarrow NP + de \ / \ NP + den \ / \ Verb + Aspect$
- $VP_{non} \rightarrow NP + de \ / \ NP_{abstract} + le \ / \ Verb + Aspect$
- $VP_{non} \rightarrow NP + de \ / \ NP_{abstract} + le \ / \ NP + den \ / \ Verb + Aspect$
- $VP_{transitive} \rightarrow Verb_{transitive} + Aspect$
- $VP_{transitive} \rightarrow NP + i \ / \ Verb_{transitive} + Aspect$
- $VP_{transitive} \rightarrow NP + den \ / \ Verb_{transitive} + Aspect$
- $VP_{transitive} \rightarrow NP + den \ / \ NP + i \ / \ Verb_{transitive} + Aspect$
- $VP_{transitive} \rightarrow NP_{ins} + le \ / \ Verb_{transitive} + Aspect$
- $VP_{transitive} \rightarrow NP_{ins} + le \ / \ NP + i \ / \ Verb_{transitive} + Aspect$
- $VP_{transitive} \rightarrow NP_{ins} + le \ / \ NP + den \ / \ Verb_{transitive} + Aspect$
- $VP_{transitive} \rightarrow NP_{ins} + le \ / \ NP + den \ / \ NP + i \ / \ Verb_{transitive} + Aspect$
- $VP_{transitive} \rightarrow NP + de \ / \ Verb_{transitive} + Aspect$
- $VP_{transitive} \rightarrow NP + de \ / \ NP + i \ / \ Verb_{transitive} + Aspect$
- $VP_{transitive} \rightarrow NP + de \ / \ NP + den \ / \ Verb_{transitive} + Aspect$

- $VP_{transitive} \rightarrow NP + de \ / \ NP + den \ / \ NP + i \ / \ Verb_{transitive} + Aspect$
- $VP_{transitive} \rightarrow NP + de \ / \ NP_{ins} + le \ / \ Verb_{transitive} + Aspect$
- $VP_{transitive} \rightarrow NP + de \ / \ NP_{ins} + le \ / \ NP + i \ / \ Verb_{transitive} + Aspect$
- $VP_{transitive} \rightarrow NP + de \ / \ NP_{ins} + le \ / \ NP + den \ / \ Verb_{transitive} + Aspect$
- $VP_{transitive} \rightarrow NP + de / NP_{ins} + le / NP + den / NP + i / Verb_{transitive} + Aspect$
- $VP_{transitive\ human} \rightarrow Verb_{transitive\ human} + Aspect$
- $VP_{transitive\ human} \rightarrow NP + i \ / \ Verb_{transitive\ human} + Aspect$
- $VP_{transitive\ human} \rightarrow NP + den \ / \ Verb_{transitive\ human} + Aspect$
- $VP_{transitive\ human} \rightarrow NP + den \ / \ NP + i \ / \ Verb_{transitive\ human} + Aspect$
- $VP_{transitive\ human} \rightarrow NP_{ins} + le \ / \ Verb_{transitive\ human} + Aspect$
- $VP_{transitive\ human} \rightarrow NP_{ins} + le \ / \ NP + i \ / \ Verb_{transitive\ human} + Aspect$
- $VP_{transitive\ human} \rightarrow NP_{ins} + le \ / \ NP + den \ / \ Verb_{transitive\ human} + Aspect$
- $VP_{transitive\ human} \rightarrow NP_{ins} + le \ / \ NP + den \ / \ NP + i \ / \ Verb_{transitive\ human} + Aspect$
- $VP_{transitive\ human} \rightarrow NP + de \ / \ Verb_{transitive\ human} + Aspect$
- $VP_{transitive\ human} \rightarrow NP + de \ / \ NP + i \ / \ Verb_{transitive\ human} + Aspect$
- $VP_{transitive\ human} \rightarrow NP + de \ / \ NP + den \ / \ Verb_{transitive\ human} + Aspect$
- $VP_{transitive\ human} \rightarrow NP + de \ / \ NP + den \ / \ NP + i \ / \ Verb_{transitive\ human} + Aspect$
- $VP_{transitive\ human} \rightarrow NP + de \ / \ NP_{ins} + le \ / \ Verb_{transitive\ human} + Aspect$
- $VP_{transitive\ human} \rightarrow NP + de \ / \ NP_{ins} + le \ / \ NP + i \ / \ Verb_{transitive\ human} + Aspect$
- $VP_{transitive\ human} \rightarrow NP + de \ / \ NP_{ins} + le \ / \ NP + den \ / \ Verb_{transitive\ human} + Aspect$
- $VP_{transitive\ human} \rightarrow NP + de / NP_{ins} + le / NP + den / NP + i / Verb_{transitive\ human} + Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow Verb_{transitive\ mobile\ 1} + Aspect$

- $VP_{transitive\ mobile\ 1} \rightarrow NP + i\ /\ Verb_{transitive\ mobile\ 1} + Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP + e\ /\ Verb_{transitive\ mobile\ 1} + Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP + e\ /\ NP + i\ /\ Verb_{transitive\ mobile\ 1} + Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP + den\ /\ Verb_{transitive\ mobile\ 1} + Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP + den\ /\ NP + i\ /\ Verb_{transitive\ mobile\ 1} + Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP + den\ /\ NP + e\ /\ Verb_{transitive\ mobile\ 1} + Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP + den\ /\ NP + e\ /\ NP + i\ /\ Verb_{transitive\ mobile\ 1} + Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{ins} + le\ /\ Verb_{transitive\ mobile\ 1} + Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{ins} + le\ /\ NP + i\ /\ Verb_{transitive\ mobile\ 1} + Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{ins} + le\ /\ NP + e\ /\ Verb_{transitive\ mobile\ 1} + Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{ins} + le\ /\ NP + e\ /\ NP + i\ /\ Verb_{transitive\ mobile\ 1} + Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{ins} + le\ /\ NP + den\ /\ Verb_{transitive\ mobile\ 1} + Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{ins} + le\ /\ NP + den\ /\ NP + i\ /\ Verb_{transitive\ mobile\ 1} + Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ Verb_{transitive\ mobile\ 1} + Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{ins} + le / NP + den / NP + e / NP + i / Verb_{transitive\ mobile\ 1} + Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de\ /\ Verb_{transitive\ mobile\ 1} + Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP + i\ /\ Verb_{transitive\ mobile\ 1} + Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP + e\ /\ Verb_{transitive\ mobile\ 1} + Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP + e\ /\ NP + i\ /\ Verb_{transitive\ mobile\ 1} + Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP + den\ /\ Verb_{transitive\ mobile\ 1} + Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de / NP + den / NP + i / Verb_{transitive\ mobile\ 1} + Aspect$

- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time}+de/NP+den/NP+e/Verb_{transitive\ mobile\ 1}+Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP + den\ /\ NP + e\ /\ NP + i\ /\ Verb_{transitive\ mobile\ 1} + Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time}+de/NP_{ins}+le/Verb_{transitive\ mobile\ 1}+Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time}+de/NP_{ins}+le/NP+i/Verb_{transitive\ mobile\ 1}+Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time}+de/NP_{ins}+le/NP+e/Verb_{transitive\ mobile\ 1}+Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + e\ /\ NP + i\ /\ Verb_{transitive\ mobile\ 1} + Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time}+de/NP_{ins}+le/NP+den/Verb_{transitive\ mobile\ 1}+Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + den\ /\ NP + i\ /\ Verb_{transitive\ mobile\ 1} + Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ Verb_{transitive\ mobile\ 1} + Aspect$
- $VP_{transitive\ mobile\ 1} \rightarrow NP_{time}+de\ /\ NP_{ins}+le\ /\ NP+den\ /\ NP+e\ /\ NP+i\ /\ Verb_{transitive\ mobile\ 1} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{mob2} + i\ /\ Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP + e\ /\ Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP + e\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP + den\ /\ Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP+den\ /\ NP_{mob2}+i\ /\ Verb_{transitive\ mobile\ 2}+Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP + den\ /\ NP + e\ /\ Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP+den/NP+e/NP_{mob2}+i/Verb_{transitive\ mobile\ 2}+Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{ins} + le\ /\ Verb_{transitive\ mobile\ 2} + Aspect$

- $VP_{transitive\ mobile\ 2} \rightarrow NP_{ins} + le\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{ins} + le\ /\ NP + e\ /\ Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{ins} + le / NP + e / NP_{mob2} + i / Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{ins} + le\ /\ NP + den\ /\ Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{ins} + le / NP + den / NP_{mob2} + i / Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{ins} + le / NP + den / NP + e / Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de\ /\ Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de / NP_{mob2} + i / Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP + e\ /\ Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de / NP + e / NP_{mob2} + i / Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de / NP + den / Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de / NP + den / NP_{mob2} + i / Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de / NP + den / NP + e / Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP + den\ /\ NP + e\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de / NP_{ins} + le / Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de / NP_{ins} + le / NP_{mob2} + i / Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de / NP_{ins} + le / NP + e / Verb_{transitive\ mobile\ 2} + Aspect$

- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + e\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de / NP_{ins} + le / NP + den / Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + den\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ mobile\ 2} + Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow Verb_{transitive\ human\ mobile\ 1} + Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP + i\ /\ Verb_{transitive\ human\ mobile\ 1} + Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP + e\ /\ Verb_{transitive\ human\ mobile\ 1} + Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP + e\ /\ NP + i\ /\ Verb_{transitive\ human\ mobile\ 1} + Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP + den\ /\ Verb_{transitive\ human\ mobile\ 1} + Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP + den / NP + i / Verb_{transitive\ human\ mobile\ 1} + Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP + den / NP + e / Verb_{transitive\ human\ mobile\ 1} + Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP + den / NP + e / NP + i / Verb_{transitive\ human\ mobile\ 1} + Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{ins} + le\ /\ Verb_{transitive\ human\ mobile\ 1} + Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{ins} + le / NP + i / Verb_{transitive\ human\ mobile\ 1} + Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{ins} + le / NP + e / Verb_{transitive\ human\ mobile\ 1} + Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{ins} + le / NP + e / NP + i / Verb_{transitive\ human\ mobile\ 1} + Aspect$

- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{ins}$+$le/NP$+$den/Verb_{transitive\ human\ mobile\ 1}$+ $Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{ins}$+$le/NP$+$den/NP$+$i/Verb_{transitive\ human\ mobile\ 1}$+ $Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{ins}$+$le/NP$+$den/NP$+$e/Verb_{transitive\ human\ mobile\ 1}$+ $Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ NP +$ $i\ /\ Verb_{transitive\ human\ mobile\ 1} + Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time}$+$de/Verb_{transitive\ human\ mobile\ 1}$+$Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time}$+$de/NP$+$i/Verb_{transitive\ human\ mobile\ 1}$+ $Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time}$+$de/NP$+$e/Verb_{transitive\ human\ mobile\ 1}$+ $Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time}$+$de/NP$+$e/NP$+$i/Verb_{transitive\ human\ mobile\ 1}$+ $Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time}$+$de/NP$+$den/Verb_{transitive\ human\ mobile\ 1}$+ $Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time}$+$de/NP$+$den/NP$+$i/Verb_{transitive\ human\ mobile\ 1}$+ $Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time}$+$de/NP$+$den/NP$+$e/Verb_{transitive\ human\ mobile\ 1}$+ $Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP + den\ /\ NP + e\ /\ NP +$ $i\ /\ Verb_{transitive\ human\ mobile\ 1} + Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time}$+$de/NP_{ins}$+$le/Verb_{transitive\ human\ mobile\ 1}$+ $Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time}$+$de/NP_{ins}$+$le/NP$+$i/Verb_{transitive\ human\ mobile\ 1}$+ $Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time}$+$de/NP_{ins}$+$le/NP$+$e/Verb_{transitive\ human\ mobile\ 1}$+ $Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + e\ /\ NP +$ $i\ /\ Verb_{transitive\ human\ mobile\ 1} + Aspect$

- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time}+de/NP_{ins}+le/NP+den/Verb_{transitive\ human\ mobile\ 1}+Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + den\ /\ NP + i\ /\ Verb_{transitive\ human\ mobile\ 1} + Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ Verb_{transitive\ human\ mobile\ 1} + Aspect$
- $VP_{transitive\ human\ mobile\ 1} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ NP + i\ /\ Verb_{transitive\ human\ mobile\ 1} + Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow Verb_{transitive\ human\ mobile\ 2} + Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{mob2}+i\,/\,Verb_{transitive\ human\ mobile\ 2}+Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP + e\ /\ Verb_{transitive\ human\ mobile\ 2} + Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP+e/NP_{mob2}+i/Verb_{transitive\ human\ mobile\ 2}+Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP+den\ /\ Verb_{transitive\ human\ mobile\ 2}+Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP+den/NP_{mob2}+i/Verb_{transitive\ human\ mobile\ 2}+Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP+den/NP+e/Verb_{transitive\ human\ mobile\ 2}+Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP+den/NP+e/NP_{mob2}+i/Verb_{transitive\ human\ mobile\ 2}+Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{ins}+le\ /\ Verb_{transitive\ human\ mobile\ 2}+Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{ins}+le/NP_{mob2}+i/Verb_{transitive\ human\ mobile\ 2}+Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{ins}+le/NP+e/Verb_{transitive\ human\ mobile\ 2}+Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{ins}+le/NP+e/NP_{mob2}+i/Verb_{transitive\ human\ mobile\ 2}+Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{ins}+le/NP+den/Verb_{transitive\ human\ mobile\ 2}+Aspect$

- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{ins}$+$le/NP$+$den/NP_{mob2}$+$i/Verb_{transitive\ human\ mobile\ 2}$+ $Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{ins}$+$le/NP$+$den/NP$+$e/Verb_{transitive\ human\ mobile\ 2}$+ $Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ NP_{mob2} +$ $i\ /\ Verb_{transitive\ human\ mobile\ 2} + Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time}$+$de/Verb_{transitive\ human\ mobile\ 2}$+$Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time}$+$de/NP_{mob2}$+$i/Verb_{transitive\ human\ mobile\ 2}$+ $Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time}$+$de/NP$+$e/Verb_{transitive\ human\ mobile\ 2}$+ $Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time}$+$de/NP$+$e/NP_{mob2}$+$i/Verb_{transitive\ human\ mobile\ 2}$+ $Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time}$+$de/NP$+$den/Verb_{transitive\ human\ mobile\ 2}$+ $Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time}$+$de/NP$+$den/NP_{mob2}$+$i/Verb_{transitive\ human\ mobile\ 2}$+ $Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time}$+$de/NP$+$den/NP$+$e/Verb_{transitive\ human\ mobile\ 2}$+ $Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP + den\ /\ NP + e\ /\ NP_{mob2} +$ $i\ /\ Verb_{transitive\ human\ mobile\ 2} + Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time}$+$de/NP_{ins}$+$le/Verb_{transitive\ human\ mobile\ 2}$+ $Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time}$+$de/NP_{ins}$+$le/NP_{mob2}$+$i/Verb_{transitive\ human\ mobile\ 2}$+ $Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time}$+$de/NP_{ins}$+$le/NP$+$e/Verb_{transitive\ human\ mobile\ 2}$+ $Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + e\ /\ NP_{mob2} +$ $i\ /\ Verb_{transitive\ human\ mobile\ 2} + Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time}$+$de/NP_{ins}$+$le/NP$+$den/Verb_{transitive\ human\ mobile\ 2}$+ $Aspect$

- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + den\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ human\ mobile\ 2} + Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ Verb_{transitive\ human\ mobile\ 2} + Aspect$
- $VP_{transitive\ human\ mobile\ 2} \rightarrow NP_{time} + de\ /\ NP_{ins} + le\ /\ NP + den\ /\ NP + e\ /\ NP_{mob2} + i\ /\ Verb_{transitive\ human\ mobile\ 2} + Aspect$
- $Adv \rightarrow PostPosPhrase_f$
- $Adv \rightarrow PostPosPhrase_d$
- $Adv \rightarrow PostPosPhrase_e$
- $Adv \rightarrow PostPosPhrase_d\ /\ PostPosPhrase_f$
- $Adv \rightarrow PostPosPhrase_e\ /\ PostPosPhrase_f$
- $Adv \rightarrow PostPosPhrase_a$
- $Adv \rightarrow PostPosPhrase_a\ /\ PostPosPhrase_f$
- $Adv \rightarrow PostPosPhrase_a\ /\ PostPosPhrase_d$
- $Adv \rightarrow PostPosPhrase_a\ /\ PostPosPhrase_e$
- $Adv \rightarrow PostPosPhrase_a\ /\ PostPosPhrase_d\ /\ PostPosPhrase_f$
- $Adv \rightarrow PostPosPhrase_a\ /\ PostPosPhrase_e\ /\ PostPosPhrase_f$
- $Adv \rightarrow PostPosPhrase_b$
- $Adv \rightarrow PostPosPhrase_b\ /\ PostPosPhrase_f$
- $Adv \rightarrow PostPosPhrase_b\ /\ PostPosPhrase_d$
- $Adv \rightarrow PostPosPhrase_b\ /\ PostPosPhrase_e$
- $Adv \rightarrow PostPosPhrase_b\ /\ PostPosPhrase_d\ /\ PostPosPhrase_f$
- $Adv \rightarrow PostPosPhrase_b\ /\ PostPosPhrase_e\ /\ PostPosPhrase_f$
- $Adv \rightarrow PostPosPhrase_c$
- $Adv \rightarrow PostPosPhrase_c\ /\ PostPosPhrase_f$
- $Adv \rightarrow PostPosPhrase_c\ /\ PostPosPhrase_d$

- $Adv \rightarrow PostPosPhrase_c \ / \ PostPosPhrase_e$
- $Adv \rightarrow PostPosPhrase_c \ / \ PostPosPhrase_d \ / \ PostPosPhrase_f$
- $Adv \rightarrow PostPosPhrase_c \ / \ PostPosPhrase_e \ / \ PostPosPhrase_f$
- $Adv \rightarrow particle_a$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_f$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_d$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_e$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_d \ / \ PostPosPhrase_f$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_e \ / \ PostPosPhrase_f$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_a$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_a \ / \ PostPosPhrase_f$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_a \ / \ PostPosPhrase_d$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_a \ / \ PostPosPhrase_e$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_a \ / \ PostPosPhrase_d \ / \ PostPosPhrase_f$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_a \ / \ PostPosPhrase_e \ / \ PostPosPhrase_f$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_b$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_b \ / \ PostPosPhrase_f$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_b \ / \ PostPosPhrase_d$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_b \ / \ PostPosPhrase_e$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_b \ / \ PostPosPhrase_d \ / \ PostPosPhrase_f$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_b \ / \ PostPosPhrase_e \ / \ PostPosPhrase_f$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_c$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_c \ / \ PostPosPhrase_f$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_c \ / \ PostPosPhrase_d$
- $Adv \rightarrow particle_a \ / \ PostPosPhrase_a \ / \ PostPosPhrase_e$

- $Adv \rightarrow particle_a$ / $PostPosPhrase_c$ / $PostPosPhrase_d$ / $PostPosPhrase_f$
- $Adv \rightarrow particle_a$ / $PostPosPhrase_c$ / $PostPosPhrase_e$ / $PostPosPhrase_f$
- $Adv \rightarrow particle_b$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_f$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_d$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_e$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_d$ / $PostPosPhrase_f$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_e$ / $PostPosPhrase_f$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_a$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_a$ / $PostPosPhrase_f$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_a$ / $PostPosPhrase_d$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_a$ / $PostPosPhrase_e$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_a$ / $PostPosPhrase_d$ / $PostPosPhrase_f$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_a$ / $PostPosPhrase_e$ / $PostPosPhrase_f$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_b$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_b$ / $PostPosPhrase_f$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_b$ / $PostPosPhrase_d$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_b$ / $PostPosPhrase_e$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_b$ / $PostPosPhrase_d$ / $PostPosPhrase_f$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_b$ / $PostPosPhrase_e$ / $PostPosPhrase_f$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_c$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_c$ / $PostPosPhrase_f$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_c$ / $PostPosPhrase_d$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_c$ / $PostPosPhrase_e$
- $Adv \rightarrow particle_b$ / $PostPosPhrase_c$ / $PostPosPhrase_d$ / $PostPosPhrase_f$

- $Adv \rightarrow particle_b$ / $PostPosPhrase_c$ / $PostPosPhrase_e$ / $PostPosPhrase_f$
- $PostPosPhrase_a \rightarrow postposition_a$
- $PostPosPhrase_a \rightarrow NP + den \,\&\, postposition_a$
- $PostPosPhrase_b \rightarrow NP + den \,\&\, postposition_b$
- $PostPosPhrase_c \rightarrow NP + den \,\&\, postposition_c$
- $PostPosPhrase_d \rightarrow postposition_d$
- $PostPosPhrase_d \rightarrow NP + e \,\&\, postposition_d$
- $PostPosPhrase_e \rightarrow PostPosPhrase_e$
- $PostPosPhrase_f \rightarrow NP \,\&\, postposition_f$
- $NP_{time} \rightarrow n_{time\,a}$
- $NP_{time} \rightarrow n_{time\,b}$
- $NP_{abstract} \rightarrow n_{abstract}$
- $NP_{abstract} \rightarrow demonstrative \,\&\, n_{abstract}$
- $NP_{ins} \rightarrow n_{item}$
- $NP_{ins} \rightarrow n_{local}$
- $NP_{ins} \rightarrow n_{local\,proper}$
- $NP_{ins} \rightarrow n_{number}$
- $NP_{ins} \rightarrow n_{substance}$
- $NP_{ins} \rightarrow n_{inanimate}$
- $NP_{ins} \rightarrow n_{animate\,transport}$
- $NP_{ins} \rightarrow n_{number} \,\&\, n_{item}$
- $NP_{ins} \rightarrow n_{number} \,\&\, n_{local}$
- $NP_{ins} \rightarrow n_{number} \,\&\, n_{local\,proper}$
- $NP_{ins} \rightarrow n_{number} \,\&\, n_{number}$
- $NP_{ins} \rightarrow n_{number} \,\&\, n_{substance}$

- $NP_{ins} \rightarrow n_{number}$ & $n_{inanimate}$
- $NP_{ins} \rightarrow n_{number}$ & $n_{animate\ transport}$
- $NP_{ins} \rightarrow demonstrative$ & n_{item}
- $NP_{ins} \rightarrow demonstrative$ & n_{local}
- $NP_{ins} \rightarrow demonstrative$ & $n_{local\ proper}$
- $NP_{ins} \rightarrow demonstrative$ & n_{number}
- $NP_{ins} \rightarrow demonstrative$ & $n_{substance}$
- $NP_{ins} \rightarrow demonstrative$ & $n_{inanimate}$
- $NP_{ins} \rightarrow demonstrative$ & $n_{animate\ transport}$
- $NP_{ins} \rightarrow demonstrative$ & n_{number} & n_{item}
- $NP_{ins} \rightarrow demonstrative$ & n_{number} & n_{local}
- $NP_{ins} \rightarrow demonstrative$ & n_{number} & $n_{local\ proper}$
- $NP_{ins} \rightarrow demonstrative$ & n_{number} & n_{number}
- $NP_{ins} \rightarrow demonstrative$ & n_{number} & $n_{substance}$
- $NP_{ins} \rightarrow demonstrative$ & n_{number} & $n_{inanimate}$
- $NP_{ins} \rightarrow demonstrative$ & n_{number} & $n_{animate\ transport}$
- $NP_{mob2} \rightarrow n_{item}$
- $NP_{mob2} \rightarrow n_{local}$
- $NP_{mob2} \rightarrow n_{local\ proper}$
- $NP_{mob2} \rightarrow n_{number}$
- $NP_{mob2} \rightarrow n_{substance}$
- $NP_{mob2} \rightarrow n_{inanimate}$
- $NP_{mob2} \rightarrow n_{number}$ & n_{item}
- $NP_{mob2} \rightarrow n_{number}$ & n_{local}
- $NP_{mob2} \rightarrow n_{number}$ & $n_{local\ proper}$

- $NP_{mob2} \rightarrow n_{number}$ & n_{number}
- $NP_{mob2} \rightarrow n_{number}$ & $n_{substance}$
- $NP_{mob2} \rightarrow n_{number}$ & $n_{inanimate}$
- $NP_{mob2} \rightarrow demonstrative$ & n_{item}
- $NP_{mob2} \rightarrow demonstrative$ & n_{local}
- $NP_{mob2} \rightarrow demonstrative$ & $n_{local\ proper}$
- $NP_{mob2} \rightarrow demonstrative$ & n_{number}
- $NP_{mob2} \rightarrow demonstrative$ & $n_{substance}$
- $NP_{mob2} \rightarrow demonstrative$ & $n_{inanimate}$
- $NP_{mob2} \rightarrow demonstrative$ & n_{number} & n_{item}
- $NP_{mob2} \rightarrow demonstrative$ & n_{number} & n_{local}
- $NP_{mob2} \rightarrow demonstrative$ & n_{number} & $n_{local\ proper}$
- $NP_{mob2} \rightarrow demonstrative$ & n_{number} & n_{number}
- $NP_{mob2} \rightarrow demonstrative$ & n_{number} & $n_{substance}$
- $NP_{mob2} \rightarrow demonstrative$ & n_{number} & $n_{inanimate}$
- $NP_{human} \rightarrow n_{animate\ human}$
- $NP_{human} \rightarrow n_{animate\ proper\ human}$
- $NP_{human} \rightarrow n_{number}$ & $n_{animate\ human}$
- $NP_{human} \rightarrow n_{number}$ & $n_{animate\ proper\ human}$
- $NP_{human} \rightarrow demonstrative$ & $n_{animate\ human}$
- $NP_{human} \rightarrow demonstrative$ & $n_{animate\ proper\ human}$
- $NP_{human} \rightarrow demonstrative$ & n_{number} & $n_{animate\ human}$
- $NP_{human} \rightarrow demonstrative$ & n_{number} & $n_{animate\ proper\ human}$
- $NP_{human} \rightarrow ben$
- $NP_{human} \rightarrow sen$

- $NP_{human} \rightarrow o$
- $NP_{mob} \rightarrow n_{animate}$
- $NP_{mob} \rightarrow n_{animate\ proper}$
- $NP_{mob} \rightarrow n_{animate\ transport}$
- $NP_{mob} \rightarrow n_{inanimate}$
- $NP_{mob} \rightarrow n_{number}\ \&\ n_{animate}$
- $NP_{mob} \rightarrow n_{number}\ \&\ n_{animate\ proper}$
- $NP_{mob} \rightarrow n_{number}\ \&\ n_{animate\ transport}$
- $NP_{mob} \rightarrow n_{number}\ \&\ n_{inanimate}$
- $NP_{mob} \rightarrow demonstrative\ \&\ n_{animate}$
- $NP_{mob} \rightarrow demonstrative\ \&\ n_{animate\ proper}$
- $NP_{mob} \rightarrow demonstrative\ \&\ n_{animate\ transport}$
- $NP_{mob} \rightarrow demonstrative\ \&\ n_{inanimate}$
- $NP_{mob} \rightarrow demonstrative\ \&\ n_{number}\ \&\ n_{animate}$
- $NP_{mob} \rightarrow demonstrative\ \&\ n_{number}\ \&\ n_{animate\ proper}$
- $NP_{mob} \rightarrow demonstrative\ \&\ n_{number}\ \&\ n_{animate\ transport}$
- $NP_{mob} \rightarrow demonstrative\ \&\ n_{number}\ \&\ n_{inanimate}$
- $NP_{mob} \rightarrow pronoun$
- $NP \rightarrow pronoun$
- $NP \rightarrow bura$
- $NP \rightarrow ura$
- $NP \rightarrow ora$
- $NP \rightarrow nere$
- $NP \rightarrow n_{time\ a}$
- $NP \rightarrow n_{time\ b}$

- $NP \rightarrow n_{abstract}$
- $NP \rightarrow demonstrative \ \& \ n_{abstract}$
- $NP \rightarrow n_{animate}$
- $NP \rightarrow n_{animate\ human}$
- $NP \rightarrow n_{animate\ proper}$
- $NP \rightarrow n_{animate\ proper\ human}$
- $NP \rightarrow n_{animate\ transport}$
- $NP \rightarrow n_{inanimate}$
- $NP \rightarrow n_{item}$
- $NP \rightarrow n_{locale}$
- $NP \rightarrow n_{locale\ proper}$
- $NP \rightarrow n_{number}$
- $NP \rightarrow n_{substance}$
- $NP \rightarrow n_{number} \ \& \ n_{animate}$
- $NP \rightarrow n_{number} \ \& \ n_{animate\ human}$
- $NP \rightarrow n_{number} \ \& \ n_{animate\ proper}$
- $NP \rightarrow n_{number} \ \& \ n_{animate\ proper\ human}$
- $NP \rightarrow n_{number} \ \& \ n_{animate\ transport}$
- $NP \rightarrow n_{number} \ \& \ n_{inanimate}$
- $NP \rightarrow n_{number} \ \& \ n_{item}$
- $NP \rightarrow n_{number} \ \& \ n_{locale}$
- $NP \rightarrow n_{number} \ \& \ n_{locale\ proper}$
- $NP \rightarrow n_{number} \ \& \ n_{number}$
- $NP \rightarrow n_{number} \ \& \ n_{substance}$
- $NP \rightarrow demonstrative \ \& \ n_{animate}$

- $NP \rightarrow demonstrative \;\&\; n_{animate\,human}$
- $NP \rightarrow demonstrative \;\&\; n_{animate\,proper}$
- $NP \rightarrow demonstrative \;\&\; n_{animate\,proper\,human}$
- $NP \rightarrow demonstrative \;\&\; n_{animate\,transport}$
- $NP \rightarrow demonstrative \;\&\; n_{inanimate}$
- $NP \rightarrow demonstrative \;\&\; n_{item}$
- $NP \rightarrow demonstrative \;\&\; n_{locale}$
- $NP \rightarrow demonstrative \;\&\; n_{locale\,proper}$
- $NP \rightarrow demonstrative \;\&\; n_{number}$
- $NP \rightarrow demonstrative \;\&\; n_{substance}$
- $NP \rightarrow demonstrative \;\&\; n_{number} \;\&\; n_{animate}$
- $NP \rightarrow demonstrative \;\&\; n_{number} \;\&\; n_{animate\,human}$
- $NP \rightarrow demonstrative \;\&\; n_{number} \;\&\; n_{animate\,proper}$
- $NP \rightarrow demonstrative \;\&\; n_{number} \;\&\; n_{animate\,proper\,human}$
- $NP \rightarrow demonstrative \;\&\; n_{number} \;\&\; n_{animate\,transport}$
- $NP \rightarrow demonstrative \;\&\; n_{number} \;\&\; n_{inanimate}$
- $NP \rightarrow demonstrative \;\&\; n_{number} \;\&\; n_{item}$
- $NP \rightarrow demonstrative \;\&\; n_{number} \;\&\; n_{locale}$
- $NP \rightarrow demonstrative \;\&\; n_{number} \;\&\; n_{locale\,proper}$
- $NP \rightarrow demonstrative \;\&\; n_{number} \;\&\; n_{number}$
- $NP \rightarrow demonstrative \;\&\; n_{number} \;\&\; n_{substance}$

9 Appendix D

9.1 grammarall.txt

This appendix is listing of definite clause grammar of Turkish, implemented in Prolog. Groups indicate rule generation sessions in section 3.3. Implementation is done using Trinc-Prolog development tool which supports ISO Prolog standards and DCGs.

```
/*1*/ s(s(VP_TREE)) --> vp(VP_TREE).

/*2*/ s(s(VP_TREE)) --> vp(VP_TREE),[x],[mi].

/*3*/ s(s(ADV_TREE,VP_TREE)) --> adv(ADV_TREE),[x],vp(VP_TREE).

/*4*/ s(s(ADV_TREE,VP_TREE)) --> adv(ADV_TREE),[x],vp(VP_TREE),[x],[mi].

/*5*/ s(s(NP_TREE,VP_N_TREE)) --> np(NP_TREE),[x],vp_n(VP_N_TREE).

/*6*/ s(s(NP_TREE,VP_N_TREE)) --> np(NP_TREE),[x],vp_n(VP_N_TREE),[x],[mi
           ].

/*7*/ s(s(NP_TREE,ADV_TREE,VP_N_TREE)) --> np(NP_TREE),[x],adv(ADV_TREE),
           [x],vp_n(VP_N_TREE).

/*8*/ s(s(NP_TREE,ADV_TREE,VP_N_TREE)) --> np(NP_TREE),[x],adv(ADV_TREE),
           [x],vp_n(VP_N_TREE),[x],[mi].

/*9*/ s(s(NP_HUMAN_TREE,VP_HUMAN_TREE)) --> np_human(NP_HUMAN_TREE),[x],v
           p_human(VP_HUMAN_TREE).

/*10*/ s(s(NP_HUMAN_TREE,VP_HUMAN_TREE)) --> np_human(NP_HUMAN_TREE),[x],v
           p_human(VP_HUMAN_TREE),[x],[mi].

/*11*/ s(s(NP_HUMAN_TREE,ADV_TREE,VP_HUMAN_TREE)) --> np_human(NP_HUMAN_TR
           EE),[x],adv(ADV_TREE),[x],vp_human(VP_HUMAN_TREE).

/*12*/ s(s(NP_HUMAN_TREE,ADV_TREE,VP_HUMAN_TREE)) --> np_human(NP_HUMAN_TR
           EE),[x],adv(ADV_TREE),[x],vp_human(VP_HUMAN_TREE),[x],[mi].
```

```
/*13*/ s(s(NP_MOB_TREE,VP_MOB_TREE)) --> np_mob(NP_MOB_TREE),[x],vp_mob(VP
           _MOB_TREE).

/*14*/ s(s(NP_MOB_TREE,VP_MOB_TREE)) --> np_mob(NP_MOB_TREE),[x],vp_mob(VP
           _MOB_TREE),[x],[mi].

/*15*/ s(s(NP_MOB_TREE,ADV_TREE,VP_MOB_TREE)) --> np_mob(NP_MOB_TREE),[x],
           adv(ADV_TREE),[x],vp_mob(VP_MOB_TREE).

/*16*/ s(s(NP_MOB_TREE,ADV_TREE,VP_MOB_TREE)) --> np_mob(NP_MOB_TREE),[x],
           adv(ADV_TREE),[x],vp_mob(VP_MOB_TREE),[x],[mi].

/*17*/ vp_n(vp_n(ADJUNCTADJ_A_TREE)) --> adjunctadj_a(ADJUNCTADJ_A_TREE).

/*18*/ vp_n(vp_n(ADJUNCTADJ_B_TREE)) --> adjunctadj_b(ADJUNCTADJ_B_TREE).

/*19*/ vp_n(vp_n(ADJUNCTADJ_C_TREE)) --> adjunctadj_c(ADJUNCTADJ_C_TREE).

/*20*/ vp_n(vp_n(ADJUNCTNP_TREE)) --> adjunctnp(ADJUNCTNP_TREE).

/*21*/ vp_n(vp_n(VP_IMPERSONAL_TREE)) --> vp_impersonal(VP_IMPERSONAL_TREE
           ).

/*22*/ vp_n(vp_n(VP_NON_TREE)) --> vp_non(VP_NON_TREE).

/*23*/ vp_n(vp_n(VP_TRANSITIVE_TREE)) --> vp_transitive(VP_TRANSITIVE_TREE
           ).

/*24*/ vp_human(vp_human(VP_TRANSITIVE_HUMAN_TREE)) --> vp_transitive_huma
           n(VP_TRANSITIVE_HUMAN_TREE).

/*25*/ vp_mob(vp_mob(VP_MOBILE_A_TREE)) --> vp_mobile_a(VP_MOBILE_A_TREE).


/*26*/ vp_mob(vp_mob(VP_MOBILE_B_TREE)) --> vp_mobile_b(VP_MOBILE_B_TREE).


/*27*/ vp_mob(vp_mob(VP_MOBILE_TREE)) --> vp_mobile(VP_MOBILE_TREE).

/*28*/ vp_mob(vp_mob(VP_TRANSITIVE_MOBILE_1_TREE)) --> vp_transitive_mobil
           e_1(VP_TRANSITIVE_MOBILE_1_TREE).
```

```
/*29*/ vp_mob(vp_mob(VP_TRANSITIVE_MOBILE_2_TREE)) --> vp_transitive_mobil
          e_2(VP_TRANSITIVE_MOBILE_2_TREE).

/*30*/ vp_human(vp_human(VP_TRANSITIVE_HUMAN_MOBILE_1_TREE)) --> vp_transi
          tive_human_mobile_1(VP_TRANSITIVE_HUMAN_MOBILE_1_TREE).

/*31*/ vp_human(vp_human(VP_TRANSITIVE_HUMAN_MOBILE_2_TREE)) --> vp_transi
          tive_human_mobile_2(VP_TRANSITIVE_HUMAN_MOBILE_2_TREE).

/*32*/ vp(vp(VP_N_TREE)) --> vp_n(VP_N_TREE).

/*33*/ vp(vp(VP_HUMAN_TREE)) --> vp_human(VP_HUMAN_TREE).

/*34*/ vp(vp(VP_MOB_TREE)) --> vp_mob(VP_MOB_TREE).

/*35*/ aspect(aspect(er)) --> [er].

/*36*/ aspect(aspect(ecek)) --> [ecek].

/*37*/ aspect(aspect(iyor)) --> [iyor].

/*38*/ aspect(aspect(meli)) --> [meli].

/*39*/ aspect(aspect(mi)) --> [mi].

/*40*/ adjunctnp(adjunctnp(NP_TREE,COPULA_TREE)) --> np(NP_TREE),copula(CO
          PULA_TREE).

/*41*/ adjunctnp(adjunctnp(NP_TREE,NP_TREE,COPULA_TREE)) --> np(NP_TREE),[
          e],[x],np(NP_TREE),copula(COPULA_TREE).

/*42*/ adjunctnp(adjunctnp(NP_TREE,NP_TREE,COPULA_TREE)) --> np(NP_TREE),[
          de],[x],np(NP_TREE),copula(COPULA_TREE).

/*43*/ adjunctnp(adjunctnp(NP_TREE,NP_TREE,NP_TREE,COPULA_TREE)) --> np(NP
          _TREE),[de],[x],np(NP_TREE),[e],[x],np(NP_TREE),copula(COPULA_T
          REE).

/*44*/ adjunctadj_a(adjunctadj_a(ADJ_A_TREE,COPULA_TREE)) --> adj_a(ADJ_A_
          TREE),copula(COPULA_TREE).
```

```
/*45*/ adjunctadj_a(adjunctadj_a(NP_TREE,ADJ_A_TREE,COPULA_TREE)) --> np(N
           P_TREE),[e],[x],adj_a(ADJ_A_TREE),copula(COPULA_TREE).

/*46*/ adjunctadj_a(adjunctadj_a(NP_TREE,ADJ_A_TREE,COPULA_TREE)) --> np(N
           P_TREE),[de],[x],adj_a(ADJ_A_TREE),copula(COPULA_TREE).

/*47*/ adjunctadj_a(adjunctadj_a(NP_TREE,NP_TREE,ADJ_A_TREE,COPULA_TREE))
           --> np(NP_TREE),[de],[x],np(NP_TREE),[e],[x],adj_a(ADJ_A_TREE),
           copula(COPULA_TREE).

/*48*/ adjunctadj_b(adjunctadj_b(ADJ_B_TREE,COPULA_TREE)) --> adj_b(ADJ_B_
           TREE),copula(COPULA_TREE).

/*49*/ adjunctadj_b(adjunctadj_b(NP_TREE,ADJ_B_TREE,COPULA_TREE)) --> np(N
           P_TREE),[e],[x],adj_b(ADJ_B_TREE),copula(COPULA_TREE).

/*50*/ adjunctadj_b(adjunctadj_b(NP_TREE,ADJ_B_TREE,COPULA_TREE)) --> np(N
           P_TREE),[de],[x],adj_b(ADJ_B_TREE),copula(COPULA_TREE).

/*51*/ adjunctadj_b(adjunctadj_b(NP_TREE,NP_TREE,ADJ_B_TREE,COPULA_TREE))
           --> np(NP_TREE),[de],[x],np(NP_TREE),[e],[x],adj_b(ADJ_B_TREE),
           copula(COPULA_TREE).

/*52*/ adjunctadj_c(adjunctadj_c(ADJ_C_TREE,COPULA_TREE)) --> adj_c(ADJ_C_
           TREE),copula(COPULA_TREE).

/*53*/ adjunctadj_c(adjunctadj_c(NP_TREE,ADJ_C_TREE,COPULA_TREE)) --> np(N
           P_TREE),[e],[x],adj_c(ADJ_C_TREE),copula(COPULA_TREE).

/*54*/ adjunctadj_c(adjunctadj_c(NP_TREE,ADJ_C_TREE,COPULA_TREE)) --> np(N
           P_TREE),[de],[x],adj_c(ADJ_C_TREE),copula(COPULA_TREE).

/*55*/ adjunctadj_c(adjunctadj_c(NP_TREE,NP_TREE,ADJ_C_TREE,COPULA_TREE))
           --> np(NP_TREE),[de],[x],np(NP_TREE),[e],[x],adj_c(ADJ_C_TREE),
           copula(COPULA_TREE).

/*56*/ adjunctadj_c(adjunctadj_c(NP_TREE,ADJ_C_TREE,COPULA_TREE)) --> np(N
           P_TREE),[den],[x],adj_c(ADJ_C_TREE),copula(COPULA_TREE).

/*57*/ adjunctadj_c(adjunctadj_c(NP_TREE,NP_TREE,ADJ_C_TREE,COPULA_TREE))
           --> np(NP_TREE),[de],[x],np(NP_TREE),[den],adj_c(ADJ_C_TREE),co
           pula(COPULA_TREE).
```

```
/*58*/ adjunctadj_d(adjunctadj_d(ADJ_D_TREE,COPULA_TREE)) --> adj_d(ADJ_D_
           TREE),copula(COPULA_TREE).

/*59*/ adjunctadj_d(adjunctadj_d(NP_INS_TREE,ADJ_D_TREE,COPULA_TREE)) -->
           np_ins(NP_INS_TREE),[le],[x],adj_d(ADJ_D_TREE),copula(COPULA_TR
           EE).

/*60*/ adjunctadj_d(adjunctadj_d(NP_TREE,ADJ_D_TREE,COPULA_TREE)) --> np(N
           P_TREE),[de],[x],adj_d(ADJ_D_TREE),copula(COPULA_TREE).

/*61*/ adjunctadj_d(adjunctadj_d(NP_TREE,NP_INS_TREE)) --> np(NP_TREE),[de
           ],[x],np_ins(NP_INS_TREE),[le].

/*62*/ vp_impersonal(vp_impersonal(VERB_IMPERSONAL_TREE,ASPECT_TREE)) -->
           verb_impersonal(VERB_IMPERSONAL_TREE),aspect(ASPECT_TREE).

/*63*/ vp_impersonal(vp_impersonal(NP_TREE,VERB_IMPERSONAL_TREE,ASPECT_TRE
           E)) --> np(NP_TREE),[e],[x],verb_impersonal(VERB_IMPERSONAL_TRE
           E),aspect(ASPECT_TREE).

/*64*/ vp_impersonal(vp_impersonal(NP_TREE,VERB_IMPERSONAL_TREE,ASPECT_TRE
           E)) --> np(NP_TREE),[den],[x],verb_impersonal(VERB_IMPERSONAL_T
           REE),aspect(ASPECT_TREE).

/*65*/ vp_impersonal(vp_impersonal(NP_TREE,NP_TREE,VERB_IMPERSONAL_TREE,AS
           PECT_TREE)) --> np(NP_TREE),[den],[x],np(NP_TREE),[e],[x],verb_
           impersonal(VERB_IMPERSONAL_TREE),aspect(ASPECT_TREE).

/*66*/ vp_impersonal(vp_impersonal(NP_ABSTRACT_TREE,VERB_IMPERSONAL_TREE,A
           SPECT_TREE)) --> np_abstract(NP_ABSTRACT_TREE),[le],[x],verb_im
           personal(VERB_IMPERSONAL_TREE),aspect(ASPECT_TREE).

/*67*/ vp_impersonal(vp_impersonal(NP_ABSTRACT_TREE,NP_TREE,VERB_IMPERSONA
           L_TREE,ASPECT_TREE)) --> np_abstract(NP_ABSTRACT_TREE),[le],[x]
           ,np(NP_TREE),[e],[x],verb_impersonal(VERB_IMPERSONAL_TREE),aspe
           ct(ASPECT_TREE).

/*68*/ vp_impersonal(vp_impersonal(NP_ABSTRACT_TREE,NP_TREE,VERB_IMPERSONA
           L_TREE,ASPECT_TREE)) --> np_abstract(NP_ABSTRACT_TREE),[le],[x]
           ,np(NP_TREE),[den],[x],verb_impersonal(VERB_IMPERSONAL_TREE),as
           pect(ASPECT_TREE).
```

```
/*69*/ vp_impersonal(vp_impersonal(NP_ABSTRACT_TREE,NP_TREE,NP_TREE,VERB_I
           MPERSONAL_TREE,ASPECT_TREE)) --> np_abstract(NP_ABSTRACT_TREE),
           [le],[x],np(NP_TREE),[den],[x],np(NP_TREE),[e],[x],verb_imperso
           nal(VERB_IMPERSONAL_TREE),aspect(ASPECT_TREE).

/*70*/ vp_impersonal(vp_impersonal(NP_TREE,VERB_IMPERSONAL_TREE,ASPECT_TRE
           E)) --> np(NP_TREE),[de],[x],verb_impersonal(VERB_IMPERSONAL_TR
           EE),aspect(ASPECT_TREE).

/*71*/ vp_impersonal(vp_impersonal(NP_TREE,NP_TREE,VERB_IMPERSONAL_TREE,AS
           PECT_TREE)) --> np(NP_TREE),[de],[x],np(NP_TREE),[e],[x],verb_i
           mpersonal(VERB_IMPERSONAL_TREE),aspect(ASPECT_TREE).

/*72*/ vp_impersonal(vp_impersonal(NP_TREE,NP_TREE,VERB_IMPERSONAL_TREE,AS
           PECT_TREE)) --> np(NP_TREE),[de],[x],np(NP_TREE),[den],[x],verb
           _impersonal(VERB_IMPERSONAL_TREE),aspect(ASPECT_TREE).

/*73*/ vp_impersonal(vp_impersonal(NP_TREE,NP_TREE,NP_TREE,VERB_IMPERSONAL
           _TREE,ASPECT_TREE)) --> np(NP_TREE),[de],[x],np(NP_TREE),[den],
           [x],np(NP_TREE),[e],[x],verb_impersonal(VERB_IMPERSONAL_TREE),a
           spect(ASPECT_TREE).

/*74*/ vp_impersonal(vp_impersonal(NP_TREE,NP_ABSTRACT_TREE,VERB_IMPERSONA
           L_TREE,ASPECT_TREE)) --> np(NP_TREE),[de],[x],np_abstract(NP_AB
           STRACT_TREE),[le],[x],verb_impersonal(VERB_IMPERSONAL_TREE),asp
           ect(ASPECT_TREE).

/*75*/ vp_impersonal(vp_impersonal(NP_TREE,NP_ABSTRACT_TREE,NP_TREE,VERB_I
           MPERSONAL_TREE,ASPECT_TREE)) --> np(NP_TREE),[de],[x],np_abstra
           ct(NP_ABSTRACT_TREE),[le],[x],np(NP_TREE),[e],[x],verb_imperson
           al(VERB_IMPERSONAL_TREE),aspect(ASPECT_TREE).

/*76*/ vp_impersonal(vp_impersonal(NP_TREE,NP_ABSTRACT_TREE,NP_TREE,VERB_I
           MPERSONAL_TREE,ASPECT_TREE)) --> np(NP_TREE),[de],[x],np_abstra
           ct(NP_ABSTRACT_TREE),[le],[x],np(NP_TREE),[den],[x],verb_impers
           onal(VERB_IMPERSONAL_TREE),aspect(ASPECT_TREE).

/*77*/ vp_impersonal(vp_impersonal(NP_TREE,NP_ABSTRACT_TREE,NP_TREE,NP_TRE
           E,VERB_IMPERSONAL_TREE,ASPECT_TREE)) --> np(NP_TREE),[de],[x],n
           p_abstract(NP_ABSTRACT_TREE),[le],[x],np(NP_TREE),[den],[x],np(
           NP_TREE),[e],[x],verb_impersonal(VERB_IMPERSONAL_TREE),aspect(A
           SPECT_TREE).
```

```
/*78*/ vp_mobile(vp_mobile(VERB_MOBILE_TREE,ASPECT_TREE)) --> verb_mobile(
          VERB_MOBILE_TREE),aspect(ASPECT_TREE).

/*79*/ vp_mobile(vp_mobile(NP_TREE,VERB_MOBILE_TREE,ASPECT_TREE)) --> np(N
          P_TREE),[e],[x],verb_mobile(VERB_MOBILE_TREE),aspect(ASPECT_TRE
          E).

/*80*/ vp_mobile(vp_mobile(NP_TREE,VERB_MOBILE_TREE,ASPECT_TREE)) --> np(N
          P_TREE),[den],[x],verb_mobile(VERB_MOBILE_TREE),aspect(ASPECT_T
          REE).

/*81*/ vp_mobile(vp_mobile(NP_TREE,NP_TREE,VERB_MOBILE_TREE,ASPECT_TREE))
          --> np(NP_TREE),[den],[x],np(NP_TREE),[e],[x],verb_mobile(VERB_
          MOBILE_TREE),aspect(ASPECT_TREE).

/*82*/ vp_mobile(vp_mobile(NP_INS_TREE,VERB_MOBILE_TREE,ASPECT_TREE)) -->
          np_ins(NP_INS_TREE),[le],[x],verb_mobile(VERB_MOBILE_TREE),aspe
          ct(ASPECT_TREE).

/*83*/ vp_mobile(vp_mobile(NP_INS_TREE,NP_TREE,VERB_MOBILE_TREE,ASPECT_TRE
          E)) --> np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[e],[x],verb_m
          obile(VERB_MOBILE_TREE),aspect(ASPECT_TREE).

/*84*/ vp_mobile(vp_mobile(NP_INS_TREE,NP_TREE,VERB_MOBILE_TREE,ASPECT_TRE
          E)) --> np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[den],[x],verb
          _mobile(VERB_MOBILE_TREE),aspect(ASPECT_TREE).

/*85*/ vp_mobile(vp_mobile(NP_INS_TREE,NP_TREE,NP_TREE,VERB_MOBILE_TREE,AS
          PECT_TREE)) --> np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[den],
          [x],np(NP_TREE),[e],[x],verb_mobile(VERB_MOBILE_TREE),aspect(AS
          PECT_TREE).

/*86*/ vp_mobile(vp_mobile(NP_TIME_TREE,VERB_MOBILE_TREE,ASPECT_TREE)) -->
           np_time(NP_TIME_TREE),[de],[x],verb_mobile(VERB_MOBILE_TREE),a
          spect(ASPECT_TREE).

/*87*/ vp_mobile(vp_mobile(NP_TIME_TREE,NP_TREE,VERB_MOBILE_TREE,ASPECT_TR
          EE)) --> np_time(NP_TIME_TREE),[de],[x],np(NP_TREE),[e],[x],ver
          b_mobile(VERB_MOBILE_TREE),aspect(ASPECT_TREE).

/*88*/ vp_mobile(vp_mobile(NP_TIME_TREE,NP_TREE,VERB_MOBILE_TREE,ASPECT_TR
          EE)) --> np_time(NP_TIME_TREE),[de],[x],np(NP_TREE),[den],[x],v
          erb_mobile(VERB_MOBILE_TREE),aspect(ASPECT_TREE).
```

```
/*89*/ vp_mobile(vp_mobile(NP_TIME_TREE,NP_TREE,NP_TREE,VERB_MOBILE_TREE,A
           SPECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np(NP_TREE),[de
           n],[x],np(NP_TREE),[e],[x],verb_mobile(VERB_MOBILE_TREE),aspect
           (ASPECT_TREE).

/*90*/ vp_mobile(vp_mobile(NP_TIME_TREE,NP_INS_TREE,VERB_MOBILE_TREE,ASPEC
           T_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np_ins(NP_INS_TREE)
           ,[le],[x],verb_mobile(VERB_MOBILE_TREE),aspect(ASPECT_TREE).

/*91*/ vp_mobile(vp_mobile(NP_TIME_TREE,NP_INS_TREE,NP_TREE,VERB_MOBILE_TR
           EE,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np_ins(NP_I
           NS_TREE),[le],[x],np(NP_TREE),[e],[x],verb_mobile(VERB_MOBILE_T
           REE),aspect(ASPECT_TREE).

/*92*/ vp_mobile(vp_mobile(NP_TIME_TREE,NP_INS_TREE,NP_TREE,VERB_MOBILE_TR
           EE,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np_ins(NP_I
           NS_TREE),[le],[x],np(NP_TREE),[den],[x],verb_mobile(VERB_MOBILE
           _TREE),aspect(ASPECT_TREE).

/*93*/ vp_mobile(vp_mobile(NP_TIME_TREE,NP_INS_TREE,NP_TREE,NP_TREE,VERB_M
           OBILE_TREE,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np_
           ins(NP_INS_TREE),[le],[x],np(NP_TREE),[den],[x],np(NP_TREE),[e]
           ,[x],verb_mobile(VERB_MOBILE_TREE),aspect(ASPECT_TREE).

/*94*/ vp_mobile_a(vp_mobile_a(VERB_MOBILE_A_TREE,ASPECT_TREE)) --> verb_m
           obile_a(VERB_MOBILE_A_TREE),aspect(ASPECT_TREE).

/*95*/ vp_mobile_a(vp_mobile_a(NP_TREE,VERB_MOBILE_A_TREE,ASPECT_TREE)) --
           > np(NP_TREE),[e],[x],verb_mobile_a(VERB_MOBILE_A_TREE),aspect(
           ASPECT_TREE).

/*96*/ vp_mobile_a(vp_mobile_a(NP_TREE,VERB_MOBILE_A_TREE,ASPECT_TREE)) --
           > np(NP_TREE),[den],[x],verb_mobile_a(VERB_MOBILE_A_TREE),aspec
           t(ASPECT_TREE).

/*97*/ vp_mobile_a(vp_mobile_a(NP_TREE,NP_TREE,VERB_MOBILE_A_TREE,ASPECT_T
           REE)) --> np(NP_TREE),[den],[x],np(NP_TREE),[e],[x],verb_mobile
           _a(VERB_MOBILE_A_TREE),aspect(ASPECT_TREE).

/*98*/ vp_mobile_a(vp_mobile_a(NP_INS_TREE,VERB_MOBILE_A_TREE,ASPECT_TREE)
           ) --> np_ins(NP_INS_TREE),[le],[x],verb_mobile_a(VERB_MOBILE_A_
           TREE),aspect(ASPECT_TREE).
```

```
/*99*/ vp_mobile_a(vp_mobile_a(NP_INS_TREE,NP_TREE,VERB_MOBILE_A_TREE,ASPE
           CT_TREE)) --> np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[e],[x],
           verb_mobile_a(VERB_MOBILE_A_TREE),aspect(ASPECT_TREE).

/*100*/ vp_mobile_a(vp_mobile_a(NP_INS_TREE,NP_TREE,VERB_MOBILE_A_TREE,ASPE
           CT_TREE)) --> np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[den],[x
           ],verb_mobile_a(VERB_MOBILE_A_TREE),aspect(ASPECT_TREE).

/*101*/ vp_mobile_a(vp_mobile_a(NP_INS_TREE,NP_TREE,NP_TREE,VERB_MOBILE_A_T
           REE,ASPECT_TREE)) --> np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),
           [den],[x],np(NP_TREE),[e],[x],verb_mobile_a(VERB_MOBILE_A_TREE)
           ,aspect(ASPECT_TREE).

/*102*/ vp_mobile_a(vp_mobile_a(NP_TIME_TREE,VERB_MOBILE_A_TREE,ASPECT_TREE
           )) --> np_time(NP_TIME_TREE),[de],[x],verb_mobile_a(VERB_MOBILE
           _A_TREE),aspect(ASPECT_TREE).

/*103*/ vp_mobile_a(vp_mobile_a(NP_TIME_TREE,NP_TREE,VERB_MOBILE_A_TREE,ASP
           ECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np(NP_TREE),[e],[
           x],verb_mobile_a(VERB_MOBILE_A_TREE),aspect(ASPECT_TREE).

/*104*/ vp_mobile_a(vp_mobile_a(NP_TIME_TREE,NP_TREE,VERB_MOBILE_A_TREE,ASP
           ECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np(NP_TREE),[den]
           ,[x],verb_mobile_a(VERB_MOBILE_A_TREE),aspect(ASPECT_TREE).

/*105*/ vp_mobile_a(vp_mobile_a(NP_TIME_TREE,NP_TREE,NP_TREE,VERB_MOBILE_A_
           TREE,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np(NP_TRE
           E),[den],[x],np(NP_TREE),[e],[x],verb_mobile_a(VERB_MOBILE_A_TR
           EE),aspect(ASPECT_TREE).

/*106*/ vp_mobile_a(vp_mobile_a(NP_TIME_TREE,NP_INS_TREE,VERB_MOBILE_A_TREE
           ,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np_ins(NP_INS
           _TREE),[le],[x],verb_mobile_a(VERB_MOBILE_A_TREE),aspect(ASPECT
           _TREE).

/*107*/ vp_mobile_a(vp_mobile_a(NP_TIME_TREE,NP_INS_TREE,NP_TREE,VERB_MOBIL
           E_A_TREE,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np_in
           s(NP_INS_TREE),[le],[x],np(NP_TREE),[e],[x],verb_mobile_a(VERB_
           MOBILE_A_TREE),aspect(ASPECT_TREE).
```

```
/*108*/ vp_mobile_a(vp_mobile_a(NP_TIME_TREE,NP_INS_TREE,NP_TREE,VERB_MOBIL
            E_A_TREE,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np_in
            s(NP_INS_TREE),[le],[x],np(NP_TREE),[den],[x],verb_mobile_a(VER
            B_MOBILE_A_TREE),aspect(ASPECT_TREE).

/*109*/ vp_mobile_a(vp_mobile_a(NP_TIME_TREE,NP_INS_TREE,NP_TREE,NP_TREE,VE
            RB_MOBILE_A_TREE,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[de],[
            x],np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[den],[x],np(NP_TRE
            E),[e],[x],verb_mobile_a(VERB_MOBILE_A_TREE),aspect(ASPECT_TREE
            ).

/*110*/ vp_mobile_b(vp_mobile_b(VERB_MOBILE_B_TREE,ASPECT_TREE)) --> verb_m
            obile_b(VERB_MOBILE_B_TREE),aspect(ASPECT_TREE).

/*111*/ vp_mobile_b(vp_mobile_b(NP_TREE,VERB_MOBILE_B_TREE,ASPECT_TREE)) --
            > np(NP_TREE),[e],[x],verb_mobile_b(VERB_MOBILE_B_TREE),aspect(
            ASPECT_TREE).

/*112*/ vp_mobile_b(vp_mobile_b(NP_TREE,VERB_MOBILE_B_TREE,ASPECT_TREE)) --
            > np(NP_TREE),[den],[x],verb_mobile_b(VERB_MOBILE_B_TREE),aspec
            t(ASPECT_TREE).

/*113*/ vp_mobile_b(vp_mobile_b(NP_TREE,NP_TREE,VERB_MOBILE_B_TREE,ASPECT_T
            REE)) --> np(NP_TREE),[den],[x],np(NP_TREE),[e],[x],verb_mobile
            _b(VERB_MOBILE_B_TREE),aspect(ASPECT_TREE).

/*114*/ vp_mobile_b(vp_mobile_b(NP_INS_TREE,VERB_MOBILE_B_TREE,ASPECT_TREE)
            ) --> np_ins(NP_INS_TREE),[le],[x],verb_mobile_b(VERB_MOBILE_B_
            TREE),aspect(ASPECT_TREE).

/*115*/ vp_mobile_b(vp_mobile_b(NP_INS_TREE,NP_TREE,VERB_MOBILE_B_TREE,ASPE
            CT_TREE)) --> np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[e],[x],
            verb_mobile_b(VERB_MOBILE_B_TREE),aspect(ASPECT_TREE).

/*116*/ vp_mobile_b(vp_mobile_b(NP_INS_TREE,NP_TREE,VERB_MOBILE_B_TREE,ASPE
            CT_TREE)) --> np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[den],[x
            ],verb_mobile_b(VERB_MOBILE_B_TREE),aspect(ASPECT_TREE).

/*117*/ vp_mobile_b(vp_mobile_b(NP_INS_TREE,NP_TREE,NP_TREE,VERB_MOBILE_B_T
            REE,ASPECT_TREE)) --> np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),
            [den],[x],np(NP_TREE),[e],[x],verb_mobile_b(VERB_MOBILE_B_TREE)
            ,aspect(ASPECT_TREE).
```

```
/*118*/ vp_mobile_b(vp_mobile_b(NP_TIME_TREE,VERB_MOBILE_B_TREE,ASPECT_TREE
            )) --> np_time(NP_TIME_TREE),[de],[x],verb_mobile_b(VERB_MOBILE
            _B_TREE),aspect(ASPECT_TREE).

/*119*/ vp_mobile_b(vp_mobile_b(NP_TIME_TREE,NP_TREE,VERB_MOBILE_B_TREE,ASP
            ECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np(NP_TREE),[e],[
            x],verb_mobile_b(VERB_MOBILE_B_TREE),aspect(ASPECT_TREE).

/*120*/ vp_mobile_b(vp_mobile_b(NP_TIME_TREE,NP_TREE,VERB_MOBILE_B_TREE,ASP
            ECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np(NP_TREE),[den]
            ,[x],verb_mobile_b(VERB_MOBILE_B_TREE),aspect(ASPECT_TREE).

/*121*/ vp_mobile_b(vp_mobile_b(NP_TIME_TREE,NP_TREE,NP_TREE,VERB_MOBILE_B_
            TREE,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np(NP_TRE
            E),[den],[x],np(NP_TREE),[e],[x],verb_mobile_b(VERB_MOBILE_B_TR
            EE),aspect(ASPECT_TREE).

/*122*/ vp_mobile_b(vp_mobile_b(NP_TIME_TREE,NP_INS_TREE,VERB_MOBILE_B_TREE
            ,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np_ins(NP_INS
            _TREE),[le],[x],verb_mobile_b(VERB_MOBILE_B_TREE),aspect(ASPECT
            _TREE).

/*123*/ vp_mobile_b(vp_mobile_b(NP_TIME_TREE,NP_INS_TREE,NP_TREE,VERB_MOBIL
            E_B_TREE,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np_in
            s(NP_INS_TREE),[le],[x],np(NP_TREE),[e],[x],verb_mobile_b(VERB_
            MOBILE_B_TREE),aspect(ASPECT_TREE).

/*124*/ vp_mobile_b(vp_mobile_b(NP_TIME_TREE,NP_INS_TREE,NP_TREE,VERB_MOBIL
            E_B_TREE,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np_in
            s(NP_INS_TREE),[le],[x],np(NP_TREE),[den],[x],verb_mobile_b(VER
            B_MOBILE_B_TREE),aspect(ASPECT_TREE).

/*125*/ vp_mobile_b(vp_mobile_b(NP_TIME_TREE,NP_INS_TREE,NP_TREE,NP_TREE,VE
            RB_MOBILE_B_TREE,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[de],[
            x],np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[den],[x],np(NP_TRE
            E),[e],[x],verb_mobile_b(VERB_MOBILE_B_TREE),aspect(ASPECT_TREE
            ).

/*126*/ vp_non(vp_non(VERB_TREE,ASPECT_TREE)) --> verb(VERB_TREE),aspect(AS
            PECT_TREE).
```

```
/*127*/ vp_non(vp_non(NP_TREE,VERB_TREE,ASPECT_TREE)) --> np(NP_TREE),[den]
            ,[x],verb(VERB_TREE),aspect(ASPECT_TREE).

/*128*/ vp_non(vp_non(NP_ABSTRACT_TREE,VERB_TREE,ASPECT_TREE)) --> np_abstr
            act(NP_ABSTRACT_TREE),[le],[x],verb(VERB_TREE),aspect(ASPECT_TR
            EE).

/*129*/ vp_non(vp_non(NP_ABSTRACT_TREE,NP_TREE,VERB_TREE,ASPECT_TREE)) -->
            np_abstract(NP_ABSTRACT_TREE),[le],[x],np(NP_TREE),[den],[x],ve
            rb(VERB_TREE),aspect(ASPECT_TREE).

/*130*/ vp_non(vp_non(NP_TREE,VERB_TREE,ASPECT_TREE)) --> np(NP_TREE),[de],
            [x],verb(VERB_TREE),aspect(ASPECT_TREE).

/*131*/ vp_non(vp_non(NP_TREE,NP_TREE,VERB_TREE,ASPECT_TREE)) --> np(NP_TRE
            E),[de],[x],np(NP_TREE),[den],[x],verb(VERB_TREE),aspect(ASPECT
            _TREE).

/*132*/ vp_non(vp_non(NP_TREE,NP_ABSTRACT_TREE,VERB_TREE,ASPECT_TREE)) -->
            np(NP_TREE),[de],[x],np_abstract(NP_ABSTRACT_TREE),[le],[x],ver
            b(VERB_TREE),aspect(ASPECT_TREE).

/*133*/ vp_non(vp_non(NP_TREE,NP_ABSTRACT_TREE,NP_TREE,VERB_TREE,ASPECT_TRE
            E)) --> np(NP_TREE),[de],[x],np_abstract(NP_ABSTRACT_TREE),[le]
            ,[x],np(NP_TREE),[den],[x],verb(VERB_TREE),aspect(ASPECT_TREE).


/*134*/ vp_transitive(vp_transitive(VERB_TRANSITIVE_TREE,ASPECT_TREE)) -->
            verb_transitive(VERB_TRANSITIVE_TREE),aspect(ASPECT_TREE).

/*135*/ vp_transitive(vp_transitive(NP_TREE,VERB_TRANSITIVE_TREE,ASPECT_TRE
            E)) --> np(NP_TREE),[i],[x],verb_transitive(VERB_TRANSITIVE_TRE
            E),aspect(ASPECT_TREE).

/*136*/ vp_transitive(vp_transitive(NP_TREE,VERB_TRANSITIVE_TREE,ASPECT_TRE
            E)) --> np(NP_TREE),[den],[x],verb_transitive(VERB_TRANSITIVE_T
            REE),aspect(ASPECT_TREE).

/*137*/ vp_transitive(vp_transitive(NP_TREE,NP_TREE,VERB_TRANSITIVE_TREE,AS
            PECT_TREE)) --> np(NP_TREE),[den],[x],np(NP_TREE),[i],[x],verb_
            transitive(VERB_TRANSITIVE_TREE),aspect(ASPECT_TREE).
```

```
/*138*/ vp_transitive(vp_transitive(NP_INS_TREE,VERB_TRANSITIVE_TREE,ASPECT
           _TREE)) --> np_ins(NP_INS_TREE),[le],[x],verb_transitive(VERB_T
           RANSITIVE_TREE),aspect(ASPECT_TREE).

/*139*/ vp_transitive(vp_transitive(NP_INS_TREE,NP_TREE,VERB_TRANSITIVE_TRE
           E,ASPECT_TREE)) --> np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[i
           ],[x],verb_transitive(VERB_TRANSITIVE_TREE),aspect(ASPECT_TREE)
           .

/*140*/ vp_transitive(vp_transitive(NP_INS_TREE,NP_TREE,VERB_TRANSITIVE_TRE
           E,ASPECT_TREE)) --> np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[d
           en],[x],verb_transitive(VERB_TRANSITIVE_TREE),aspect(ASPECT_TRE
           E).

/*141*/ vp_transitive(vp_transitive(NP_INS_TREE,NP_TREE,NP_TREE,VERB_TRANSI
           TIVE_TREE,ASPECT_TREE)) --> np_ins(NP_INS_TREE),[le],[x],np(NP_
           TREE),[den],[x],np(NP_TREE),[i],[x],verb_transitive(VERB_TRANSI
           TIVE_TREE),aspect(ASPECT_TREE).

/*142*/ vp_transitive(vp_transitive(NP_TREE,VERB_TRANSITIVE_TREE,ASPECT_TRE
           E)) --> np(NP_TREE),[de],[x],verb_transitive(VERB_TRANSITIVE_TR
           EE),aspect(ASPECT_TREE).

/*143*/ vp_transitive(vp_transitive(NP_TREE,NP_TREE,VERB_TRANSITIVE_TREE,AS
           PECT_TREE)) --> np(NP_TREE),[de],[x],np(NP_TREE),[i],[x],verb_t
           ransitive(VERB_TRANSITIVE_TREE),aspect(ASPECT_TREE).

/*144*/ vp_transitive(vp_transitive(NP_TREE,NP_TREE,VERB_TRANSITIVE_TREE,AS
           PECT_TREE)) --> np(NP_TREE),[de],[x],np(NP_TREE),[den],[x],verb
           _transitive(VERB_TRANSITIVE_TREE),aspect(ASPECT_TREE).

/*145*/ vp_transitive(vp_transitive(NP_TREE,NP_TREE,NP_TREE,VERB_TRANSITIVE
           _TREE,ASPECT_TREE)) --> np(NP_TREE),[de],[x],np(NP_TREE),[den],
           [x],np(NP_TREE),[i],[x],verb_transitive(VERB_TRANSITIVE_TREE),a
           spect(ASPECT_TREE).

/*146*/ vp_transitive(vp_transitive(NP_TREE,NP_INS_TREE,VERB_TRANSITIVE_TRE
           E,ASPECT_TREE)) --> np(NP_TREE),[de],[x],np_ins(NP_INS_TREE),[l
           e],[x],verb_transitive(VERB_TRANSITIVE_TREE),aspect(ASPECT_TREE
           ).
```

```
/*147*/ vp_transitive(vp_transitive(NP_TREE,NP_INS_TREE,NP_TREE,VERB_TRANSI
           TIVE_TREE,ASPECT_TREE)) --> np(NP_TREE),[de],[x],np_ins(NP_INS_
           TREE),[le],[x],np(NP_TREE),[i],[x],verb_transitive(VERB_TRANSIT
           IVE_TREE),aspect(ASPECT_TREE).

/*148*/ vp_transitive(vp_transitive(NP_TREE,NP_INS_TREE,NP_TREE,VERB_TRANSI
           TIVE_TREE,ASPECT_TREE)) --> np(NP_TREE),[de],[x],np_ins(NP_INS_
           TREE),[le],[x],np(NP_TREE),[den],[x],verb_transitive(VERB_TRANS
           ITIVE_TREE),aspect(ASPECT_TREE).

/*149*/ vp_transitive(vp_transitive(NP_TREE,NP_INS_TREE,NP_TREE,NP_TREE,VER
           B_TRANSITIVE_TREE,ASPECT_TREE)) --> np(NP_TREE),[de],[x],np_ins
           (NP_INS_TREE),[le],[x],np(NP_TREE),[den],[x],np(NP_TREE),[i],[x
           ],verb_transitive(VERB_TRANSITIVE_TREE),aspect(ASPECT_TREE).

/*150*/ vp_transitive_human(vp_transitive_human(VERB_TRANSITIVE_HUMAN_TREE,
           ASPECT_TREE)) --> verb_transitive_human(VERB_TRANSITIVE_HUMAN_T
           REE),aspect(ASPECT_TREE).

/*151*/ vp_transitive_human(vp_transitive_human(NP_TREE,VERB_TRANSITIVE_HUM
           AN_TREE,ASPECT_TREE)) --> np(NP_TREE),[i],[x],verb_transitive_h
           uman(VERB_TRANSITIVE_HUMAN_TREE),aspect(ASPECT_TREE).

/*152*/ vp_transitive_human(vp_transitive_human(NP_TREE,VERB_TRANSITIVE_HUM
           AN_TREE,ASPECT_TREE)) --> np(NP_TREE),[den],[x],verb_transitive
           _human(VERB_TRANSITIVE_HUMAN_TREE),aspect(ASPECT_TREE).

/*153*/ vp_transitive_human(vp_transitive_human(NP_TREE,NP_TREE,VERB_TRANSI
           TIVE_HUMAN_TREE,ASPECT_TREE)) --> np(NP_TREE),[den],[x],np(NP_T
           REE),[i],[x],verb_transitive_human(VERB_TRANSITIVE_HUMAN_TREE),
           aspect(ASPECT_TREE).

/*154*/ vp_transitive_human(vp_transitive_human(NP_INS_TREE,VERB_TRANSITIVE
           _HUMAN_TREE,ASPECT_TREE)) --> np_ins(NP_INS_TREE),[le],[x],verb
           _transitive_human(VERB_TRANSITIVE_HUMAN_TREE),aspect(ASPECT_TRE
           E).

/*155*/ vp_transitive_human(vp_transitive_human(NP_INS_TREE,NP_TREE,VERB_TR
           ANSITIVE_HUMAN_TREE,ASPECT_TREE)) --> np_ins(NP_INS_TREE),[le],
           [x],np(NP_TREE),[i],[x],verb_transitive_human(VERB_TRANSITIVE_H
           UMAN_TREE),aspect(ASPECT_TREE).
```

```
/*156*/ vp_transitive_human(vp_transitive_human(NP_INS_TREE,NP_TREE,VERB_TR
            ANSITIVE_HUMAN_TREE,ASPECT_TREE)) --> np_ins(NP_INS_TREE),[le],
            [x],np(NP_TREE),[den],[x],verb_transitive_human(VERB_TRANSITIVE
            _HUMAN_TREE),aspect(ASPECT_TREE).

/*157*/ vp_transitive_human(vp_transitive_human(NP_INS_TREE,NP_TREE,NP_TREE
            ,VERB_TRANSITIVE_HUMAN_TREE,ASPECT_TREE)) --> np_ins(NP_INS_TRE
            E),[le],[x],np(NP_TREE),[den],[x],np(NP_TREE),[i],[x],verb_tran
            sitive_human(VERB_TRANSITIVE_HUMAN_TREE),aspect(ASPECT_TREE).

/*158*/ vp_transitive_human(vp_transitive_human(NP_TREE,VERB_TRANSITIVE_HUM
            AN_TREE,ASPECT_TREE)) --> np(NP_TREE),[de],[x],verb_transitive_
            human(VERB_TRANSITIVE_HUMAN_TREE),aspect(ASPECT_TREE).

/*159*/ vp_transitive_human(vp_transitive_human(NP_TREE,NP_TREE,VERB_TRANSI
            TIVE_HUMAN_TREE,ASPECT_TREE)) --> np(NP_TREE),[de],[x],np(NP_TR
            EE),[i],[x],verb_transitive_human(VERB_TRANSITIVE_HUMAN_TREE),a
            spect(ASPECT_TREE).

/*160*/ vp_transitive_human(vp_transitive_human(NP_TREE,NP_TREE,VERB_TRANSI
            TIVE_HUMAN_TREE,ASPECT_TREE)) --> np(NP_TREE),[de],[x],np(NP_TR
            EE),[den],[x],verb_transitive_human(VERB_TRANSITIVE_HUMAN_TREE)
            ,aspect(ASPECT_TREE).

/*161*/ vp_transitive_human(vp_transitive_human(NP_TREE,NP_TREE,NP_TREE,VER
            B_TRANSITIVE_HUMAN_TREE,ASPECT_TREE)) --> np(NP_TREE),[de],[x],
            np(NP_TREE),[den],[x],np(NP_TREE),[i],[x],verb_transitive_human
            (VERB_TRANSITIVE_HUMAN_TREE),aspect(ASPECT_TREE).

/*162*/ vp_transitive_human(vp_transitive_human(NP_TREE,NP_INS_TREE,VERB_TR
            ANSITIVE_HUMAN_TREE,ASPECT_TREE)) --> np(NP_TREE),[de],[x],np_i
            ns(NP_INS_TREE),[le],[x],verb_transitive_human(VERB_TRANSITIVE_
            HUMAN_TREE),aspect(ASPECT_TREE).

/*163*/ vp_transitive_human(vp_transitive_human(NP_TREE,NP_INS_TREE,NP_TREE
            ,VERB_TRANSITIVE_HUMAN_TREE,ASPECT_TREE)) --> np(NP_TREE),[de],
            [x],np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[i],[x],verb_trans
            itive_human(VERB_TRANSITIVE_HUMAN_TREE),aspect(ASPECT_TREE).

/*164*/ vp_transitive_human(vp_transitive_human(NP_TREE,NP_INS_TREE,NP_TREE
            ,VERB_TRANSITIVE_HUMAN_TREE,ASPECT_TREE)) --> np(NP_TREE),[de],
            [x],np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[den],[x],verb_tra
            nsitive_human(VERB_TRANSITIVE_HUMAN_TREE),aspect(ASPECT_TREE).
```

```
/*165*/ vp_transitive_human(vp_transitive_human(NP_TREE,NP_INS_TREE,NP_TREE
            ,NP_TREE,VERB_TRANSITIVE_HUMAN_TREE,ASPECT_TREE)) --> np(NP_TRE
            E),[de],[x],np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[den],[x],
            np(NP_TREE),[i],[x],verb_transitive_human(VERB_TRANSITIVE_HUMAN
            _TREE),aspect(ASPECT_TREE).

/*166*/ vp_transitive_mobile_1(vp_transitive_mobile_1(VERB_TRANSITIVE_MOBIL
            E_1_TREE,ASPECT_TREE)) --> verb_transitive_mobile_1(VERB_TRANSI
            TIVE_MOBILE_1_TREE),aspect(ASPECT_TREE).

/*167*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_TREE,VERB_TRANSITI
            VE_MOBILE_1_TREE,ASPECT_TREE)) --> np(NP_TREE),[i],[x],verb_tra
            nsitive_mobile_1(VERB_TRANSITIVE_MOBILE_1_TREE),aspect(ASPECT_T
            REE).

/*168*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_TREE,VERB_TRANSITI
            VE_MOBILE_1_TREE,ASPECT_TREE)) --> np(NP_TREE),[e],[x],verb_tra
            nsitive_mobile_1(VERB_TRANSITIVE_MOBILE_1_TREE),aspect(ASPECT_T
            REE).

/*169*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_TREE,NP_TREE,VERB_
            TRANSITIVE_MOBILE_1_TREE,ASPECT_TREE)) --> np(NP_TREE),[e],[x],
            np(NP_TREE),[i],[x],verb_transitive_mobile_1(VERB_TRANSITIVE_MO
            BILE_1_TREE),aspect(ASPECT_TREE).

/*170*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_TREE,VERB_TRANSITI
            VE_MOBILE_1_TREE,ASPECT_TREE)) --> np(NP_TREE),[den],[x],verb_t
            ransitive_mobile_1(VERB_TRANSITIVE_MOBILE_1_TREE),aspect(ASPECT
            _TREE).

/*171*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_TREE,NP_TREE,VERB_
            TRANSITIVE_MOBILE_1_TREE,ASPECT_TREE)) --> np(NP_TREE),[den],[x
            ],np(NP_TREE),[i],[x],verb_transitive_mobile_1(VERB_TRANSITIVE_
            MOBILE_1_TREE),aspect(ASPECT_TREE).

/*172*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_TREE,NP_TREE,VERB_
            TRANSITIVE_MOBILE_1_TREE,ASPECT_TREE)) --> np(NP_TREE),[den],[x
            ],np(NP_TREE),[e],[x],verb_transitive_mobile_1(VERB_TRANSITIVE_
            MOBILE_1_TREE),aspect(ASPECT_TREE).
```

```
/*173*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_TREE,NP_TREE,NP_TR
            EE,VERB_TRANSITIVE_MOBILE_1_TREE,ASPECT_TREE)) --> np(NP_TREE),
            [den],[x],np(NP_TREE),[e],[x],np(NP_TREE),[i],[x],verb_transiti
            ve_mobile_1(VERB_TRANSITIVE_MOBILE_1_TREE),aspect(ASPECT_TREE).


/*174*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_INS_TREE,VERB_TRAN
            SITIVE_MOBILE_1_TREE,ASPECT_TREE)) --> np_ins(NP_INS_TREE),[le]
            ,[x],verb_transitive_mobile_1(VERB_TRANSITIVE_MOBILE_1_TREE),as
            pect(ASPECT_TREE).

/*175*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_INS_TREE,NP_TREE,V
            ERB_TRANSITIVE_MOBILE_1_TREE,ASPECT_TREE)) --> np_ins(NP_INS_TR
            EE),[le],[x],np(NP_TREE),[i],[x],verb_transitive_mobile_1(VERB_
            TRANSITIVE_MOBILE_1_TREE),aspect(ASPECT_TREE).

/*176*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_INS_TREE,NP_TREE,V
            ERB_TRANSITIVE_MOBILE_1_TREE,ASPECT_TREE)) --> np_ins(NP_INS_TR
            EE),[le],[x],np(NP_TREE),[e],[x],verb_transitive_mobile_1(VERB_
            TRANSITIVE_MOBILE_1_TREE),aspect(ASPECT_TREE).

/*177*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_INS_TREE,NP_TREE,N
            P_TREE,VERB_TRANSITIVE_MOBILE_1_TREE,ASPECT_TREE)) --> np_ins(N
            P_INS_TREE),[le],[x],np(NP_TREE),[e],[x],np(NP_TREE),[i],[x],ve
            rb_transitive_mobile_1(VERB_TRANSITIVE_MOBILE_1_TREE),aspect(AS
            PECT_TREE).

/*178*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_INS_TREE,NP_TREE,V
            ERB_TRANSITIVE_MOBILE_1_TREE,ASPECT_TREE)) --> np_ins(NP_INS_TR
            EE),[le],[x],np(NP_TREE),[den],[x],verb_transitive_mobile_1(VER
            B_TRANSITIVE_MOBILE_1_TREE),aspect(ASPECT_TREE).

/*179*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_INS_TREE,NP_TREE,N
            P_TREE,VERB_TRANSITIVE_MOBILE_1_TREE,ASPECT_TREE)) --> np_ins(N
            P_INS_TREE),[le],[x],np(NP_TREE),[den],[x],np(NP_TREE),[i],[x],
            verb_transitive_mobile_1(VERB_TRANSITIVE_MOBILE_1_TREE),aspect(
            ASPECT_TREE).

/*180*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_INS_TREE,NP_TREE,N
            P_TREE,VERB_TRANSITIVE_MOBILE_1_TREE,ASPECT_TREE)) --> np_ins(N
            P_INS_TREE),[le],[x],np(NP_TREE),[den],[x],np(NP_TREE),[e],[x],
            verb_transitive_mobile_1(VERB_TRANSITIVE_MOBILE_1_TREE),aspect(
            ASPECT_TREE).
```

```
/*181*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_INS_TREE,NP_TREE,N
            P_TREE,NP_TREE,VERB_TRANSITIVE_MOBILE_1_TREE,ASPECT_TREE)) -->
            np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[den],[x],np(NP_TREE),
            [e],[x],np(NP_TREE),[i],[x],verb_transitive_mobile_1(VERB_TRANS
            ITIVE_MOBILE_1_TREE),aspect(ASPECT_TREE).

/*182*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_TIME_TREE,VERB_TRA
            NSITIVE_MOBILE_1_TREE,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[
            de],[x],verb_transitive_mobile_1(VERB_TRANSITIVE_MOBILE_1_TREE)
            ,aspect(ASPECT_TREE).

/*183*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_TIME_TREE,NP_TREE,
            VERB_TRANSITIVE_MOBILE_1_TREE,ASPECT_TREE)) --> np_time(NP_TIME
            _TREE),[de],[x],np(NP_TREE),[i],[x],verb_transitive_mobile_1(VE
            RB_TRANSITIVE_MOBILE_1_TREE),aspect(ASPECT_TREE).

/*184*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_TIME_TREE,NP_TREE,
            VERB_TRANSITIVE_MOBILE_1_TREE,ASPECT_TREE)) --> np_time(NP_TIME
            _TREE),[de],[x],np(NP_TREE),[e],[x],verb_transitive_mobile_1(VE
            RB_TRANSITIVE_MOBILE_1_TREE),aspect(ASPECT_TREE).

/*185*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_TIME_TREE,NP_TREE,
            NP_TREE,VERB_TRANSITIVE_MOBILE_1_TREE,ASPECT_TREE)) --> np_time
            (NP_TIME_TREE),[de],[x],np(NP_TREE),[e],[x],np(NP_TREE),[i],[x]
            ,verb_transitive_mobile_1(VERB_TRANSITIVE_MOBILE_1_TREE),aspect
            (ASPECT_TREE).

/*186*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_TIME_TREE,NP_TREE,
            VERB_TRANSITIVE_MOBILE_1_TREE,ASPECT_TREE)) --> np_time(NP_TIME
            _TREE),[de],[x],np(NP_TREE),[den],[x],verb_transitive_mobile_1(
            VERB_TRANSITIVE_MOBILE_1_TREE),aspect(ASPECT_TREE).

/*187*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_TIME_TREE,NP_TREE,
            NP_TREE,VERB_TRANSITIVE_MOBILE_1_TREE,ASPECT_TREE)) --> np_time
            (NP_TIME_TREE),[de],[x],np(NP_TREE),[den],[x],np(NP_TREE),[i],[
            x],verb_transitive_mobile_1(VERB_TRANSITIVE_MOBILE_1_TREE),aspe
            ct(ASPECT_TREE).

/*188*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_TIME_TREE,NP_TREE,
            NP_TREE,VERB_TRANSITIVE_MOBILE_1_TREE,ASPECT_TREE)) --> np_time
            (NP_TIME_TREE),[de],[x],np(NP_TREE),[den],[x],np(NP_TREE),[e],[
            x],verb_transitive_mobile_1(VERB_TRANSITIVE_MOBILE_1_TREE),aspe
            ct(ASPECT_TREE).
```

```
/*189*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_TIME_TREE,NP_TREE,
            NP_TREE,NP_TREE,VERB_TRANSITIVE_MOBILE_1_TREE,ASPECT_TREE)) -->
             np_time(NP_TIME_TREE),[de],[x],np(NP_TREE),[den],[x],np(NP_TRE
            E),[e],[x],np(NP_TREE),[i],[x],verb_transitive_mobile_1(VERB_TR
            ANSITIVE_MOBILE_1_TREE),aspect(ASPECT_TREE).

/*190*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_TIME_TREE,NP_INS_T
            REE,VERB_TRANSITIVE_MOBILE_1_TREE,ASPECT_TREE)) --> np_time(NP_
            TIME_TREE),[de],[x],np_ins(NP_INS_TREE),[le],[x],verb_transitiv
            e_mobile_1(VERB_TRANSITIVE_MOBILE_1_TREE),aspect(ASPECT_TREE).

/*191*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_TIME_TREE,NP_INS_T
            REE,NP_TREE,VERB_TRANSITIVE_MOBILE_1_TREE,ASPECT_TREE)) --> np_
            time(NP_TIME_TREE),[de],[x],np_ins(NP_INS_TREE),[le],[x],np(NP_
            TREE),[i],[x],verb_transitive_mobile_1(VERB_TRANSITIVE_MOBILE_1
            _TREE),aspect(ASPECT_TREE).

/*192*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_TIME_TREE,NP_INS_T
            REE,NP_TREE,VERB_TRANSITIVE_MOBILE_1_TREE,ASPECT_TREE)) --> np_
            time(NP_TIME_TREE),[de],[x],np_ins(NP_INS_TREE),[le],[x],np(NP_
            TREE),[e],[x],verb_transitive_mobile_1(VERB_TRANSITIVE_MOBILE_1
            _TREE),aspect(ASPECT_TREE).

/*193*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_TIME_TREE,NP_INS_T
            REE,NP_TREE,NP_TREE,VERB_TRANSITIVE_MOBILE_1_TREE,ASPECT_TREE))
             --> np_time(NP_TIME_TREE),[de],[x],np_ins(NP_INS_TREE),[le],[x
            ],np(NP_TREE),[e],[x],np(NP_TREE),[i],[x],verb_transitive_mobil
            e_1(VERB_TRANSITIVE_MOBILE_1_TREE),aspect(ASPECT_TREE).

/*194*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_TIME_TREE,NP_INS_T
            REE,NP_TREE,VERB_TRANSITIVE_MOBILE_1_TREE,ASPECT_TREE)) --> np_
            time(NP_TIME_TREE),[de],[x],np_ins(NP_INS_TREE),[le],[x],np(NP_
            TREE),[den],[x],verb_transitive_mobile_1(VERB_TRANSITIVE_MOBILE
            _1_TREE),aspect(ASPECT_TREE).

/*195*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_TIME_TREE,NP_INS_T
            REE,NP_TREE,NP_TREE,VERB_TRANSITIVE_MOBILE_1_TREE,ASPECT_TREE))
             --> np_time(NP_TIME_TREE),[de],[x],np_ins(NP_INS_TREE),[le],[x
            ],np(NP_TREE),[den],[x],np(NP_TREE),[i],[x],verb_transitive_mob
            ile_1(VERB_TRANSITIVE_MOBILE_1_TREE),aspect(ASPECT_TREE).
```

```
/*196*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_TIME_TREE,NP_INS_T
            REE,NP_TREE,NP_TREE,VERB_TRANSITIVE_MOBILE_1_TREE,ASPECT_TREE))
             --> np_time(NP_TIME_TREE),[de],[x],np_ins(NP_INS_TREE),[le],[x
            ],np(NP_TREE),[den],[x],np(NP_TREE),[e],[x],verb_transitive_mob
            ile_1(VERB_TRANSITIVE_MOBILE_1_TREE),aspect(ASPECT_TREE).

/*197*/ vp_transitive_mobile_1(vp_transitive_mobile_1(NP_TIME_TREE,NP_INS_T
            REE,NP_TREE,NP_TREE,NP_TREE,VERB_TRANSITIVE_MOBILE_1_TREE,ASPEC
            T_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np_ins(NP_INS_TREE)
            ,[le],[x],np(NP_TREE),[den],[x],np(NP_TREE),[e],[x],np(NP_TREE)
            ,[i],[x],verb_transitive_mobile_1(VERB_TRANSITIVE_MOBILE_1_TREE
            ),aspect(ASPECT_TREE).

/*198*/ vp_transitive_mobile_2(vp_transitive_mobile_2(VERB_TRANSITIVE_MOBIL
            E_2_TREE,ASPECT_TREE)) --> verb_transitive_mobile_2(VERB_TRANSI
            TIVE_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*199*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_MOB2_TREE,VERB_TRA
            NSITIVE_MOBILE_2_TREE,ASPECT_TREE)) --> np_mob2(NP_MOB2_TREE),[
            i],[x],verb_transitive_mobile_2(VERB_TRANSITIVE_MOBILE_2_TREE),
            aspect(ASPECT_TREE).

/*200*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_TREE,VERB_TRANSITI
            VE_MOBILE_2_TREE,ASPECT_TREE)) --> np(NP_TREE),[e],[x],verb_tra
            nsitive_mobile_2(VERB_TRANSITIVE_MOBILE_2_TREE),aspect(ASPECT_T
            REE).

/*201*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_TREE,NP_MOB2_TREE,
            VERB_TRANSITIVE_MOBILE_2_TREE,ASPECT_TREE)) --> np(NP_TREE),[e]
            ,[x],np_mob2(NP_MOB2_TREE),[i],[x],verb_transitive_mobile_2(VER
            B_TRANSITIVE_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*202*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_TREE,VERB_TRANSITI
            VE_MOBILE_2_TREE,ASPECT_TREE)) --> np(NP_TREE),[den],[x],verb_t
            ransitive_mobile_2(VERB_TRANSITIVE_MOBILE_2_TREE),aspect(ASPECT
            _TREE).

/*203*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_TREE,NP_MOB2_TREE,
            VERB_TRANSITIVE_MOBILE_2_TREE,ASPECT_TREE)) --> np(NP_TREE),[de
            n],[x],np_mob2(NP_MOB2_TREE),[i],[x],verb_transitive_mobile_2(V
            ERB_TRANSITIVE_MOBILE_2_TREE),aspect(ASPECT_TREE).
```

```
/*204*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_TREE,NP_TREE,VERB_
            TRANSITIVE_MOBILE_2_TREE,ASPECT_TREE)) --> np(NP_TREE),[den],[x
            ],np(NP_TREE),[e],[x],verb_transitive_mobile_2(VERB_TRANSITIVE_
            MOBILE_2_TREE),aspect(ASPECT_TREE).

/*205*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_TREE,NP_TREE,NP_MO
            B2_TREE,VERB_TRANSITIVE_MOBILE_2_TREE,ASPECT_TREE)) --> np(NP_T
            REE),[den],[x],np(NP_TREE),[e],[x],np_mob2(NP_MOB2_TREE),[i],[x
            ],verb_transitive_mobile_2(VERB_TRANSITIVE_MOBILE_2_TREE),aspec
            t(ASPECT_TREE).

/*206*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_INS_TREE,VERB_TRAN
            SITIVE_MOBILE_2_TREE,ASPECT_TREE)) --> np_ins(NP_INS_TREE),[le]
            ,[x],verb_transitive_mobile_2(VERB_TRANSITIVE_MOBILE_2_TREE),as
            pect(ASPECT_TREE).

/*207*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_INS_TREE,NP_MOB2_T
            REE,VERB_TRANSITIVE_MOBILE_2_TREE,ASPECT_TREE)) --> np_ins(NP_I
            NS_TREE),[le],[x],np_mob2(NP_MOB2_TREE),[i],[x],verb_transitive
            _mobile_2(VERB_TRANSITIVE_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*208*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_INS_TREE,NP_TREE,V
            ERB_TRANSITIVE_MOBILE_2_TREE,ASPECT_TREE)) --> np_ins(NP_INS_TR
            EE),[le],[x],np(NP_TREE),[e],[x],verb_transitive_mobile_2(VERB_
            TRANSITIVE_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*209*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_INS_TREE,NP_TREE,N
            P_MOB2_TREE,VERB_TRANSITIVE_MOBILE_2_TREE,ASPECT_TREE)) --> np_
            ins(NP_INS_TREE),[le],[x],np(NP_TREE),[e],[x],np_mob2(NP_MOB2_T
            REE),[i],[x],verb_transitive_mobile_2(VERB_TRANSITIVE_MOBILE_2_
            TREE),aspect(ASPECT_TREE).

/*210*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_INS_TREE,NP_TREE,V
            ERB_TRANSITIVE_MOBILE_2_TREE,ASPECT_TREE)) --> np_ins(NP_INS_TR
            EE),[le],[x],np(NP_TREE),[den],[x],verb_transitive_mobile_2(VER
            B_TRANSITIVE_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*211*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_INS_TREE,NP_TREE,N
            P_MOB2_TREE,VERB_TRANSITIVE_MOBILE_2_TREE,ASPECT_TREE)) --> np_
            ins(NP_INS_TREE),[le],[x],np(NP_TREE),[den],[x],np_mob2(NP_MOB2
            _TREE),[i],[x],verb_transitive_mobile_2(VERB_TRANSITIVE_MOBILE_
            2_TREE),aspect(ASPECT_TREE).
```

```
/*212*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_INS_TREE,NP_TREE,N
            P_TREE,VERB_TRANSITIVE_MOBILE_2_TREE,ASPECT_TREE)) --> np_ins(N
            P_INS_TREE),[le],[x],np(NP_TREE),[den],[x],np(NP_TREE),[e],[x],
            verb_transitive_mobile_2(VERB_TRANSITIVE_MOBILE_2_TREE),aspect(
            ASPECT_TREE).

/*213*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_INS_TREE,NP_TREE,N
            P_TREE,NP_MOB2_TREE,VERB_TRANSITIVE_MOBILE_2_TREE,ASPECT_TREE))
             --> np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[den],[x],np(NP_T
            REE),[e],[x],np_mob2(NP_MOB2_TREE),[i],[x],verb_transitive_mobi
            le_2(VERB_TRANSITIVE_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*214*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_TIME_TREE,VERB_TRA
            NSITIVE_MOBILE_2_TREE,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[
            de],[x],verb_transitive_mobile_2(VERB_TRANSITIVE_MOBILE_2_TREE)
            ,aspect(ASPECT_TREE).

/*215*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_TIME_TREE,NP_MOB2_
            TREE,VERB_TRANSITIVE_MOBILE_2_TREE,ASPECT_TREE)) --> np_time(NP
            _TIME_TREE),[de],[x],np_mob2(NP_MOB2_TREE),[i],[x],verb_transit
            ive_mobile_2(VERB_TRANSITIVE_MOBILE_2_TREE),aspect(ASPECT_TREE)
            .

/*216*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_TIME_TREE,NP_TREE,
            VERB_TRANSITIVE_MOBILE_2_TREE,ASPECT_TREE)) --> np_time(NP_TIME
            _TREE),[de],[x],np(NP_TREE),[e],[x],verb_transitive_mobile_2(VE
            RB_TRANSITIVE_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*217*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_TIME_TREE,NP_TREE,
            NP_MOB2_TREE,VERB_TRANSITIVE_MOBILE_2_TREE,ASPECT_TREE)) --> np
            _time(NP_TIME_TREE),[de],[x],np(NP_TREE),[e],[x],np_mob2(NP_MOB
            2_TREE),[i],[x],verb_transitive_mobile_2(VERB_TRANSITIVE_MOBILE
            _2_TREE),aspect(ASPECT_TREE).

/*218*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_TIME_TREE,NP_TREE,
            VERB_TRANSITIVE_MOBILE_2_TREE,ASPECT_TREE)) --> np_time(NP_TIME
            _TREE),[de],[x],np(NP_TREE),[den],[x],verb_transitive_mobile_2(
            VERB_TRANSITIVE_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*219*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_TIME_TREE,NP_TREE,
            NP_MOB2_TREE,VERB_TRANSITIVE_MOBILE_2_TREE,ASPECT_TREE)) --> np
            _time(NP_TIME_TREE),[de],[x],np(NP_TREE),[den],[x],np_mob2(NP_M
            OB2_TREE),[i],[x],verb_transitive_mobile_2(VERB_TRANSITIVE_MOBI
            LE_2_TREE),aspect(ASPECT_TREE).
```

```
/*220*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_TIME_TREE,NP_TREE,
            NP_TREE,VERB_TRANSITIVE_MOBILE_2_TREE,ASPECT_TREE)) --> np_time
            (NP_TIME_TREE),[de],[x],np(NP_TREE),[den],[x],np(NP_TREE),[e],[
            x],verb_transitive_mobile_2(VERB_TRANSITIVE_MOBILE_2_TREE),aspe
            ct(ASPECT_TREE).

/*221*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_TIME_TREE,NP_TREE,
            NP_TREE,NP_MOB2_TREE,VERB_TRANSITIVE_MOBILE_2_TREE,ASPECT_TREE)
            ) --> np_time(NP_TIME_TREE),[de],[x],np(NP_TREE),[den],[x],np(N
            P_TREE),[e],[x],np_mob2(NP_MOB2_TREE),[i],[x],verb_transitive_m
            obile_2(VERB_TRANSITIVE_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*222*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_TIME_TREE,NP_INS_T
            REE,VERB_TRANSITIVE_MOBILE_2_TREE,ASPECT_TREE)) --> np_time(NP_
            TIME_TREE),[de],[x],np_ins(NP_INS_TREE),[le],[x],verb_transitiv
            e_mobile_2(VERB_TRANSITIVE_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*223*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_TIME_TREE,NP_INS_T
            REE,NP_MOB2_TREE,VERB_TRANSITIVE_MOBILE_2_TREE,ASPECT_TREE)) --
            > np_time(NP_TIME_TREE),[de],[x],np_ins(NP_INS_TREE),[le],[x],n
            p_mob2(NP_MOB2_TREE),[i],[x],verb_transitive_mobile_2(VERB_TRAN
            SITIVE_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*224*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_TIME_TREE,NP_INS_T
            REE,NP_TREE,VERB_TRANSITIVE_MOBILE_2_TREE,ASPECT_TREE)) --> np_
            time(NP_TIME_TREE),[de],[x],np_ins(NP_INS_TREE),[le],[x],np(NP_
            TREE),[e],[x],verb_transitive_mobile_2(VERB_TRANSITIVE_MOBILE_2
            _TREE),aspect(ASPECT_TREE).

/*225*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_TIME_TREE,NP_INS_T
            REE,NP_TREE,NP_MOB2_TREE,VERB_TRANSITIVE_MOBILE_2_TREE,ASPECT_T
            REE)) --> np_time(NP_TIME_TREE),[de],[x],np_ins(NP_INS_TREE),[l
            e],[x],np(NP_TREE),[e],[x],np_mob2(NP_MOB2_TREE),[i],[x],verb_t
            ransitive_mobile_2(VERB_TRANSITIVE_MOBILE_2_TREE),aspect(ASPECT
            _TREE).

/*226*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_TIME_TREE,NP_INS_T
            REE,NP_TREE,VERB_TRANSITIVE_MOBILE_2_TREE,ASPECT_TREE)) --> np_
            time(NP_TIME_TREE),[de],[x],np_ins(NP_INS_TREE),[le],[x],np(NP_
            TREE),[den],[x],verb_transitive_mobile_2(VERB_TRANSITIVE_MOBILE
            _2_TREE),aspect(ASPECT_TREE).
```

```
/*227*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_TIME_TREE,NP_INS_T
            REE,NP_TREE,NP_MOB2_TREE,VERB_TRANSITIVE_MOBILE_2_TREE,ASPECT_T
            REE)) --> np_time(NP_TIME_TREE),[de],[x],np_ins(NP_INS_TREE),[l
            e],[x],np(NP_TREE),[den],[x],np_mob2(NP_MOB2_TREE),[i],[x],verb
            _transitive_mobile_2(VERB_TRANSITIVE_MOBILE_2_TREE),aspect(ASPE
            CT_TREE).

/*228*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_TIME_TREE,NP_INS_T
            REE,NP_TREE,NP_TREE,VERB_TRANSITIVE_MOBILE_2_TREE,ASPECT_TREE))
             --> np_time(NP_TIME_TREE),[de],[x],np_ins(NP_INS_TREE),[le],[x
            ],np(NP_TREE),[den],[x],np(NP_TREE),[e],[x],verb_transitive_mob
            ile_2(VERB_TRANSITIVE_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*229*/ vp_transitive_mobile_2(vp_transitive_mobile_2(NP_TIME_TREE,NP_INS_T
            REE,NP_TREE,NP_TREE,NP_MOB2_TREE,VERB_TRANSITIVE_MOBILE_2_TREE,
            ASPECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np_ins(NP_INS_
            TREE),[le],[x],np(NP_TREE),[den],[x],np(NP_TREE),[e],[x],np_mob
            2(NP_MOB2_TREE),[i],[x],verb_transitive_mobile_2(VERB_TRANSITIV
            E_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*230*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(VERB_TRAN
            SITIVE_HUMAN_MOBILE_1_TREE,ASPECT_TREE)) --> verb_transitive_hu
            man_mobile_1(VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE),aspect(ASPECT
            _TREE).

/*231*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_TREE,V
            ERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,ASPECT_TREE)) --> np(NP_TREE
            ),[i],[x],verb_transitive_human_mobile_1(VERB_TRANSITIVE_HUMAN_
            MOBILE_1_TREE),aspect(ASPECT_TREE).

/*232*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_TREE,V
            ERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,ASPECT_TREE)) --> np(NP_TREE
            ),[e],[x],verb_transitive_human_mobile_1(VERB_TRANSITIVE_HUMAN_
            MOBILE_1_TREE),aspect(ASPECT_TREE).

/*233*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_TREE,N
            P_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,ASPECT_TREE)) --> np
            (NP_TREE),[e],[x],np(NP_TREE),[i],[x],verb_transitive_human_mob
            ile_1(VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE),aspect(ASPECT_TREE).
```

```
/*234*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_TREE,V
            ERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,ASPECT_TREE)) --> np(NP_TREE
            ),[den],[x],verb_transitive_human_mobile_1(VERB_TRANSITIVE_HUMA
            N_MOBILE_1_TREE),aspect(ASPECT_TREE).

/*235*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_TREE,N
            P_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,ASPECT_TREE)) --> np
            (NP_TREE),[den],[x],np(NP_TREE),[i],[x],verb_transitive_human_m
            obile_1(VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE),aspect(ASPECT_TREE
            ).

/*236*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_TREE,N
            P_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,ASPECT_TREE)) --> np
            (NP_TREE),[den],[x],np(NP_TREE),[e],[x],verb_transitive_human_m
            obile_1(VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE),aspect(ASPECT_TREE
            ).

/*237*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_TREE,N
            P_TREE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,ASPECT_TREE)
            ) --> np(NP_TREE),[den],[x],np(NP_TREE),[e],[x],np(NP_TREE),[i]
            ,[x],verb_transitive_human_mobile_1(VERB_TRANSITIVE_HUMAN_MOBIL
            E_1_TREE),aspect(ASPECT_TREE).

/*238*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_INS_TR
            EE,VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,ASPECT_TREE)) --> np_ins
            (NP_INS_TREE),[le],[x],verb_transitive_human_mobile_1(VERB_TRAN
            SITIVE_HUMAN_MOBILE_1_TREE),aspect(ASPECT_TREE).

/*239*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_INS_TR
            EE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,ASPECT_TREE)) --
            > np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[i],[x],verb_transit
            ive_human_mobile_1(VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE),aspect(
            ASPECT_TREE).

/*240*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_INS_TR
            EE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,ASPECT_TREE)) --
            > np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[e],[x],verb_transit
            ive_human_mobile_1(VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE),aspect(
            ASPECT_TREE).
```

```
/*241*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_INS_TR
            EE,NP_TREE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,ASPECT_T
            REE)) --> np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[e],[x],np(N
            P_TREE),[i],[x],verb_transitive_human_mobile_1(VERB_TRANSITIVE_
            HUMAN_MOBILE_1_TREE),aspect(ASPECT_TREE).

/*242*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_INS_TR
            EE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,ASPECT_TREE)) --
            > np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[den],[x],verb_trans
            itive_human_mobile_1(VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE),aspec
            t(ASPECT_TREE).

/*243*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_INS_TR
            EE,NP_TREE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,ASPECT_T
            REE)) --> np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[den],[x],np
            (NP_TREE),[i],[x],verb_transitive_human_mobile_1(VERB_TRANSITIV
            E_HUMAN_MOBILE_1_TREE),aspect(ASPECT_TREE).

/*244*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_INS_TR
            EE,NP_TREE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,ASPECT_T
            REE)) --> np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[den],[x],np
            (NP_TREE),[e],[x],verb_transitive_human_mobile_1(VERB_TRANSITIV
            E_HUMAN_MOBILE_1_TREE),aspect(ASPECT_TREE).

/*245*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_INS_TR
            EE,NP_TREE,NP_TREE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,
            ASPECT_TREE)) --> np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[den
            ],[x],np(NP_TREE),[e],[x],np(NP_TREE),[i],[x],verb_transitive_h
            uman_mobile_1(VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE),aspect(ASPEC
            T_TREE).

/*246*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_TIME_T
            REE,VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,ASPECT_TREE)) --> np_ti
            me(NP_TIME_TREE),[de],[x],verb_transitive_human_mobile_1(VERB_T
            RANSITIVE_HUMAN_MOBILE_1_TREE),aspect(ASPECT_TREE).

/*247*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_TIME_T
            REE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,ASPECT_TREE)) -
            -> np_time(NP_TIME_TREE),[de],[x],np(NP_TREE),[i],[x],verb_tran
            sitive_human_mobile_1(VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE),aspe
            ct(ASPECT_TREE).
```

```
/*248*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_TIME_T
            REE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,ASPECT_TREE)) -
            -> np_time(NP_TIME_TREE),[de],[x],np(NP_TREE),[e],[x],verb_tran
            sitive_human_mobile_1(VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE),aspe
            ct(ASPECT_TREE).

/*249*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_TIME_T
            REE,NP_TREE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,ASPECT_
            TREE)) --> np_time(NP_TIME_TREE),[de],[x],np(NP_TREE),[e],[x],n
            p(NP_TREE),[i],[x],verb_transitive_human_mobile_1(VERB_TRANSITI
            VE_HUMAN_MOBILE_1_TREE),aspect(ASPECT_TREE).

/*250*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_TIME_T
            REE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,ASPECT_TREE)) -
            -> np_time(NP_TIME_TREE),[de],[x],np(NP_TREE),[den],[x],verb_tr
            ansitive_human_mobile_1(VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE),as
            pect(ASPECT_TREE).

/*251*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_TIME_T
            REE,NP_TREE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,ASPECT_
            TREE)) --> np_time(NP_TIME_TREE),[de],[x],np(NP_TREE),[den],[x]
            ,np(NP_TREE),[i],[x],verb_transitive_human_mobile_1(VERB_TRANSI
            TIVE_HUMAN_MOBILE_1_TREE),aspect(ASPECT_TREE).

/*252*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_TIME_T
            REE,NP_TREE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,ASPECT_
            TREE)) --> np_time(NP_TIME_TREE),[de],[x],np(NP_TREE),[den],[x]
            ,np(NP_TREE),[e],[x],verb_transitive_human_mobile_1(VERB_TRANSI
            TIVE_HUMAN_MOBILE_1_TREE),aspect(ASPECT_TREE).

/*253*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_TIME_T
            REE,NP_TREE,NP_TREE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE
            ,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np(NP_TREE),[
            den],[x],np(NP_TREE),[e],[x],np(NP_TREE),[i],[x],verb_transitiv
            e_human_mobile_1(VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE),aspect(AS
            PECT_TREE).

/*254*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_TIME_T
            REE,NP_INS_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,ASPECT_TREE
            )) --> np_time(NP_TIME_TREE),[de],[x],np_ins(NP_INS_TREE),[le],
            [x],verb_transitive_human_mobile_1(VERB_TRANSITIVE_HUMAN_MOBILE
            _1_TREE),aspect(ASPECT_TREE).
```

```
/*255*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_TIME_T
            REE,NP_INS_TREE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,ASP
            ECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np_ins(NP_INS_TRE
            E),[le],[x],np(NP_TREE),[i],[x],verb_transitive_human_mobile_1(
            VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE),aspect(ASPECT_TREE).

/*256*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_TIME_T
            REE,NP_INS_TREE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,ASP
            ECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np_ins(NP_INS_TRE
            E),[le],[x],np(NP_TREE),[e],[x],verb_transitive_human_mobile_1(
            VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE),aspect(ASPECT_TREE).

/*257*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_TIME_T
            REE,NP_INS_TREE,NP_TREE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_1_
            TREE,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np_ins(NP
            _INS_TREE),[le],[x],np(NP_TREE),[e],[x],np(NP_TREE),[i],[x],ver
            b_transitive_human_mobile_1(VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE
            ),aspect(ASPECT_TREE).

/*258*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_TIME_T
            REE,NP_INS_TREE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE,ASP
            ECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np_ins(NP_INS_TRE
            E),[le],[x],np(NP_TREE),[den],[x],verb_transitive_human_mobile_
            1(VERB_TRANSITIVE_HUMAN_MOBILE_1_TREE),aspect(ASPECT_TREE).

/*259*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_TIME_T
            REE,NP_INS_TREE,NP_TREE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_1_
            TREE,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np_ins(NP
            _INS_TREE),[le],[x],np(NP_TREE),[den],[x],np(NP_TREE),[i],[x],v
            erb_transitive_human_mobile_1(VERB_TRANSITIVE_HUMAN_MOBILE_1_TR
            EE),aspect(ASPECT_TREE).

/*260*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_TIME_T
            REE,NP_INS_TREE,NP_TREE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_1_
            TREE,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np_ins(NP
            _INS_TREE),[le],[x],np(NP_TREE),[den],[x],np(NP_TREE),[e],[x],v
            erb_transitive_human_mobile_1(VERB_TRANSITIVE_HUMAN_MOBILE_1_TR
            EE),aspect(ASPECT_TREE).

/*261*/ vp_transitive_human_mobile_1(vp_transitive_human_mobile_1(NP_TIME_T
            REE,NP_INS_TREE,NP_TREE,NP_TREE,NP_TREE,VERB_TRANSITIVE_HUMAN_M
```

```
            OBILE_1_TREE,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],n
            p_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[den],[x],np(NP_TREE),[
            e],[x],np(NP_TREE),[i],[x],verb_transitive_human_mobile_1(VERB_
            TRANSITIVE_HUMAN_MOBILE_1_TREE),aspect(ASPECT_TREE).

/*262*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(VERB_TRAN
            SITIVE_HUMAN_MOBILE_2_TREE,ASPECT_TREE)) --> verb_transitive_hu
            man_mobile_2(VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE),aspect(ASPECT
            _TREE).

/*263*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_MOB2_T
            REE,VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE,ASPECT_TREE)) --> np_mo
            b2(NP_MOB2_TREE),[i],[x],verb_transitive_human_mobile_2(VERB_TR
            ANSITIVE_HUMAN_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*264*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_TREE,V
            ERB_TRANSITIVE_HUMAN_MOBILE_2_TREE,ASPECT_TREE)) --> np(NP_TREE
            ),[e],[x],verb_transitive_human_mobile_2(VERB_TRANSITIVE_HUMAN_
            MOBILE_2_TREE),aspect(ASPECT_TREE).

/*265*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_TREE,N
            P_MOB2_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE,ASPECT_TREE)) -
            -> np(NP_TREE),[e],[x],np_mob2(NP_MOB2_TREE),[i],[x],verb_trans
            itive_human_mobile_2(VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE),aspec
            t(ASPECT_TREE).

/*266*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_TREE,V
            ERB_TRANSITIVE_HUMAN_MOBILE_2_TREE,ASPECT_TREE)) --> np(NP_TREE
            ),[den],[x],verb_transitive_human_mobile_2(VERB_TRANSITIVE_HUMA
            N_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*267*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_TREE,N
            P_MOB2_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE,ASPECT_TREE)) -
            -> np(NP_TREE),[den],[x],np_mob2(NP_MOB2_TREE),[i],[x],verb_tra
            nsitive_human_mobile_2(VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE),asp
            ect(ASPECT_TREE).

/*268*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_TREE,N
            P_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE,ASPECT_TREE)) --> np
            (NP_TREE),[den],[x],np(NP_TREE),[e],[x],verb_transitive_human_m
            obile_2(VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE),aspect(ASPECT_TREE
            ).
```

```
/*269*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_TREE,N
            P_TREE,NP_MOB2_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE,ASPECT_
            TREE)) --> np(NP_TREE),[den],[x],np(NP_TREE),[e],[x],np_mob2(NP
            _MOB2_TREE),[i],[x],verb_transitive_human_mobile_2(VERB_TRANSIT
            IVE_HUMAN_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*270*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_INS_TR
            EE,VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE,ASPECT_TREE)) --> np_ins
            (NP_INS_TREE),[le],[x],verb_transitive_human_mobile_2(VERB_TRAN
            SITIVE_HUMAN_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*271*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_INS_TR
            EE,NP_MOB2_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE,ASPECT_TREE
            )) --> np_ins(NP_INS_TREE),[le],[x],np_mob2(NP_MOB2_TREE),[i],[
            x],verb_transitive_human_mobile_2(VERB_TRANSITIVE_HUMAN_MOBILE_
            2_TREE),aspect(ASPECT_TREE).

/*272*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_INS_TR
            EE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE,ASPECT_TREE)) --
            > np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[e],[x],verb_transit
            ive_human_mobile_2(VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE),aspect(
            ASPECT_TREE).

/*273*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_INS_TR
            EE,NP_TREE,NP_MOB2_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE,ASP
            ECT_TREE)) --> np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[e],[x]
            ,np_mob2(NP_MOB2_TREE),[i],[x],verb_transitive_human_mobile_2(V
            ERB_TRANSITIVE_HUMAN_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*274*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_INS_TR
            EE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE,ASPECT_TREE)) --
            > np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[den],[x],verb_trans
            itive_human_mobile_2(VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE),aspec
            t(ASPECT_TREE).

/*275*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_INS_TR
            EE,NP_TREE,NP_MOB2_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE,ASP
            ECT_TREE)) --> np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[den],[
            x],np_mob2(NP_MOB2_TREE),[i],[x],verb_transitive_human_mobile_2
            (VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE),aspect(ASPECT_TREE).
```

```
/*276*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_INS_TR
            EE,NP_TREE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE,ASPECT_T
            REE)) --> np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[den],[x],np
            (NP_TREE),[e],[x],verb_transitive_human_mobile_2(VERB_TRANSITIV
            E_HUMAN_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*277*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_INS_TR
            EE,NP_TREE,NP_TREE,NP_MOB2_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_2_
            TREE,ASPECT_TREE)) --> np_ins(NP_INS_TREE),[le],[x],np(NP_TREE)
            ,[den],[x],np(NP_TREE),[e],[x],np_mob2(NP_MOB2_TREE),[i],[x],ve
            rb_transitive_human_mobile_2(VERB_TRANSITIVE_HUMAN_MOBILE_2_TRE
            E),aspect(ASPECT_TREE).

/*278*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_TIME_T
            REE,VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE,ASPECT_TREE)) --> np_ti
            me(NP_TIME_TREE),[de],[x],verb_transitive_human_mobile_2(VERB_T
            RANSITIVE_HUMAN_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*279*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_TIME_T
            REE,NP_MOB2_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE,ASPECT_TRE
            E)) --> np_time(NP_TIME_TREE),[de],[x],np_mob2(NP_MOB2_TREE),[i
            ],[x],verb_transitive_human_mobile_2(VERB_TRANSITIVE_HUMAN_MOBI
            LE_2_TREE),aspect(ASPECT_TREE).

/*280*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_TIME_T
            REE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE,ASPECT_TREE)) -
            -> np_time(NP_TIME_TREE),[de],[x],np(NP_TREE),[e],[x],verb_tran
            sitive_human_mobile_2(VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE),aspe
            ct(ASPECT_TREE).

/*281*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_TIME_T
            REE,NP_TREE,NP_MOB2_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE,AS
            PECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np(NP_TREE),[e],
            [x],np_mob2(NP_MOB2_TREE),[i],[x],verb_transitive_human_mobile_
            2(VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*282*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_TIME_T
            REE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE,ASPECT_TREE)) -
            -> np_time(NP_TIME_TREE),[de],[x],np(NP_TREE),[den],[x],verb_tr
            ansitive_human_mobile_2(VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE),as
            pect(ASPECT_TREE).
```

```
/*283*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_TIME_T
            REE,NP_TREE,NP_MOB2_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE,AS
            PECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np(NP_TREE),[den
            ],[x],np_mob2(NP_MOB2_TREE),[i],[x],verb_transitive_human_mobil
            e_2(VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*284*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_TIME_T
            REE,NP_TREE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE,ASPECT_
            TREE)) --> np_time(NP_TIME_TREE),[de],[x],np(NP_TREE),[den],[x]
            ,np(NP_TREE),[e],[x],verb_transitive_human_mobile_2(VERB_TRANSI
            TIVE_HUMAN_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*285*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_TIME_T
            REE,NP_TREE,NP_TREE,NP_MOB2_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_2
            _TREE,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np(NP_TR
            EE),[den],[x],np(NP_TREE),[e],[x],np_mob2(NP_MOB2_TREE),[i],[x]
            ,verb_transitive_human_mobile_2(VERB_TRANSITIVE_HUMAN_MOBILE_2_
            TREE),aspect(ASPECT_TREE).

/*286*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_TIME_T
            REE,NP_INS_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE,ASPECT_TREE
            )) --> np_time(NP_TIME_TREE),[de],[x],np_ins(NP_INS_TREE),[le],
            [x],verb_transitive_human_mobile_2(VERB_TRANSITIVE_HUMAN_MOBILE
            _2_TREE),aspect(ASPECT_TREE).

/*287*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_TIME_T
            REE,NP_INS_TREE,NP_MOB2_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_2_TRE
            E,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np_ins(NP_IN
            S_TREE),[le],[x],np_mob2(NP_MOB2_TREE),[i],[x],verb_transitive_
            human_mobile_2(VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE),aspect(ASPE
            CT_TREE).

/*288*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_TIME_T
            REE,NP_INS_TREE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE,ASP
            ECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np_ins(NP_INS_TRE
            E),[le],[x],np(NP_TREE),[e],[x],verb_transitive_human_mobile_2(
            VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*289*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_TIME_T
            REE,NP_INS_TREE,NP_TREE,NP_MOB2_TREE,VERB_TRANSITIVE_HUMAN_MOBI
            LE_2_TREE,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np_i
```

```
            ns(NP_INS_TREE),[le],[x],np(NP_TREE),[e],[x],np_mob2(NP_MOB2_TR
            EE),[i],[x],verb_transitive_human_mobile_2(VERB_TRANSITIVE_HUMA
            N_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*290*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_TIME_T
            REE,NP_INS_TREE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE,ASP
            ECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np_ins(NP_INS_TRE
            E),[le],[x],np(NP_TREE),[den],[x],verb_transitive_human_mobile_
            2(VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*291*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_TIME_T
            REE,NP_INS_TREE,NP_TREE,NP_MOB2_TREE,VERB_TRANSITIVE_HUMAN_MOBI
            LE_2_TREE,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np_i
            ns(NP_INS_TREE),[le],[x],np(NP_TREE),[den],[x],np_mob2(NP_MOB2_
            TREE),[i],[x],verb_transitive_human_mobile_2(VERB_TRANSITIVE_HU
            MAN_MOBILE_2_TREE),aspect(ASPECT_TREE).

/*292*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_TIME_T
            REE,NP_INS_TREE,NP_TREE,NP_TREE,VERB_TRANSITIVE_HUMAN_MOBILE_2_
            TREE,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[de],[x],np_ins(NP
            _INS_TREE),[le],[x],np(NP_TREE),[den],[x],np(NP_TREE),[e],[x],v
            erb_transitive_human_mobile_2(VERB_TRANSITIVE_HUMAN_MOBILE_2_TR
            EE),aspect(ASPECT_TREE).

/*293*/ vp_transitive_human_mobile_2(vp_transitive_human_mobile_2(NP_TIME_T
            REE,NP_INS_TREE,NP_TREE,NP_TREE,NP_MOB2_TREE,VERB_TRANSITIVE_HU
            MAN_MOBILE_2_TREE,ASPECT_TREE)) --> np_time(NP_TIME_TREE),[de],
            [x],np_ins(NP_INS_TREE),[le],[x],np(NP_TREE),[den],[x],np(NP_TR
            EE),[e],[x],np_mob2(NP_MOB2_TREE),[i],[x],verb_transitive_human
            _mobile_2(VERB_TRANSITIVE_HUMAN_MOBILE_2_TREE),aspect(ASPECT_TR
            EE).

/*294*/ adv(adv(POSTPOSPHRASE_F_TREE)) --> postposphrase_f(POSTPOSPHRASE_F_
            TREE).

/*295*/ adv(adv(POSTPOSPHRASE_D_TREE)) --> postposphrase_d(POSTPOSPHRASE_D_
            TREE).

/*296*/ adv(adv(POSTPOSPHRASE_E_TREE)) --> postposphrase_e(POSTPOSPHRASE_E_
            TREE).
```

```
/*297*/ adv(adv(POSTPOSPHRASE_D_TREE,POSTPOSPHRASE_F_TREE)) --> postposphra
            se_d(POSTPOSPHRASE_D_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_
            TREE).

/*298*/ adv(adv(POSTPOSPHRASE_E_TREE,POSTPOSPHRASE_F_TREE)) --> postposphra
            se_e(POSTPOSPHRASE_E_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_
            TREE).

/*299*/ adv(adv(POSTPOSPHRASE_A_TREE)) --> postposphrase_a(POSTPOSPHRASE_A_
            TREE).

/*300*/ adv(adv(POSTPOSPHRASE_F_TREE,POSTPOSPHRASE_F_TREE)) --> postposphra
            se_f(POSTPOSPHRASE_F_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_
            TREE).

/*301*/ adv(adv(POSTPOSPHRASE_A_TREE,POSTPOSPHRASE_D_TREE)) --> postposphra
            se_a(POSTPOSPHRASE_A_TREE),[x],postposphrase_d(POSTPOSPHRASE_D_
            TREE).

/*302*/ adv(adv(POSTPOSPHRASE_A_TREE,POSTPOSPHRASE_E_TREE)) --> postposphra
            se_a(POSTPOSPHRASE_A_TREE),[x],postposphrase_e(POSTPOSPHRASE_E_
            TREE).

/*303*/ adv(adv(POSTPOSPHRASE_A_TREE,POSTPOSPHRASE_D_TREE,POSTPOSPHRASE_F_T
            REE)) --> postposphrase_a(POSTPOSPHRASE_A_TREE),[x],postposphra
            se_d(POSTPOSPHRASE_D_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_
            TREE).

/*304*/ adv(adv(POSTPOSPHRASE_A_TREE,POSTPOSPHRASE_E_TREE,POSTPOSPHRASE_F_T
            REE)) --> postposphrase_a(POSTPOSPHRASE_A_TREE),[x],postposphra
            se_e(POSTPOSPHRASE_E_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_
            TREE).

/*305*/ adv(adv(POSTPOSPHRASE_B_TREE)) --> postposphrase_b(POSTPOSPHRASE_B_
            TREE),[x].

/*306*/ adv(adv(POSTPOSPHRASE_B_TREE,POSTPOSPHRASE_F_TREE)) --> postposphra
            se_b(POSTPOSPHRASE_B_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_
            TREE).

/*307*/ adv(adv(POSTPOSPHRASE_B_TREE,POSTPOSPHRASE_D_TREE)) --> postposphra
            se_b(POSTPOSPHRASE_B_TREE),[x],postposphrase_d(POSTPOSPHRASE_D_
            TREE).
```

```
/*308*/ adv(adv(POSTPOSPHRASE_B_TREE,POSTPOSPHRASE_E_TREE)) --> postposphra
            se_b(POSTPOSPHRASE_B_TREE),[x],postposphrase_e(POSTPOSPHRASE_E_
            TREE).

/*309*/ adv(adv(POSTPOSPHRASE_B_TREE,POSTPOSPHRASE_D_TREE,POSTPOSPHRASE_F_T
            REE)) --> postposphrase_b(POSTPOSPHRASE_B_TREE),[x],postposphra
            se_d(POSTPOSPHRASE_D_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_
            TREE).

/*310*/ adv(adv(POSTPOSPHRASE_B_TREE,POSTPOSPHRASE_E_TREE,POSTPOSPHRASE_F_T
            REE)) --> postposphrase_b(POSTPOSPHRASE_B_TREE),[x],postposphra
            se_e(POSTPOSPHRASE_E_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_
            TREE).

/*311*/ adv(adv(POSTPOSPHRASE_C_TREE)) --> postposphrase_c(POSTPOSPHRASE_C_
            TREE),[x].

/*312*/ adv(adv(POSTPOSPHRASE_C_TREE,POSTPOSPHRASE_F_TREE)) --> postposphra
            se_c(POSTPOSPHRASE_C_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_
            TREE).

/*313*/ adv(adv(POSTPOSPHRASE_C_TREE,POSTPOSPHRASE_D_TREE)) --> postposphra
            se_c(POSTPOSPHRASE_C_TREE),[x],postposphrase_d(POSTPOSPHRASE_D_
            TREE).

/*314*/ adv(adv(POSTPOSPHRASE_C_TREE,POSTPOSPHRASE_E_TREE)) --> postposphra
            se_c(POSTPOSPHRASE_C_TREE),[x],postposphrase_e(POSTPOSPHRASE_E_
            TREE).

/*315*/ adv(adv(POSTPOSPHRASE_C_TREE,POSTPOSPHRASE_D_TREE,POSTPOSPHRASE_F_T
            REE)) --> postposphrase_c(POSTPOSPHRASE_C_TREE),[x],postposphra
            se_d(POSTPOSPHRASE_D_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_
            TREE).

/*316*/ adv(adv(POSTPOSPHRASE_C_TREE,POSTPOSPHRASE_E_TREE,POSTPOSPHRASE_F_T
            REE)) --> postposphrase_c(POSTPOSPHRASE_C_TREE),[x],postposphra
            se_e(POSTPOSPHRASE_E_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_
            TREE).

/*317*/ adv(adv(PARTICLE_A_TREE)) --> particle_a(PARTICLE_A_TREE).
```

```
/*318*/ adv(adv(PARTICLE_A_TREE,POSTPOSPHRASE_F_TREE)) --> particle_a(PARTI
           CLE_A_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_TREE).

/*319*/ adv(adv(PARTICLE_A_TREE,POSTPOSPHRASE_D_TREE)) --> particle_a(PARTI
           CLE_A_TREE),[x],postposphrase_d(POSTPOSPHRASE_D_TREE).

/*320*/ adv(adv(PARTICLE_A_TREE,POSTPOSPHRASE_E_TREE)) --> particle_a(PARTI
           CLE_A_TREE),[x],postposphrase_e(POSTPOSPHRASE_E_TREE).

/*321*/ adv(adv(PARTICLE_A_TREE,POSTPOSPHRASE_D_TREE,POSTPOSPHRASE_F_TREE))
            --> particle_a(PARTICLE_A_TREE),[x],postposphrase_d(POSTPOSPHR
           ASE_D_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_TREE).

/*322*/ adv(adv(PARTICLE_A_TREE,POSTPOSPHRASE_E_TREE,POSTPOSPHRASE_F_TREE))
            --> particle_a(PARTICLE_A_TREE),[x],postposphrase_e(POSTPOSPHR
           ASE_E_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_TREE).

/*323*/ adv(adv(PARTICLE_A_TREE,POSTPOSPHRASE_A_TREE)) --> particle_a(PARTI
           CLE_A_TREE),[x],postposphrase_a(POSTPOSPHRASE_A_TREE).

/*324*/ adv(adv(PARTICLE_A_TREE,POSTPOSPHRASE_F_TREE,POSTPOSPHRASE_F_TREE))
            --> particle_a(PARTICLE_A_TREE),[x],postposphrase_f(POSTPOSPHR
           ASE_F_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_TREE).

/*325*/ adv(adv(PARTICLE_A_TREE,POSTPOSPHRASE_A_TREE,POSTPOSPHRASE_D_TREE))
            --> particle_a(PARTICLE_A_TREE),[x],postposphrase_a(POSTPOSPHR
           ASE_A_TREE),[x],postposphrase_d(POSTPOSPHRASE_D_TREE).

/*326*/ adv(adv(PARTICLE_A_TREE,POSTPOSPHRASE_A_TREE,POSTPOSPHRASE_E_TREE))
            --> particle_a(PARTICLE_A_TREE),[x],postposphrase_a(POSTPOSPHR
           ASE_A_TREE),[x],postposphrase_e(POSTPOSPHRASE_E_TREE).

/*327*/ adv(adv(PARTICLE_A_TREE,POSTPOSPHRASE_A_TREE,POSTPOSPHRASE_D_TREE,P
           OSTPOSPHRASE_F_TREE)) --> particle_a(PARTICLE_A_TREE),[x],postp
           osphrase_a(POSTPOSPHRASE_A_TREE),[x],postposphrase_d(POSTPOSPHR
           ASE_D_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_TREE).

/*328*/ adv(adv(PARTICLE_A_TREE,POSTPOSPHRASE_A_TREE,POSTPOSPHRASE_E_TREE,P
           OSTPOSPHRASE_F_TREE)) --> particle_a(PARTICLE_A_TREE),[x],postp
           osphrase_a(POSTPOSPHRASE_A_TREE),[x],postposphrase_e(POSTPOSPHR
           ASE_E_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_TREE).
```

```
/*329*/ adv(adv(PARTICLE_A_TREE,POSTPOSPHRASE_B_TREE)) --> particle_a(PARTI
            CLE_A_TREE),[x],postposphrase_b(POSTPOSPHRASE_B_TREE),[x].

/*330*/ adv(adv(PARTICLE_A_TREE,POSTPOSPHRASE_B_TREE,POSTPOSPHRASE_F_TREE))
             --> particle_a(PARTICLE_A_TREE),[x],postposphrase_b(POSTPOSPHR
            ASE_B_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_TREE).

/*331*/ adv(adv(PARTICLE_A_TREE,POSTPOSPHRASE_B_TREE,POSTPOSPHRASE_D_TREE))
             --> particle_a(PARTICLE_A_TREE),[x],postposphrase_b(POSTPOSPHR
            ASE_B_TREE),[x],postposphrase_d(POSTPOSPHRASE_D_TREE).

/*332*/ adv(adv(PARTICLE_A_TREE,POSTPOSPHRASE_B_TREE,POSTPOSPHRASE_E_TREE))
             --> particle_a(PARTICLE_A_TREE),[x],postposphrase_b(POSTPOSPHR
            ASE_B_TREE),[x],postposphrase_e(POSTPOSPHRASE_E_TREE).

/*333*/ adv(adv(PARTICLE_A_TREE,POSTPOSPHRASE_B_TREE,POSTPOSPHRASE_D_TREE,P
            OSTPOSPHRASE_F_TREE)) --> particle_a(PARTICLE_A_TREE),[x],postp
            osphrase_b(POSTPOSPHRASE_B_TREE),[x],postposphrase_d(POSTPOSPHR
            ASE_D_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_TREE).

/*334*/ adv(adv(PARTICLE_A_TREE,POSTPOSPHRASE_B_TREE,POSTPOSPHRASE_E_TREE,P
            OSTPOSPHRASE_F_TREE)) --> particle_a(PARTICLE_A_TREE),[x],postp
            osphrase_b(POSTPOSPHRASE_B_TREE),[x],postposphrase_e(POSTPOSPHR
            ASE_E_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_TREE).

/*335*/ adv(adv(PARTICLE_A_TREE,POSTPOSPHRASE_C_TREE)) --> particle_a(PARTI
            CLE_A_TREE),[x],postposphrase_c(POSTPOSPHRASE_C_TREE),[x].

/*336*/ adv(adv(PARTICLE_A_TREE,POSTPOSPHRASE_C_TREE,POSTPOSPHRASE_F_TREE))
             --> particle_a(PARTICLE_A_TREE),[x],postposphrase_c(POSTPOSPHR
            ASE_C_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_TREE).

/*337*/ adv(adv(PARTICLE_A_TREE,POSTPOSPHRASE_C_TREE,POSTPOSPHRASE_D_TREE))
             --> particle_a(PARTICLE_A_TREE),[x],postposphrase_c(POSTPOSPHR
            ASE_C_TREE),[x],postposphrase_d(POSTPOSPHRASE_D_TREE).

/*338*/ adv(adv(PARTICLE_A_TREE,POSTPOSPHRASE_C_TREE,POSTPOSPHRASE_E_TREE))
             --> particle_a(PARTICLE_A_TREE),[x],postposphrase_c(POSTPOSPHR
            ASE_C_TREE),[x],postposphrase_e(POSTPOSPHRASE_E_TREE).
```

```
/*339*/ adv(adv(PARTICLE_A_TREE,POSTPOSPHRASE_C_TREE,POSTPOSPHRASE_D_TREE,P
           OSTPOSPHRASE_F_TREE)) --> particle_a(PARTICLE_A_TREE),[x],postp
           osphrase_c(POSTPOSPHRASE_C_TREE),[x],postposphrase_d(POSTPOSPHR
           ASE_D_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_TREE).

/*340*/ adv(adv(PARTICLE_A_TREE,POSTPOSPHRASE_C_TREE,POSTPOSPHRASE_E_TREE,P
           OSTPOSPHRASE_F_TREE)) --> particle_a(PARTICLE_A_TREE),[x],postp
           osphrase_c(POSTPOSPHRASE_C_TREE),[x],postposphrase_e(POSTPOSPHR
           ASE_E_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_TREE).

/*341*/ adv(adv(PARTICLE_B_TREE)) --> particle_b(PARTICLE_B_TREE).

/*342*/ adv(adv(PARTICLE_B_TREE,POSTPOSPHRASE_F_TREE)) --> particle_b(PARTI
           CLE_B_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_TREE).

/*343*/ adv(adv(PARTICLE_B_TREE,POSTPOSPHRASE_D_TREE)) --> particle_b(PARTI
           CLE_B_TREE),[x],postposphrase_d(POSTPOSPHRASE_D_TREE).

/*344*/ adv(adv(PARTICLE_B_TREE,POSTPOSPHRASE_E_TREE)) --> particle_b(PARTI
           CLE_B_TREE),[x],postposphrase_e(POSTPOSPHRASE_E_TREE).

/*345*/ adv(adv(PARTICLE_B_TREE,POSTPOSPHRASE_D_TREE,POSTPOSPHRASE_F_TREE))
            --> particle_b(PARTICLE_B_TREE),[x],postposphrase_d(POSTPOSPHR
           ASE_D_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_TREE).

/*346*/ adv(adv(PARTICLE_B_TREE,POSTPOSPHRASE_E_TREE,POSTPOSPHRASE_F_TREE))
            --> particle_b(PARTICLE_B_TREE),[x],postposphrase_e(POSTPOSPHR
           ASE_E_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_TREE).

/*347*/ adv(adv(PARTICLE_B_TREE,POSTPOSPHRASE_A_TREE)) --> particle_b(PARTI
           CLE_B_TREE),[x],postposphrase_a(POSTPOSPHRASE_A_TREE).

/*348*/ adv(adv(PARTICLE_B_TREE,POSTPOSPHRASE_F_TREE,POSTPOSPHRASE_F_TREE))
            --> particle_b(PARTICLE_B_TREE),[x],postposphrase_f(POSTPOSPHR
           ASE_F_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_TREE).

/*349*/ adv(adv(PARTICLE_B_TREE,POSTPOSPHRASE_A_TREE,POSTPOSPHRASE_D_TREE))
            --> particle_b(PARTICLE_B_TREE),[x],postposphrase_a(POSTPOSPHR
           ASE_A_TREE),[x],postposphrase_d(POSTPOSPHRASE_D_TREE).

/*350*/ adv(adv(PARTICLE_B_TREE,POSTPOSPHRASE_A_TREE,POSTPOSPHRASE_E_TREE))
            --> particle_b(PARTICLE_B_TREE),[x],postposphrase_a(POSTPOSPHR
           ASE_A_TREE),[x],postposphrase_e(POSTPOSPHRASE_E_TREE).
```

```
/*351*/ adv(adv(PARTICLE_B_TREE,POSTPOSPHRASE_A_TREE,POSTPOSPHRASE_D_TREE,P
            OSTPOSPHRASE_F_TREE)) --> particle_b(PARTICLE_B_TREE),[x],postp
            osphrase_a(POSTPOSPHRASE_A_TREE),[x],postposphrase_d(POSTPOSPHR
            ASE_D_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_TREE).

/*352*/ adv(adv(PARTICLE_B_TREE,POSTPOSPHRASE_A_TREE,POSTPOSPHRASE_E_TREE,P
            OSTPOSPHRASE_F_TREE)) --> particle_b(PARTICLE_B_TREE),[x],postp
            osphrase_a(POSTPOSPHRASE_A_TREE),[x],postposphrase_e(POSTPOSPHR
            ASE_E_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_TREE).

/*353*/ adv(adv(PARTICLE_B_TREE,POSTPOSPHRASE_B_TREE)) --> particle_b(PARTI
            CLE_B_TREE),[x],postposphrase_b(POSTPOSPHRASE_B_TREE),[x].

/*354*/ adv(adv(PARTICLE_B_TREE,POSTPOSPHRASE_B_TREE,POSTPOSPHRASE_F_TREE))
             --> particle_b(PARTICLE_B_TREE),[x],postposphrase_b(POSTPOSPHR
            ASE_B_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_TREE).

/*355*/ adv(adv(PARTICLE_B_TREE,POSTPOSPHRASE_B_TREE,POSTPOSPHRASE_D_TREE))
             --> particle_b(PARTICLE_B_TREE),[x],postposphrase_b(POSTPOSPHR
            ASE_B_TREE),[x],postposphrase_d(POSTPOSPHRASE_D_TREE).

/*356*/ adv(adv(PARTICLE_B_TREE,POSTPOSPHRASE_B_TREE,POSTPOSPHRASE_E_TREE))
             --> particle_b(PARTICLE_B_TREE),[x],postposphrase_b(POSTPOSPHR
            ASE_B_TREE),[x],postposphrase_e(POSTPOSPHRASE_E_TREE).

/*357*/ adv(adv(PARTICLE_B_TREE,POSTPOSPHRASE_B_TREE,POSTPOSPHRASE_D_TREE,P
            OSTPOSPHRASE_F_TREE)) --> particle_b(PARTICLE_B_TREE),[x],postp
            osphrase_b(POSTPOSPHRASE_B_TREE),[x],postposphrase_d(POSTPOSPHR
            ASE_D_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_TREE).

/*358*/ adv(adv(PARTICLE_B_TREE,POSTPOSPHRASE_B_TREE,POSTPOSPHRASE_E_TREE,P
            OSTPOSPHRASE_F_TREE)) --> particle_b(PARTICLE_B_TREE),[x],postp
            osphrase_b(POSTPOSPHRASE_B_TREE),[x],postposphrase_e(POSTPOSPHR
            ASE_E_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_TREE).

/*359*/ adv(adv(PARTICLE_B_TREE,POSTPOSPHRASE_C_TREE)) --> particle_b(PARTI
            CLE_B_TREE),[x],postposphrase_c(POSTPOSPHRASE_C_TREE),[x].

/*360*/ adv(adv(PARTICLE_B_TREE,POSTPOSPHRASE_C_TREE,POSTPOSPHRASE_F_TREE))
             --> particle_b(PARTICLE_B_TREE),[x],postposphrase_c(POSTPOSPHR
            ASE_C_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_TREE).
```

```
/*361*/ adv(adv(PARTICLE_B_TREE,POSTPOSPHRASE_C_TREE,POSTPOSPHRASE_D_TREE))
            --> particle_b(PARTICLE_B_TREE),[x],postposphrase_c(POSTPOSPHR
          ASE_C_TREE),[x],postposphrase_d(POSTPOSPHRASE_D_TREE).

/*362*/ adv(adv(PARTICLE_B_TREE,POSTPOSPHRASE_C_TREE,POSTPOSPHRASE_E_TREE))
            --> particle_b(PARTICLE_B_TREE),[x],postposphrase_c(POSTPOSPHR
          ASE_C_TREE),[x],postposphrase_e(POSTPOSPHRASE_E_TREE).

/*363*/ adv(adv(PARTICLE_B_TREE,POSTPOSPHRASE_C_TREE,POSTPOSPHRASE_D_TREE,P
          OSTPOSPHRASE_F_TREE)) --> particle_b(PARTICLE_B_TREE),[x],postp
          osphrase_c(POSTPOSPHRASE_C_TREE),[x],postposphrase_d(POSTPOSPHR
          ASE_D_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_TREE).

/*364*/ adv(adv(PARTICLE_B_TREE,POSTPOSPHRASE_C_TREE,POSTPOSPHRASE_E_TREE,P
          OSTPOSPHRASE_F_TREE)) --> particle_b(PARTICLE_B_TREE),[x],postp
          osphrase_c(POSTPOSPHRASE_C_TREE),[x],postposphrase_e(POSTPOSPHR
          ASE_E_TREE),[x],postposphrase_f(POSTPOSPHRASE_F_TREE).

/*365*/ postposphrase_a(postposphrase_a(POSTPOSITION_A_TREE)) --> postposit
          ion_a(POSTPOSITION_A_TREE).

/*366*/ postposphrase_a(postposphrase_a(NP_TREE,POSTPOSITION_A_TREE)) --> n
          p(NP_TREE),[den],[x],postposition_a(POSTPOSITION_A_TREE).

/*367*/ postposphrase_b(postposphrase_b(NP_TREE,POSTPOSITION_B_TREE)) --> n
          p(NP_TREE),[den],[x],postposition_b(POSTPOSITION_B_TREE).

/*368*/ postposphrase_c(postposphrase_c(NP_TREE,POSTPOSITION_C_TREE)) --> n
          p(NP_TREE),[den],[x],postposition_c(POSTPOSITION_C_TREE).

/*369*/ postposphrase_d(postposphrase_d(POSTPOSITION_D_TREE)) --> postposit
          ion_d(POSTPOSITION_D_TREE).

/*370*/ postposphrase_d(postposphrase_d(NP_TREE,POSTPOSITION_D_TREE)) --> n
          p(NP_TREE),[e],[x],postposition_d(POSTPOSITION_D_TREE).

/*371*/ postposphrase_e(postposphrase_e(NP_TREE,POSTPOSITION_E_TREE)) --> n
          p(NP_TREE),[e],[x],postposition_e(POSTPOSITION_E_TREE).

/*372*/ postposphrase_f(postposphrase_f(NP_TREE,POSTPOSITION_F_TREE)) --> n
          p(NP_TREE),[x],postposition_f(POSTPOSITION_F_TREE).
```

```
/*373*/ np_time(np_time(N_TIME_A_TREE)) --> n_time_a(N_TIME_A_TREE).

/*374*/ np_time(np_time(N_TIME_B_TREE)) --> n_time_b(N_TIME_B_TREE).

/*375*/ np_abstract(np_abstract(N_ABSTRACT_TREE)) --> n_abstract(N_ABSTRACT
           _TREE).

/*376*/ np_abstract(np_abstract(DEMONSTRATIVE_TREE,N_ABSTRACT_TREE)) --> de
           monstrative(DEMONSTRATIVE_TREE),[x],n_abstract(N_ABSTRACT_TREE)
           .

/*377*/ np_ins(np_ins(N_ITEM_TREE)) --> n_item(N_ITEM_TREE).

/*378*/ np_ins(np_ins(N_LOCALE_TREE)) --> n_locale(N_LOCALE_TREE).

/*379*/ np_ins(np_ins(N_LOCALE_PROPER_TREE)) --> n_locale_proper(N_LOCALE_P
           ROPER_TREE).

/*380*/ np_ins(np_ins(N_NUMBER_TREE)) --> n_number(N_NUMBER_TREE).

/*381*/ np_ins(np_ins(N_SUBSTANCE_TREE)) --> n_substance(N_SUBSTANCE_TREE).


/*382*/ np_ins(np_ins(N_INANIMATE_TREE)) --> n_inanimate(N_INANIMATE_TREE).


/*383*/ np_ins(np_ins(N_ANIMATE_TRANSPORT_TREE)) --> n_animate_transport(N_
           ANIMATE_TRANSPORT_TREE).

/*384*/ np_ins(np_ins(N_NUMBER_TREE,N_ITEM_TREE)) --> n_number(N_NUMBER_TRE
           E),[x],n_item(N_ITEM_TREE).

/*385*/ np_ins(np_ins(N_NUMBER_TREE,N_LOCALE_TREE)) --> n_number(N_NUMBER_T
           REE),[x],n_locale(N_LOCALE_TREE).

/*386*/ np_ins(np_ins(N_NUMBER_TREE,N_LOCALE_PROPER_TREE)) --> n_number(N_N
           UMBER_TREE),[x],n_locale_proper(N_LOCALE_PROPER_TREE).

/*387*/ np_ins(np_ins(N_NUMBER_TREE,N_NUMBER_TREE)) --> n_number(N_NUMBER_T
           REE),[x],n_number(N_NUMBER_TREE).
```

```
/*388*/ np_ins(np_ins(N_NUMBER_TREE,N_SUBSTANCE_TREE)) --> n_number(N_NUMBE
           R_TREE),[x],n_substance(N_SUBSTANCE_TREE).

/*389*/ np_ins(np_ins(N_NUMBER_TREE,N_INANIMATE_TREE)) --> n_number(N_NUMBE
           R_TREE),[x],n_inanimate(N_INANIMATE_TREE).

/*390*/ np_ins(np_ins(N_NUMBER_TREE,N_ANIMATE_TRANSPORT_TREE)) --> n_number
           (N_NUMBER_TREE),[x],n_animate_transport(N_ANIMATE_TRANSPORT_TRE
           E).

/*391*/ np_ins(np_ins(DEMONSTRATIVE_TREE,N_ITEM_TREE)) --> demonstrative(DE
           MONSTRATIVE_TREE),[x],n_item(N_ITEM_TREE).

/*392*/ np_ins(np_ins(DEMONSTRATIVE_TREE,N_LOCALE_TREE)) --> demonstrative(
           DEMONSTRATIVE_TREE),[x],n_locale(N_LOCALE_TREE).

/*393*/ np_ins(np_ins(DEMONSTRATIVE_TREE,N_LOCALE_PROPER_TREE)) --> demonst
           rative(DEMONSTRATIVE_TREE),[x],n_locale_proper(N_LOCALE_PROPER_
           TREE).

/*394*/ np_ins(np_ins(DEMONSTRATIVE_TREE,N_NUMBER_TREE)) --> demonstrative(
           DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_TREE).

/*395*/ np_ins(np_ins(DEMONSTRATIVE_TREE,N_SUBSTANCE_TREE)) --> demonstrati
           ve(DEMONSTRATIVE_TREE),[x],n_substance(N_SUBSTANCE_TREE).

/*396*/ np_ins(np_ins(DEMONSTRATIVE_TREE,N_INANIMATE_TREE)) --> demonstrati
           ve(DEMONSTRATIVE_TREE),[x],n_inanimate(N_INANIMATE_TREE).

/*397*/ np_ins(np_ins(DEMONSTRATIVE_TREE,N_ANIMATE_TRANSPORT_TREE)) --> dem
           onstrative(DEMONSTRATIVE_TREE),[x],n_animate_transport(N_ANIMAT
           E_TRANSPORT_TREE).

/*398*/ np_ins(np_ins(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_ITEM_TREE)) --> de
           monstrative(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_TREE),[x]
           ,n_item(N_ITEM_TREE).

/*399*/ np_ins(np_ins(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_LOCALE_TREE)) -->
           demonstrative(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_TREE),[
           x],n_locale(N_LOCALE_TREE).
```

```
/*400*/ np_ins(np_ins(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_LOCALE_PROPER_TREE
            )) --> demonstrative(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_
            TREE),[x],n_locale_proper(N_LOCALE_PROPER_TREE).

/*401*/ np_ins(np_ins(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_NUMBER_TREE)) -->
            demonstrative(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_TREE),[
            x],n_number(N_NUMBER_TREE).

/*402*/ np_ins(np_ins(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_SUBSTANCE_TREE)) -
            -> demonstrative(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_TREE
            ),[x],n_substance(N_SUBSTANCE_TREE).

/*403*/ np_ins(np_ins(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_INANIMATE_TREE)) -
            -> demonstrative(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_TREE
            ),[x],n_inanimate(N_INANIMATE_TREE).

/*404*/ np_ins(np_ins(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_ANIMATE_TRANSPORT_
            TREE)) --> demonstrative(DEMONSTRATIVE_TREE),[x],n_number(N_NUM
            BER_TREE),[x],n_animate_transport(N_ANIMATE_TRANSPORT_TREE).

/*405*/ np_mob2(np_mob2(N_ITEM_TREE)) --> n_item(N_ITEM_TREE).

/*406*/ np_mob2(np_mob2(N_LOCALE_TREE)) --> n_locale(N_LOCALE_TREE).

/*407*/ np_mob2(np_mob2(N_LOCALE_PROPER_TREE)) --> n_locale_proper(N_LOCALE
            _PROPER_TREE).

/*408*/ np_mob2(np_mob2(N_NUMBER_TREE)) --> n_number(N_NUMBER_TREE).

/*409*/ np_mob2(np_mob2(N_SUBSTANCE_TREE)) --> n_substance(N_SUBSTANCE_TREE
            ).

/*410*/ np_mob2(np_mob2(N_INANIMATE_TREE)) --> n_inanimate(N_INANIMATE_TREE
            ).

/*411*/ np_mob2(np_mob2(N_NUMBER_TREE,N_ITEM_TREE)) --> n_number(N_NUMBER_T
            REE),[x],n_item(N_ITEM_TREE).

/*412*/ np_mob2(np_mob2(N_NUMBER_TREE,N_LOCALE_TREE)) --> n_number(N_NUMBER
            _TREE),[x],n_locale(N_LOCALE_TREE).
```

```
/*413*/ np_mob2(np_mob2(N_NUMBER_TREE,N_LOCALE_PROPER_TREE)) --> n_number(N
            _NUMBER_TREE),[x],n_locale_proper(N_LOCALE_PROPER_TREE).

/*414*/ np_mob2(np_mob2(N_NUMBER_TREE,N_NUMBER_TREE)) --> n_number(N_NUMBER
            _TREE),[x],n_number(N_NUMBER_TREE).

/*415*/ np_mob2(np_mob2(N_NUMBER_TREE,N_SUBSTANCE_TREE)) --> n_number(N_NUM
            BER_TREE),[x],n_substance(N_SUBSTANCE_TREE).

/*416*/ np_mob2(np_mob2(N_NUMBER_TREE,N_INANIMATE_TREE)) --> n_number(N_NUM
            BER_TREE),[x],n_inanimate(N_INANIMATE_TREE).

/*417*/ np_mob2(np_mob2(DEMONSTRATIVE_TREE,N_ITEM_TREE)) --> demonstrative(
            DEMONSTRATIVE_TREE),[x],n_item(N_ITEM_TREE).

/*418*/ np_mob2(np_mob2(DEMONSTRATIVE_TREE,N_LOCALE_TREE)) --> demonstrativ
            e(DEMONSTRATIVE_TREE),[x],n_locale(N_LOCALE_TREE).

/*419*/ np_mob2(np_mob2(DEMONSTRATIVE_TREE,N_LOCALE_PROPER_TREE)) --> demon
            strative(DEMONSTRATIVE_TREE),[x],n_locale_proper(N_LOCALE_PROPE
            R_TREE).

/*420*/ np_mob2(np_mob2(DEMONSTRATIVE_TREE,N_NUMBER_TREE)) --> demonstrativ
            e(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_TREE).

/*421*/ np_mob2(np_mob2(DEMONSTRATIVE_TREE,N_SUBSTANCE_TREE)) --> demonstra
            tive(DEMONSTRATIVE_TREE),[x],n_substance(N_SUBSTANCE_TREE).

/*422*/ np_mob2(np_mob2(DEMONSTRATIVE_TREE,N_INANIMATE_TREE)) --> demonstra
            tive(DEMONSTRATIVE_TREE),[x],n_inanimate(N_INANIMATE_TREE).

/*423*/ np_mob2(np_mob2(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_ITEM_TREE)) -->
            demonstrative(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_TREE),[
            x],n_item(N_ITEM_TREE).

/*424*/ np_mob2(np_mob2(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_LOCALE_TREE)) --
            > demonstrative(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_TREE)
            ,[x],n_locale(N_LOCALE_TREE).

/*425*/ np_mob2(np_mob2(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_LOCALE_PROPER_TR
            EE)) --> demonstrative(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBE
            R_TREE),[x],n_locale_proper(N_LOCALE_PROPER_TREE).
```

```
/*426*/ np_mob2(np_mob2(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_NUMBER_TREE)) --
            > demonstrative(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_TREE)
            ,[x],n_number(N_NUMBER_TREE).

/*427*/ np_mob2(np_mob2(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_SUBSTANCE_TREE))
             --> demonstrative(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_TR
            EE),[x],n_substance(N_SUBSTANCE_TREE).

/*428*/ np_mob2(np_mob2(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_INANIMATE_TREE))
             --> demonstrative(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_TR
            EE),[x],n_inanimate(N_INANIMATE_TREE).

/*429*/ np_human(np_human(N_ANIMATE_HUMAN_TREE)) --> n_animate_human(N_ANIM
            ATE_HUMAN_TREE).

/*430*/ np_human(np_human(N_ANIMATE_PROPER_HUMAN_TREE)) --> n_animate_prope
            r_human(N_ANIMATE_PROPER_HUMAN_TREE).

/*431*/ np_human(np_human(N_NUMBER_TREE,N_ANIMATE_HUMAN_TREE)) --> n_number
            (N_NUMBER_TREE),[x],n_animate_human(N_ANIMATE_HUMAN_TREE).

/*432*/ np_human(np_human(N_NUMBER_TREE,N_ANIMATE_PROPER_HUMAN_TREE)) --> n
            _number(N_NUMBER_TREE),[x],n_animate_proper_human(N_ANIMATE_PRO
            PER_HUMAN_TREE).

/*433*/ np_human(np_human(DEMONSTRATIVE_TREE,N_ANIMATE_HUMAN_TREE)) --> dem
            onstrative(DEMONSTRATIVE_TREE),[x],n_animate_human(N_ANIMATE_HU
            MAN_TREE).

/*434*/ np_human(np_human(DEMONSTRATIVE_TREE,N_ANIMATE_PROPER_HUMAN_TREE))
            --> demonstrative(DEMONSTRATIVE_TREE),[x],n_animate_proper_huma
            n(N_ANIMATE_PROPER_HUMAN_TREE).

/*435*/ np_human(np_human(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_ANIMATE_HUMAN_
            TREE)) --> demonstrative(DEMONSTRATIVE_TREE),[x],n_number(N_NUM
            BER_TREE),[x],n_animate_human(N_ANIMATE_HUMAN_TREE).

/*436*/ np_human(np_human(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_ANIMATE_PROPER
            _HUMAN_TREE)) --> demonstrative(DEMONSTRATIVE_TREE),[x],n_numbe
            r(N_NUMBER_TREE),[x],n_animate_proper_human(N_ANIMATE_PROPER_HU
            MAN_TREE).
```

```
/*437*/ np_human(np_human(ben)) --> [ben].

/*438*/ np_human(np_human(sen)) --> [sen].

/*439*/ np_human(np_human(o)) --> [o].

/*440*/ np_mob(np_mob(N_ANIMATE_TREE)) --> n_animate(N_ANIMATE_TREE).

/*441*/ np_mob(np_mob(N_ANIMATE_PROPER_TREE)) --> n_animate_proper(N_ANIMAT
            E_PROPER_TREE).

/*442*/ np_mob(np_mob(N_ANIMATE_TRANSPORT_TREE)) --> n_animate_transport(N_
            ANIMATE_TRANSPORT_TREE).

/*443*/ np_mob(np_mob(N_INANIMATE_TREE)) --> n_inanimate(N_INANIMATE_TREE).


/*444*/ np_mob(np_mob(N_NUMBER_TREE,N_ANIMATE_TREE)) --> n_number(N_NUMBER_
            TREE),[x],n_animate(N_ANIMATE_TREE).

/*445*/ np_mob(np_mob(N_NUMBER_TREE,N_ANIMATE_PROPER_TREE)) --> n_number(N_
            NUMBER_TREE),[x],n_animate_proper(N_ANIMATE_PROPER_TREE).

/*446*/ np_mob(np_mob(N_NUMBER_TREE,N_ANIMATE_TRANSPORT_TREE)) --> n_number
            (N_NUMBER_TREE),[x],n_animate_transport(N_ANIMATE_TRANSPORT_TRE
            E).

/*447*/ np_mob(np_mob(N_NUMBER_TREE,N_INANIMATE_TREE)) --> n_number(N_NUMBE
            R_TREE),[x],n_inanimate(N_INANIMATE_TREE).

/*448*/ np_mob(np_mob(DEMONSTRATIVE_TREE,N_ANIMATE_TREE)) --> demonstrative
            (DEMONSTRATIVE_TREE),[x],n_animate(N_ANIMATE_TREE).

/*449*/ np_mob(np_mob(DEMONSTRATIVE_TREE,N_ANIMATE_PROPER_TREE)) --> demons
            trative(DEMONSTRATIVE_TREE),[x],n_animate_proper(N_ANIMATE_PROP
            ER_TREE).

/*450*/ np_mob(np_mob(DEMONSTRATIVE_TREE,N_ANIMATE_TRANSPORT_TREE)) --> dem
            onstrative(DEMONSTRATIVE_TREE),[x],n_animate_transport(N_ANIMAT
            E_TRANSPORT_TREE).
```

```
/*451*/ np_mob(np_mob(DEMONSTRATIVE_TREE,N_INANIMATE_TREE)) --> demonstrati
            ve(DEMONSTRATIVE_TREE),[x],n_inanimate(N_INANIMATE_TREE).

/*452*/ np_mob(np_mob(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_ANIMATE_TREE)) -->
             demonstrative(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_TREE),
            [x],n_animate(N_ANIMATE_TREE).

/*453*/ np_mob(np_mob(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_ANIMATE_PROPER_TRE
            E)) --> demonstrative(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER
            _TREE),[x],n_animate_proper(N_ANIMATE_PROPER_TREE).

/*454*/ np_mob(np_mob(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_ANIMATE_TRANSPORT_
            TREE)) --> demonstrative(DEMONSTRATIVE_TREE),[x],n_number(N_NUM
            BER_TREE),[x],n_animate_transport(N_ANIMATE_TRANSPORT_TREE).

/*455*/ np_mob(np_mob(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_INANIMATE_TREE)) -
            -> demonstrative(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_TREE
            ),[x],n_inanimate(N_INANIMATE_TREE).

/*456*/ np_mob(np_mob(PRONOUN_TREE)) --> pronoun(PRONOUN_TREE).

/*457*/ np(np(PRONOUN_TREE)) --> pronoun(PRONOUN_TREE).

/*458*/ np(np(bura)) --> [bura].

/*459*/ np(np(ura)) --> [ura].

/*460*/ np(np(ora)) --> [ora].

/*461*/ np(np(nere)) --> [nere].

/*462*/ np(np(N_TIME_A_TREE)) --> n_time_a(N_TIME_A_TREE).

/*463*/ np(np(N_TIME_B_TREE)) --> n_time_b(N_TIME_B_TREE).

/*464*/ np(np(N_ABSTRACT_TREE)) --> n_abstract(N_ABSTRACT_TREE).

/*465*/ np(np(DEMONSTRATIVE_TREE,N_ABSTRACT_TREE)) --> demonstrative(DEMONS
            TRATIVE_TREE),[x],n_abstract(N_ABSTRACT_TREE).

/*466*/ np(np(N_ANIMATE_TREE)) --> n_animate(N_ANIMATE_TREE).
```

```
/*467*/ np(np(N_ANIMATE_HUMAN_TREE)) --> n_animate_human(N_ANIMATE_HUMAN_TR
           EE).

/*468*/ np(np(N_ANIMATE_PROPER_TREE)) --> n_animate_proper(N_ANIMATE_PROPER
           _TREE).

/*469*/ np(np(N_ANIMATE_PROPER_HUMAN_TREE)) --> n_animate_proper_human(N_AN
           IMATE_PROPER_HUMAN_TREE).

/*470*/ np(np(N_ANIMATE_TRANSPORT_TREE)) --> n_animate_transport(N_ANIMATE_
           TRANSPORT_TREE).

/*471*/ np(np(N_INANIMATE_TREE)) --> n_inanimate(N_INANIMATE_TREE).

/*472*/ np(np(N_ITEM_TREE)) --> n_item(N_ITEM_TREE).

/*473*/ np(np(N_LOCALE_TREE)) --> n_locale(N_LOCALE_TREE).

/*474*/ np(np(N_LOCALE_PROPER_TREE)) --> n_locale_proper(N_LOCALE_PROPER_TR
           EE).

/*475*/ np(np(N_NUMBER_TREE)) --> n_number(N_NUMBER_TREE).

/*476*/ np(np(N_SUBSTANCE_TREE)) --> n_substance(N_SUBSTANCE_TREE).

/*477*/ np(np(N_NUMBER_TREE,N_ANIMATE_TREE)) --> n_number(N_NUMBER_TREE),[x
           ],n_animate(N_ANIMATE_TREE).

/*478*/ np(np(N_NUMBER_TREE,N_ANIMATE_HUMAN_TREE)) --> n_number(N_NUMBER_TR
           EE),[x],n_animate_human(N_ANIMATE_HUMAN_TREE).

/*479*/ np(np(N_NUMBER_TREE,N_ANIMATE_PROPER_TREE)) --> n_number(N_NUMBER_T
           REE),[x],n_animate_proper(N_ANIMATE_PROPER_TREE).

/*480*/ np(np(N_NUMBER_TREE,N_ANIMATE_PROPER_HUMAN_TREE)) --> n_number(N_NU
           MBER_TREE),[x],n_animate_proper_human(N_ANIMATE_PROPER_HUMAN_TR
           EE).

/*481*/ np(np(N_NUMBER_TREE,N_ANIMATE_TRANSPORT_TREE)) --> n_number(N_NUMBE
           R_TREE),[x],n_animate_transport(N_ANIMATE_TRANSPORT_TREE).
```

```
/*482*/ np(np(N_NUMBER_TREE,N_INANIMATE_TREE)) --> n_number(N_NUMBER_TREE),
            [x],n_inanimate(N_INANIMATE_TREE).

/*483*/ np(np(N_NUMBER_TREE,N_ITEM_TREE)) --> n_number(N_NUMBER_TREE),[x],n
            _item(N_ITEM_TREE).

/*484*/ np(np(N_NUMBER_TREE,N_LOCALE_TREE)) --> n_number(N_NUMBER_TREE),[x]
            ,n_locale(N_LOCALE_TREE).

/*485*/ np(np(N_NUMBER_TREE,N_LOCALE_PROPER_TREE)) --> n_number(N_NUMBER_TR
            EE),[x],n_locale_proper(N_LOCALE_PROPER_TREE).

/*486*/ np(np(N_NUMBER_TREE,N_NUMBER_TREE)) --> n_number(N_NUMBER_TREE),[x]
            ,n_number(N_NUMBER_TREE).

/*487*/ np(np(N_NUMBER_TREE,N_SUBSTANCE_TREE)) --> n_number(N_NUMBER_TREE),
            [x],n_substance(N_SUBSTANCE_TREE).

/*488*/ np(np(DEMONSTRATIVE_TREE,N_ANIMATE_TREE)) --> demonstrative(DEMONST
            RATIVE_TREE),[x],n_animate(N_ANIMATE_TREE).

/*489*/ np(np(DEMONSTRATIVE_TREE,N_ANIMATE_HUMAN_TREE)) --> demonstrative(D
            EMONSTRATIVE_TREE),[x],n_animate_human(N_ANIMATE_HUMAN_TREE).

/*490*/ np(np(DEMONSTRATIVE_TREE,N_ANIMATE_PROPER_TREE)) --> demonstrative(
            DEMONSTRATIVE_TREE),[x],n_animate_proper(N_ANIMATE_PROPER_TREE)
            .

/*491*/ np(np(DEMONSTRATIVE_TREE,N_ANIMATE_PROPER_HUMAN_TREE)) --> demonstr
            ative(DEMONSTRATIVE_TREE),[x],n_animate_proper_human(N_ANIMATE_
            PROPER_HUMAN_TREE).

/*492*/ np(np(DEMONSTRATIVE_TREE,N_ANIMATE_TRANSPORT_TREE)) --> demonstrati
            ve(DEMONSTRATIVE_TREE),[x],n_animate_transport(N_ANIMATE_TRANSP
            ORT_TREE).

/*493*/ np(np(DEMONSTRATIVE_TREE,N_INANIMATE_TREE)) --> demonstrative(DEMON
            STRATIVE_TREE),[x],n_inanimate(N_INANIMATE_TREE).

/*494*/ np(np(DEMONSTRATIVE_TREE,N_ITEM_TREE)) --> demonstrative(DEMONSTRAT
            IVE_TREE),[x],n_item(N_ITEM_TREE).
```

```
/*495*/ np(np(DEMONSTRATIVE_TREE,N_LOCALE_TREE)) --> demonstrative(DEMONSTR
            ATIVE_TREE),[x],n_locale(N_LOCALE_TREE).

/*496*/ np(np(DEMONSTRATIVE_TREE,N_LOCALE_PROPER_TREE)) --> demonstrative(D
            EMONSTRATIVE_TREE),[x],n_locale_proper(N_LOCALE_PROPER_TREE).

/*497*/ np(np(DEMONSTRATIVE_TREE,N_NUMBER_TREE)) --> demonstrative(DEMONSTR
            ATIVE_TREE),[x],n_number(N_NUMBER_TREE).

/*498*/ np(np(DEMONSTRATIVE_TREE,N_SUBSTANCE_TREE)) --> demonstrative(DEMON
            STRATIVE_TREE),[x],n_substance(N_SUBSTANCE_TREE).

/*499*/ np(np(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_ANIMATE_TREE)) --> demonst
            rative(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_TREE),[x],n_an
            imate(N_ANIMATE_TREE).

/*500*/ np(np(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_ANIMATE_HUMAN_TREE)) --> d
            emonstrative(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_TREE),[x
            ],n_animate_human(N_ANIMATE_HUMAN_TREE).

/*501*/ np(np(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_ANIMATE_PROPER_TREE)) -->
            demonstrative(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_TREE),[
            x],n_animate_proper(N_ANIMATE_PROPER_TREE).

/*502*/ np(np(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_ANIMATE_PROPER_HUMAN_TREE)
            ) --> demonstrative(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_T
            REE),[x],n_animate_proper_human(N_ANIMATE_PROPER_HUMAN_TREE).

/*503*/ np(np(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_ANIMATE_TRANSPORT_TREE)) -
            -> demonstrative(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_TREE
            ),[x],n_animate_transport(N_ANIMATE_TRANSPORT_TREE).

/*504*/ np(np(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_INANIMATE_TREE)) --> demon
            strative(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_TREE),[x],n_
            inanimate(N_INANIMATE_TREE).

/*505*/ np(np(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_ITEM_TREE)) --> demonstrat
            ive(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_TREE),[x],n_item(
            N_ITEM_TREE).

/*506*/ np(np(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_LOCALE_TREE)) --> demonstr
            ative(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_TREE),[x],n_loc
            ale(N_LOCALE_TREE).
```

```
/*507*/ np(np(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_LOCALE_PROPER_TREE)) --> d
            emonstrative(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_TREE),[x
            ],n_locale_proper(N_LOCALE_PROPER_TREE).

/*508*/ np(np(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_NUMBER_TREE)) --> demonstr
            ative(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_TREE),[x],n_num
            ber(N_NUMBER_TREE).

/*509*/ np(np(DEMONSTRATIVE_TREE,N_NUMBER_TREE,N_SUBSTANCE_TREE)) --> demon
            strative(DEMONSTRATIVE_TREE),[x],n_number(N_NUMBER_TREE),[x],n_
            substance(N_SUBSTANCE_TREE).

/*510*/ vp_n(vp_n(ADJUNCTADJ_D_TREE)) --> adjunctadj_d(ADJUNCTADJ_D_TREE).

/*511*/ copula(copula(dir)) -->[dir].

/*512*/ adj_a(adj_a(cazip)) -->[cazip].

/*513*/ adj_a(adj_a(gzel)) -->[gzel].

/*514*/ adj_a(adj_a(iyi)) -->[iyi].

/*515*/ adj_a(adj_a(var)) -->[var].

/*516*/ adj_a(adj_a(zor)) -->[zor].

/*517*/ adj_b(adj_b(az)) -->[az].

/*518*/ adj_b(adj_b(ok)) -->[ok].

/*519*/ adj_b(adj_b(fazla)) -->[fazla].

/*520*/ adj_b(adj_b(pek)) -->[pek].

/*521*/ adj_c(adj_c(emin)) -->[emin].

/*522*/ adj_c(adj_c(rak)) -->[rak].

/*523*/ adj_c(adj_c(masun)) -->[masun].

/*524*/ adj_c(adj_c(memnun)) -->[memnun].

/*525*/ adj_c(adj_c(raz)) -->[raz].
```

```
/*526*/ adj_d(adj_d(megul)) -->[megul].

/*527*/ demonstrative(demonstrative(bu)) -->[bu].

/*528*/ demonstrative(demonstrative(u)) -->[u].

/*529*/ demonstrative(demonstrative(o)) -->[o].

/*530*/ n_abstract(n_abstract(cesaret)) -->[cesaret].

/*531*/ n_abstract(n_abstract(hararet)) -->[hararet].

/*532*/ n_abstract(n_abstract(heves)) -->[heves].

/*533*/ n_abstract(n_abstract(memnuniyet)) -->[memnuniyet].

/*534*/ n_abstract(n_abstract(iddet)) -->[iddet].

/*535*/ n_animate(n_animate(kedi)) -->[kedi].

/*536*/ n_animate(n_animate(kpek)) -->[kpek].

/*537*/ n_animate(n_animate(ku)) -->[ku].

/*538*/ n_animate(n_animate(kuzu)) -->[kuzu].

/*539*/ n_animate(n_animate(maymun)) -->[maymun].

/*540*/ n_animate_human(n_animate_human(adam)) -->[adam].

/*541*/ n_animate_human(n_animate_human(kadn)) -->[kadn].

/*542*/ n_animate_human(n_animate_human(kz)) -->[kz].

/*543*/ n_animate_human(n_animate_human(memur)) -->[memur].

/*544*/ n_animate_human(n_animate_human(talebe)) -->[talebe].

/*545*/ n_animate_proper(n_animate_proper(bobi)) -->[bobi].

/*546*/ n_animate_proper(n_animate_proper(mavi)) -->[mavi].

/*547*/ n_animate_proper(n_animate_proper(sarman)) -->[sarman].
```

```
/*548*/ n_animate_proper(n_animate_proper(tekir)) -->[tekir].

/*549*/ n_animate_proper_human(n_animate_proper_human(ahmet)) -->[ahmet].

/*550*/ n_animate_proper_human(n_animate_proper_human(glen)) -->[glen].

/*551*/ n_animate_proper_human(n_animate_proper_human(mehmet)) -->[mehmet].

/*552*/ n_animate_proper_human(n_animate_proper_human(oktay)) -->[oktay].

/*553*/ n_animate_proper_human(n_animate_proper_human(ermin)) -->[ermin].

/*554*/ n_animate_transport(n_animate_transport(at)) -->[at].

/*555*/ n_animate_transport(n_animate_transport(deve)) -->[deve].

/*556*/ n_animate_transport(n_animate_transport(eek)) -->[eek].

/*557*/ n_animate_transport(n_animate_transport(fil)) -->[fil].

/*558*/ n_animate_transport(n_animate_transport(manda)) -->[manda].

/*559*/ n_inanimate(n_inanimate(araba)) -->[araba].

/*560*/ n_inanimate(n_inanimate(otobs)) -->[otobs].

/*561*/ n_inanimate(n_inanimate(ses)) -->[ses].

/*562*/ n_inanimate(n_inanimate(tren)) -->[tren].

/*563*/ n_item(n_item(kalem)) -->[kalem].

/*564*/ n_item(n_item(masa)) -->[masa].

/*565*/ n_item(n_item(mektup)) -->[mektup].

/*566*/ n_item(n_item(ie)) -->[ie].

/*567*/ n_item(n_item(tabak)) -->[tabak].

/*568*/ n_locale(n_locale(ev)) --> [ev].
```

```
/*569*/ n_locale(n_locale(da)) -->[da].

/*570*/ n_locale(n_locale(ke)) -->[ke].

/*571*/ n_locale(n_locale(taraf)) -->[taraf].

/*572*/ n_locale(n_locale(ufuk)) -->[ufuk].

/*573*/ n_locale(n_locale(yer)) -->[yer].

/*574*/ n_locale_proper(n_locale_proper(stanbul)) -->[stanbul].

/*575*/ n_locale_proper(n_locale_proper(mersin)) -->[mersin].

/*576*/ n_locale_proper(n_locale_proper(samsun)) -->[samsun].

/*577*/ n_number(n_number(bir)) -->[bir].

/*578*/ n_number(n_number(iki)) -->[iki].

/*579*/ n_number(n_number()) -->[].

/*580*/ n_number(n_number(drt)) -->[drt].

/*581*/ n_substance(n_substance(altn)) -->[altn].

/*582*/ n_substance(n_substance(kat)) -->[kat].

/*583*/ n_substance(n_substance(tahta)) -->[tahta].

/*584*/ n_substance(n_substance(ta)) -->[ta].

/*585*/ n_time_a(n_time_a(ay)) -->[ay].

/*586*/ n_time_a(n_time_a(gn)) -->[gn].

/*587*/ n_time_a(n_time_a(hafta)) -->[hafta].

/*588*/ n_time_a(n_time_a(sabah)) -->[sabah].

/*589*/ n_time_a(n_time_a(yl)) -->[yl].

/*590*/ n_time_b(n_time_b(bir)) -->[bir].
```

```
/*591*/ n_time_b(n_time_b(iki)) -->[iki].

/*592*/ n_time_b(n_time_b()) -->[].

/*593*/ n_time_b(n_time_b(drt)) -->[drt].

/*594*/ n_time_b(n_time_b(be)) -->[be].

/*595*/ n_time_b(n_time_b(alt)) -->[alt].

/*596*/ n_time_b(n_time_b(yedi)) -->[yedi].

/*597*/ n_time_b(n_time_b(sekiz)) -->[sekiz].

/*598*/ n_time_b(n_time_b(dokuz)) -->[dokuz].

/*599*/ n_time_b(n_time_b(on)) -->[on].

/*600*/ n_time_b(n_time_b(onbir)) -->[onbir].

/*601*/ n_time_b(n_time_b(oniki)) -->[oniki].

/*602*/ n_time_b(n_time_b(on)) -->[on].

/*603*/ n_time_b(n_time_b(ondrt)) -->[ondrt].

/*604*/ n_time_b(n_time_b(onbe)) -->[onbe].

/*605*/ n_time_b(n_time_b(onalt)) -->[onalt].

/*606*/ n_time_b(n_time_b(onyedi)) -->[onyedi].

/*607*/ n_time_b(n_time_b(onsekiz)) -->[onsekiz].

/*608*/ n_time_b(n_time_b(ondokuz)) -->[ondokuz].

/*609*/ n_time_b(n_time_b(yirmi)) -->[yirmi].

/*610*/ n_time_b(n_time_b(yirmibir)) -->[yirmibir].

/*611*/ n_time_b(n_time_b(yirmiiki)) -->[yirmiiki].

/*612*/ n_time_b(n_time_b(yirmi)) -->[yirmi].
```

```
/*613*/ n_time_b(n_time_b(yirmidrt)) -->[yirmidrt].

/*614*/ particle_a(particle_a(artk)) -->[artk].

/*615*/ particle_a(particle_a(bari)) -->[bari].

/*616*/ particle_a(particle_a(gene)) -->[gene].

/*617*/ particle_a(particle_a(hemen)) -->[hemen].

/*618*/ particle_a(particle_a(henz)) -->[henz].

/*619*/ particle_b(particle_b(ama)) -->[ama].

/*620*/ particle_b(particle_b(nk)) -->[nk].

/*621*/ particle_b(particle_b(evet)) -->[evet].

/*622*/ particle_b(particle_b(fakat)) -->[fakat].

/*623*/ particle_b(particle_b(hayr)) -->[hayr].

/*624*/ postposition_a(postposition_a(baka)) -->[baka].

/*625*/ postposition_a(postposition_a(nce)) -->[nce].

/*626*/ postposition_a(postposition_a(sonra)) -->[sonra].

/*627*/ postposition_b(postposition_b(gayri)) -->[gayri].

/*628*/ postposition_b(postposition_b(itibaren)) -->[itibaren].

/*629*/ postposition_b(postposition_b(maada)) -->[maada].

/*630*/ postposition_b(postposition_b(te)) -->[te].

/*631*/ postposition_c(postposition_c(beri)) -->[beri].

/*632*/ postposition_c(postposition_c(dolay)) -->[dolay].

/*633*/ postposition_c(postposition_c(nai)) -->[nai].

/*634*/ postposition_c(postposition_c(tr)) -->[tr].
```

```
/*635*/ postposition_d(postposition_d(nispeten)) -->[nispeten].

/*636*/ postposition_e(postposition_e(ait)) -->[ait].

/*637*/ postposition_e(postposition_e(dair)) -->[dair].

/*638*/ postposition_e(postposition_e(gre)) -->[gre].

/*639*/ postposition_e(postposition_e(kadar)) -->[kadar].

/*640*/ postposition_e(postposition_e(ramen)) -->[ramen].

/*641*/ postposition_f(postposition_f(gibi)) -->[gibi].

/*642*/ postposition_f(postposition_f(iin)) -->[iin].

/*643*/ postposition_f(postposition_f(kadar)) -->[kadar].

/*644*/ pronoun(pronoun(ben)) -->[ben].

/*645*/ pronoun(pronoun(sen)) -->[sen].

/*646*/ pronoun(pronoun(o)) -->[o].

/*647*/ pronoun(pronoun(bu)) -->[bu].

/*648*/ pronoun(pronoun(u)) -->[u].

/*649*/ verb(verb(bk)) -->[bk].

/*650*/ verb(verb(l)) -->[l].

/*651*/ verb(verb(kork)) -->[kork].

/*652*/ verb_impersonal(verb_impersonal(ak)) -->[ak].

/*653*/ verb_impersonal(verb_impersonal(birik)) -->[birik].

/*654*/ verb_impersonal(verb_impersonal(dol)) -->[dol].

/*655*/ verb_impersonal(verb_impersonal(es)) -->[es].

/*656*/ verb_impersonal(verb_impersonal(kayna)) -->[kayna].
```

```
/*657*/ verb_impersonal(verb_impersonal(pi)) -->[pi].

/*658*/ verb_mobile(verb_mobile(bak)) -->[bak].

/*659*/ verb_mobile(verb_mobile(k)) -->[k].

/*660*/ verb_mobile(verb_mobile(dn)) -->[dn].

/*661*/ verb_mobile(verb_mobile(d)) -->[d].

/*662*/ verb_mobile(verb_mobile(ka)) -->[ka].

/*663*/ verb_mobile_a(verb_mobile_a(git)) -->[git].

/*664*/ verb_mobile_b(verb_mobile_b(gel)) -->[gel].

/*665*/ verb_transitive(verb_transitive(bekle)) -->[bekle].

/*666*/ verb_transitive(verb_transitive(duy)) -->[duy].

/*667*/ verb_transitive(verb_transitive(gr)) -->[gr].

/*668*/ verb_transitive(verb_transitive(ye)) -->[ye].

/*669*/ verb_transitive_human(verb_transitive_human(anla)) -->[anla].

/*670*/ verb_transitive_human(verb_transitive_human(bil)) -->[bil].

/*671*/ verb_transitive_human(verb_transitive_human(dn)) -->[dn].

/*672*/ verb_transitive_human_mobile_1(verb_transitive_human_mobile_1(konu
           )) -->[konu].

/*673*/ verb_transitive_human_mobile_1(verb_transitive_human_mobile_1(syle
           )) -->[syle].

/*674*/ verb_transitive_human_mobile_2(verb_transitive_human_mobile_2(oku))
            -->[oku].

/*675*/ verb_transitive_human_mobile_2(verb_transitive_human_mobile_2(yaz))
            -->[yaz].

/*676*/ verb_transitive_mobile_1(verb_transitive_mobile_1(ara)) -->[ara].
```

```
/*677*/ verb_transitive_mobile_1(verb_transitive_mobile_1(bul)) -->[bul].

/*678*/ verb_transitive_mobile_1(verb_transitive_mobile_1(ek)) -->[ek].

/*679*/ verb_transitive_mobile_1(verb_transitive_mobile_1(ar)) -->[ar
           ].

/*680*/ verb_transitive_mobile_1(verb_transitive_mobile_1(kr)) -->[kr].

/*681*/ verb_transitive_mobile_1(verb_transitive_mobile_1(sar)) -->[sar].

/*682*/ verb_transitive_mobile_2(verb_transitive_mobile_2(a)) -->[a].

/*683*/ verb_transitive_mobile_2(verb_transitive_mobile_2(al)) -->[al].

/*684*/ verb_transitive_mobile_2(verb_transitive_mobile_2(i)) -->[i].

/*685*/ verb_transitive_mobile_2(verb_transitive_mobile_2(oyna)) -->[oyna].


/*686*/ verb_transitive_mobile_2(verb_transitive_mobile_2(yap)) -->[yap].

/*687*/ verb_transitive_mobile_2(verb_transitive_mobile_2(yap)) -->[yap].
```

References

[1] Hasan Mesut Meral; Emre Sevinç; Ersin Únkar; Búlent Sankur; Sumru Özsoy; Tunga Güngór. Syntactic tools for text watermarking. In *Electronic Imaging: Security, Steganography, and Watermarking of Multimedia Contents Conference*, San Jose, CA, 2007.

[2] Umut Topkara;Mercan Topkara;Mikhail J. Atallah. Words are not enough: Sentence level natural language watermarking. *Proceedings of ACM Workshop on Content Protection and Security (in conjunction with ACM Multimedia)*, 2002.

[3] Hasan Mesut Meral;Búlent Sankur; A.Sumru Özsoy. Watermarking tools for turkish texts. In *Proceedings of National Conference on Signal Processing and Application*, Antalya, 2006. Sabanc University.

[4] Gúlşen Eryiǵit;Joakim Nivre;Kemal Oflazer. The incremental use of morphological information and lexicalization in data-driven dependency parsing. In *Proceedings of ICCPOL 2006, The 21st International Conference on Computer Processing of Oriental Languages*, Singapore, December 2006.

[5] K.Oflazer;B.Say;D.Z.Hakkani Tr;G.Tr. *In Building and Exploiting Syntactically-annotated Corpora*, chapter Building a Turkish Treebank. Kluwer Academic Publishers, 2003.

[6] Marie-Catherine de Marneffe; Bill MacCartney; Cristopher D.Mannig. Generating typed dependency parses from phrase structure parses. In *5th International Conference on Language Resources and Evaluation (LREC 2006)*, 2006.

[7] Dan Klein; Chris Manning. Fast exact natural language parsing with a factored model. *Advances in Neural Information Processing Systems*, 15, 2002.

[8] Robert H.Meskill. *A Transformational Analysis Of Turkish Syntax.* PhD thesis, Brown University, 1970.

[9] Gerald Gazdar;Chris Mellish. *Natural Language Processing In Prolog-An Introduction to Computational Linguistcs.* Addsion-Wasley Publishing Company.

[10] Nart B.Atalay; Kemal Oflazer; Bilge Say. The annotation process in the turkish treebank. In *Proceedings of the EACL Workshop on Linguistically Interpreted Corpora - LINC*, Budapest, Hungary, April 2003.