DEVELOPMENT OF ALGORITHMS AND COMPUTATIONAL GRAMMAR FOR URDU

Thesis · April 2007

DOI: 10.13140/RG.2.1.4129.3846

CITATIONS
6

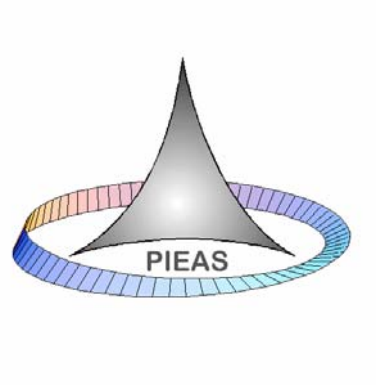
READS
5,361

1 author:

S.M.J. Rizvi
Pakistan Institute of Engineering and Applied Sciences
13 PUBLICATIONS **83** CITATIONS

SEE PROFILE

DEVELOPMENT OF ALGORITHMS AND COMPUTATIONAL GRAMMAR FOR URDU

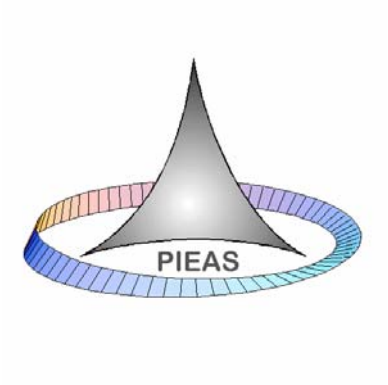


SYED MUHAMMAD JAFAR RIZVI

PAKISTAN INSTITUTE OF ENGINEERING AND APPLIED SCIENCES
NILORE ISLAMABAD 45650 PAKISTAN

March 2007

DEVELOPMENT OF ALGORITHMS AND COMPUTATIONAL GRAMMAR FOR URDU



SYED MUHAMMAD JAFAR RIZVI

THESIS SUBMITTED TO
DEPARTMENT OF COMPUTER AND INFORMATION SCIENCES
IN PARTIAL FULFILLMENT OF REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

PAKISTAN INSTITUTE OF ENGINEERING AND APPLIED SCIENCES
NILORE ISLAMABAD 45650 PAKISTAN

March 2007

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِيْم

شروع اللہ کے نام سے جو بڑا مہربان اور رحم کرنے والا ہے

In the name of Allah, the Most Beneficent, the Most Merciful

Au nom d'Allah, le Tout Miséricordieux, le Très Miséricordieux

Im Namen Allahs, des Gnädigen, des Barmherzigen

奉至仁至慈的真主之名

به نام خداوند بخشنده بخشایشگر

शुरू करता हूँ अल्लाह के नाम से जो रहमान और रहीम (दयालु और कृपालु) है

Ба номи Худованди бахшандаи мехрубон

CERTIFICATE

Certified that the work contained in this thesis is carried out by Mr. Syed Muhammad Jafar Rizvi under my supervision.

(Dr. Mutawarra Hussain)
Department of Computer & Information Sciences
PIEAS, P.O. Nilore
Islamabad, Pakistan.

Submitted Through

(Dr. Anila Usman)
Head
Department of Computer and Information Sciences
PIEAS, Post Office Nilore
Islamabad, Pakistan

DEDICATION

This document is dedicated to Professor Dr. Atta-ur-Rahman, TI, SI, HI, NI, Chairman, Higher Education Commission, Pakistan. The research and development (R&D) activity in our country was suffering due to paucity of funds for the higher education as well as due to lack of priority and interest. However, presently plenty of funds have been made available by Dr. Atta-ur-Rahman through HEC. He took numerous intitiatives including indigenous and foreign scholarship schemes, human rescource development, expatriate and foreign faculty hiring, quality assurance, etc. He sponsored a number of research proposals submitted by various Universities of Pakistan. The contribution of Dr. Atta-ur-Rahman will always be remembered for promoting research activities in Pakistan.

This study has been sponsored by HEC under the indigenous Ph.D. Scholarship Scheme 2001.

ACKNOWLEDGEMENTS

I thank my supervisor Dr. Mutawarra Hussain (DCS, PIEAS) for guiding me as a Ph.D. student. Whenever, he was busy in office hours, he gave me time after office hours. He helped me in gathering books, papers, software tools, etc. Being in the first batch of the indigenous Ph.D. program in Pakistan, I had to face difficulties and to solve difficulties he was always with me. At many times, I felt that I was not able to show good performance and to focus in a particular direction, he always encouraged me in all those difficult times.

I thank Dr. Miriam Butt (Professor, Universität Konstanz, Germany), who is helpful to me since the start of choosing the topic for Ph.D. work. She continuously gave feedback to my work through email contact despite her busy schedules. She sent me study material which otherwise were not available. Most of my work in this thesis is based on her papers, books, handouts, email conversations and her lectures at Lahore.

I thank Dr. Ron Kaplan (Xerox, USA), Dr. Tracy Holloway King (Xerox, USA) and Dr. Miriam Butt for helping and providing me linguistic software: Xerox Linguistic Environment (XLE), Xerox Finite State Tool (XFST) and Two level Rule Compiler (TWOLC). These software tools proved useful during my research work.

I thank Dr. Mohammad Abid Khan (Chairman, Department of Computer Science, University of Peshawar) for his guidance and study material. The study of his Ph.D. thesis gave me insight of the topic. Discussions with him on machine translation were useful in clarifying my point of view.

I thank Dr. Sarmad Hussain (Head CRULP, FAST NUCES, Lahore) for providing me study material for readings and inviting me for the various study sessions held at the CRULP, Lahore. His guidance helped me narrow down my direction.

I thank Dr. Khalid Ibrahim and Dr. Saeed Ahmad Durrani, who took time from their busy schedules for thoroughly reading my thesis, they gave valuable suggestions. I thank Dr. Sikander Majid Mirza (DCS, PIEAS) for chatting about my topic and for nice advises. I thank Dr. Abdul Jalil, Dr. Anila Usman, Dr. Muhammad Arif as well as all other faculty/ staff personnel at DCIS and other departments of PIEAS. I thank all my colleague Ph.D. scholars at PIEAS, who encouraged me during Ph.D. studies. I thank my employer for the grant of study leave for the Ph.D. Studies. I thank all my family members, relatives and friends for their well wishes.

ABSTRACT

This work presents the linguistics-based grammar modeling of Urdu language under the framework of Lexical Functional Grammar (LFG) and at places under Head-driven Phrase Structure Grammar (HPSG). The grammar modeling has been done by considering two interlinked parts: the morphology and the syntax.

Urdu has a rich verb morphology comprising 60 basic verb forms categorized into infinitive, perfective, repetitive, subjunctive and imperative forms. The 60 forms are not enough to represent all the features of Urdu verbs. Various verb features are composed when verb auxiliaries and/or light verbs combine with these verb forms. Linguistically, verb auxiliaries are needed to combine at the syntactic level. However, this work shows that the grammar model is simplified and the complex agreement requirements can be avoided if auxiliaries are lumped with verb forms at the lexical level. The work proposes the analysis of perfective, progressive, repetitive and inceptive aspects as well as the analysis of declarative, permissive, prohibitive, imperative, capacitive, suggestive, compulsive, presumptive and subjunctive moods. The structure of a passive is analyzed by assuming a default argument.

This work, based on difference in grammar modeling and conceptualization, classifies Urdu case markers and post-positions into noun forms, core case markers, functional case markers, possession markers and post-positions. Noun forms are modeled morphologically using lexical transducers, possession markers require two noun phrases, post-position appear as adjuncts, while core and functional case markers appear in the argument structure of verbs.

To classify core and functional case markers the use of semantic features has been proposed. The semantic features based classification particularly demonstrated better taxonomy of different 'instrumental cases' in Urdu. This classification of 'instrumental case' exposed the presence of 'indirect subjects' for Urdu causative verbs which further suggested that some causative verbs are tetravalent because the argument structure of these verbs has four arguments.

The study of case-markers reveals that the agreement between a noun and a case marker is difficult to handle. It is argued that the head of phrase should be a noun because the resultant is a noun phrase, but features of the case marker also transfer to the resultant phrase, therefore, a modification to head-feature rule is proposed. The same argument also helped to reaffirm that Urdu case markers are different from Urdu possession markers, which require a different rule needing two noun phrases as a

specifier and a complement to make a resultant noun phrase. The adjective-noun agreement is also modeled on the same grounds for their gender and number agreement.

The work proposes an algorithm for the parsing Urdu sentences based on Urdu closed-word-classes. This helps in identifying chunks based on the linguistic characteristics of the word classes. The rule selection is simplified by providing a guess of the word class that may appear before or after it.

The work also presents a novel roman script for Urdu language for transliteration, which is not only phonetic like other roman scripts, but also makes possible to transfer text in this roman script to or from Urdu script, in both directions, using a computer program.

This thesis, therefore, presents novel ideas for the computational grammar of Urdu, which can be utilized in various natural language processing tasks, such as machine translation, text summarization, grammar checker, information retrieval, etc.

TABLE OF CONTENTS

PART II: MORPHOLOGICAL ANALYSIS AND LEXICAL ATTRIBUTES

PART III: SYNTACTICAL ANALYSIS AND MODELING

LIST OF TABLES

LIST OF FIGURES

SYMBOLS AND ABBREVIATIONS

Attributes:

action	ACTION
adjunct	ADJ
aspect	ASPECT
base language	BASELANG
case	CASE
conjunction	CONJ
gender	GEND
noun class	N-CLASS
noun concept	N-CONCEPT
noun form	N-FORM
noun type	N-TYPE
number	NUM
object	OBJ
person	PERS
predicate	PRED
prepositional case	PCASE
semantics	SEM
specifier	SPEC
subject	SUBJ
tense	TENSE
verb form	V-FORM
verb mood	V-MOOD
verb voice	V-VOICE

Values:

accusative	*acc*
dative	*dat*
ergative	*erg*
feminine	*fem*
first person	*1st*
genitive	*gen*
locative	*loc*
masculine	*masc*
no, not present	*–*
nominative	*nom*
oblique	*obl*
plural	*pl*
second person	*2nd*
singular	*sg*
third person	*3rd*
yes, present	*+*
vocative	*voc*

Part of Speech Tags:

adjective	Adj
adverb	Adv
case marker	CM
possession marker	PM
noun	N
postposition, preposition	PP
verb	V
verb base	VB
verb morpheme	VM
conjunction coordinating	CJC
conjunction subordinating	CJS
conjunction correlative	CJR
Interjection	IJ
pronoun subjective	PNS
pronoun objective	PNO
pronoun possessive	PNP
pronoun reflexive	PNR
pronoun indefinite	PNI
negation marker	NM
question marker	QM
focus marker	FM
topic marker	TM
titles	TLE
numbers ordinal	NO
numvers cardinal	NC
auxiliary	AUX
auxiliary perfective aspect	APA
auxiliary progressive aspect	APrA
auxiliary repetitive aspect	ARA
auxiliary inceptive aspect	AIA
auxiliary compulsive mood	ACoM
auxiliary capacitive mood	ACaM
auxiliary suggestive mood	ASM
auxiliary declarative mood	ADM
auxiliary permissive mood	APeM
auxiliary prohibitive mood	APrM
sentence	S
postpositional phrase	PPP
noun phrase	NP
verb phrase	VP
adjunct phrase	AJP

PART I

INTRODUCTION AND REVIEW

Chapter 1

RESEARCH OBJECTIVES

1.1 Objectives Statement

The objective of the Ph.D. work carried out was to develop a computational grammar by investigating the formation of Urdu words and sentences; to find some suitable mathematical formalism that can handle various constructions of Urdu grammar in a universal manner; to determine formulations of grammar rules under the selected framework; to investigate and develop associated algorithms.

While developing the computational grammar, the main application under vision was Machine Translation (MT) between English and Urdu languages. However, the computational grammar thus developed may be utilized for various other Natural Language Processing (NLP) applications. Some of the applications among many others are: grammar checker, machine translation, text summarization, text categorization, information extraction, speech processing and knowledge engineering.

1.2 Domain of Investigation

Mainly the linguistics-based and statistical-based approaches are used for the development of computational grammar. However, in this study, the linguistics-based grammar theories have been investigated. The linguistics based Natural Language Processing (NLP) employs human knowledge of word and sentence structures to formulate rules or equations for representing acceptable structures. The statistical NLP, on the other hand, employs statistical pattern matching and other training algorithms on the given data to learn the structure of the language.

The study investigates Urdu sentences composed of individual basic characters in the text format as opposed to the sentence as a single image, thus this study is not related to image processing or optical character recognition. The study is divided into two parts: the study of the structure of word formation, i.e., morphology, and the study of the structure of sentence formation, i.e., syntax.

The word classes investigated under morphology are verbs, nouns and adjectives. Xerox finite state lexicon compiler 'LEXC' and Xerox finite state tool 'XFST' are used for morphological analysis of Urdu words. The lexicon compiler

'LEXC' has its own language for entering the lexical data and morphological information, and it builds a finite state network usually referred to as a 'lexical transducer'. The lexical transducer 'looks-up' surface morphological form of a word into a lexicon and finds lexical form of a word and 'looks-down' lexical word and gives corresponding morphological form.

For modeling the Urdu syntax, sentences from frequently used constructions in Urdu are investigated. Lexical Functional Grammar (LFG) is used for the mathematical formulation of Urdu grammar. At places, the formulation is carried out under Head-driven Phrase Structure Grammar (HPSG). Both of these grammar-modeling theories are linguistic based extensions to Context Free Grammar (CFG). Although both are different in details, yet both have evolved from a single base and both have attributes and values associated with lexical entries. The well-known CFG parsing algorithms work with linguistics based constraints and rules to achieve linguistic criteria. Xerox Linguistic Environment 'XLE' is used for the testing and validation of Urdu grammar formulation using LFG, which has interface with morphological tool 'LEXC' and has parsing and unification algorithms required for LFG.

The hash-table and deterministic finite-state automata (DFA) minimization algorithms for the implementation of Urdu lexicon were explored and programmed. Work on shallow parsing algorithms that utilize closed word classes in Urdu was also carried out using a novel 'ordered context free grammar', which has additional attributes 'order' and 'type' associated with each CFG production rule. The algorithm has been implemented such that it utilizes the advantages of object oriented paradigm.

1.3 Organization of Thesis

The thesis is organized in three main parts. The Part I (Chapter 1–3), comprises introduction, review and preliminary information on grammar modeling that forms a context for further discussion in the next chapters. In Part II (Chapter 4–6), the work on Urdu morphology is presented. The characteristics and morphology of verbs, nouns and adjectives in Urdu are investigated. The features necessary to model lexical categories are identified. The algorithms for computational lexicon representation were reviewed and implemented. In Part III (Chapter 7–9), the work on Urdu syntax is presented. The modeling of nominal and verbal structure is carried out under the framework of LFG by proposing novel ideas. A chunking based parsing algorithm for Urdu language is proposed that utilizes ordered context free grammar.

In Chapter 1, an objective statement is given, the domain of investigation for the work carried out is defined and the organization of the thesis is described.

In Chapter 2, an introduction to the field of machine translation is given. The ambiguities involved at various stages in machine translation have been described

with reference to English and Urdu languages. The data is presented to show that Urdu has two more reasons for lexical ambiguities in addition to two sources of lexical ambiguities in English language. Some examples are presented to show that 'attachment of prepositional phrase', which is the basic reason of syntactic ambiguity in English, is rarely a cause of ambiguity in Urdu. However, the Urdu language has some other sources for syntactic ambiguities such as 'attachment of a participle adjunct', 'modifier scope with the noun phrase' and 'conjunction scope'. Various machine translation paradigms have been briefly reviewed. Linguistics-based approaches typically employ manual investigation of language features in comparison with non-linguistic approaches, which employ computational methods to extract features automatically.

In Chapter 3, a brief review of grammar modeling is presented. Among context free phrase structure grammar modeling and linguistics based grammar modeling it is found that linguistics based grammar modeling is a better solution. A brief review of popular grammar modeling theories like Lexical Functional Grammar (LFG) formalism is presented with examples from Urdu language to determine suitability of the framework for the modeling of the Urdu language grammar. Head driven Phrase Structure Grammar (HPSG) is another popular theory for the grammar modeling of natural languages, the newer version of which appeared in 2004. The chapter presents some basic features of HPSG theory and explores its usage to model the noun-case agreement, the noun-adjective agreement and the possession marking for the Urdu language. The HPSG has the advantage of having object-oriented hierarchical inheritance based architecture. However, it will be explored in forthcoming chapters that the grammar modeling using LFG is more language-neutral than by using HPSG. Moreover, LFG covers linguistic variations across world languages in a more natural manner.

In Chapter 4, Urdu verb morphology and characteristics have been investigated. Urdu, like some other languages, has intransitive, transitive and ditransitive verbs. Urdu has three stem forms named as the root form, the causative form 1 and the causative form 2. Each of these three stem forms are further divided into 20 verb forms under five categories, i.e., infinitive, perfective, repetitive, subjunctive and imperative verb forms. Hence, three stem forms, further divided into 20 forms, make 60 verb forms of a single Urdu verb. A finite-state-automaton is presented to represent these 60 forms. The tags necessary to distinguish person, gender, number, respect, tense, aspect, and mode, are also tabulated.

In Chapter 5, Urdu noun morphology and characteristics are investigated. A noun in Urdu has gender attribute for all nouns, but very few nouns in Urdu have overt gender morpheme. The nouns have nominative form if they appear without a

case-marker or post-position, have oblique form if they appear with a case-marker or post-position and have vocative form in subjunctive mood. Again, not all nouns have visible morpheme to distinguish nominative, oblique and vocative forms. The adjectives also have 'gender', 'number' and 'form' morphemes, which require agreement with the noun. The tags required to distinguish various noun categories or characteristics are looked into and listed.

In Chapter 6, the review and implementation of algorithms for constructing a computational lexicon has been carried out. Some hash functions were implemented for constructing a lexicon without morphological considerations. Similarly, some deterministic-finite-state-automaton minimization algorithms were implemented to construct lexicon using 'lexical transducers'. A comparison between the two approaches is made to check which method requires more time and space and how much morphological analysis is needed for each implementation.

In Chapter 7, the modeling of the nominal syntax in Urdu is carried out. In Urdu, nouns accompany various case-markers and post-positions to form phrases that fill various grammatical roles in the argument structure of a verb. In this Chapter, the classification of case-markers and post-position is described. The classification is based on the difference in modeling and conceptualization, such as on the basis that whether they are handled morphologically or syntactically, whether they are controlled by verb's argument-structure or not, whether they are attached to a core function or an oblique function. To resolve some of the ambiguities involved semantic class of nouns, such as animate, instrumental, location is employed. The case marker '*sey*' appears in different roles with different nouns. To distinguish these roles the noun's semantic class has been found useful. Possession markers are different from the case-markers, because they require two noun phrases – the possessor and the possessee. Moreover, possession markers require agreement in 'gender' and 'number' and these are not controlled by the argument-structure of a verb. It is also proposed, in this Chapter, that the argument-structure of some causative form 2 verbs may have four noun-phrases – an agent marked with '*ney*', an intermediate agent marked with '*sey*', an indirect object phrase marked with '*kao*' and a nominative object. This analysis assumes that the intermediate agent, like an agent in a passive sentence, is sometimes omitted, which is semantically implied.

In Chapter 8, the modeling of the verbal syntax in Urdu is carried out. The main features represented by a verb are tense, aspect and mood. The verb agreement in Urdu has many dimensions for the dependency, due to which verbs and auxiliary verbs change their form. The tense, aspect and mood features represented by various verb morphemes and auxiliaries are identified and phrase structure rules for the formation of sentences are presented. It is proposed that computationally a verb in

Urdu may be separated into two lexical parts: (i) the root or stem of a verb, which carries the principal meaning of a verb and contains information about the transitivity and argument-structure; (ii) the inflectional morphemes and auxiliary verbs, which carry information about tense, mood and aspect. The computational equations are simpler using this approach, however, other approach of combining verbs and auxiliaries at syntactic level has other advantages. The perfect, progressive, repetitive and inceptive aspects in Urdu are modeled under LFG. The declarative, permissive, prohibitive, imperative, capacitive and suggestive moods in Urdu are modeled under LFG by presenting c-structures and f-structures.

In Chapter 9, the parsing by chunking is explored based on morphologically closed word classes in Urdu and using a novel Ordered Context Free Grammar (OCFG). The proposed OCFG rules have additional attributes, i.e., order and type associated with each rule. The order of a rule employs linguistic features of words to make chunks with neighbor words, e.g., the case-marker make chunks with nouns to make noun phrases. The final parse is achieved after chunks of basic phrases have been made. While chunking and parsing drive parse tree (i.e., c-structure), the features unification may be carried out simultaneously to improve the proposed method.

In Chapter 10, the summary and conclusions of the work done in this thesis are described. The applications of the work done and future directions are discussed.

In Appendix A, a roman-script is proposed, which is used for the transcription of Urdu sentences in this thesis. The characters of this roman-script are selected in such a way that computerized transfer of text to this roman-script from Urdu-script is possible and vice versa. It is also taken care that the mapped characters in these scripts be phonetically the same or as close as possible. In Appendix B, algorithms for lexicon representation used for lexicon implementation comparison in Chapter 6 have been given. In Appendix C, sample sentences for chunking based parsing described in Chapter 9 have been listed. In Appendix D, constituent-structures corresponding to feature-structures given in Chapter 7 have been included. In Appendix E, Urdu grammar implementation in the coding format of Xerox tools have been listed. The morphology implementation code is in the format of LEXC. The morphology-syntax interface code is used by XLE for porting the morphology information to syntax. The listed syntax rules have been coded in XLE format, which generate c-structures and f-structures for the Urdu sentences.

Chapter 2

INTRODUCTION TO MACHINE TRANSLATION

Natural languages are used by humans for communication among themselves in contrast to programming languages, which are used for the communication between humans and machines. Natural Language Processing (NLP) is the field that deals with the computer processing of natural languages, mainly evolved by people working in the field of Artificial Intelligence (AI). Computational Linguistics (CL) deals with the computational aspects of natural languages and this discipline is primarily evolved by linguists. Currently there are many branches of NLP like Machine Translation (MT), speech processing, information retrieval, text summarization, etc. Although the computational grammar developed in this work can be utilized for various NLP applications, yet machine translation is the main application targeted while developing the grammar.

2.1 Machine Translation (MT)

Machine Translation is the transfer of text from one natural language, known as *source language*, to another natural language, known as *target language*, by means of a computer program or a machine (Arnold, Balkan et al. 1994; Khan 1995; Hutchins and Somers 1997; Trujillo 1999).

2.2 Challenges for Machine Translation

Machine Translation is a challenging problem. The challenge for machine translation is to develop a grammar formulation for handling different kinds of ambiguities that are present in a source and a target language. These ambiguities sometimes arise due to the inability of robust formulation of grammar under any modeling theory and sometimes these are naturally present in the sentences and require knowledge of semantics and pragmatics for their resolution. Natural languages are multifaceted, if one language is expressing some concept using one way, other language uses another way of representing the same concept. Modeling of a natural language under any linguistics based grammatical theory is still a challenge. Multiword units like idioms and collocations found in languages are difficult to handle (Arnold, Balkan et al. 1994; Hutchins and Somers 1997). Anaphora and cataphora resolution in discourse is a complex problem (Khan 1995). Review of some

basic ambiguity related problems is described below along with examples from English and Urdu languages.

2.2.1 Lexical Ambiguity

Ideally, each word in a language should have a unique meaning, but for natural languages, many words have two or more interpretations. When a sentence becomes ambiguous due to a word then this type of ambiguity is called lexical ambiguity. The lexical ambiguity may arise due to two main reasons: (i) one word belongs to two or more lexical categories (ii) one word has more than one interpretation.

The lexical ambiguity, in which a word belongs to more than one lexical category, causes the word to have a different meaning due to the difference of category. The different meanings of the same word make the word ambiguous. These words are multinational in the lexicon's world. In such a case of lexical ambiguity, performing the syntactical analysis normally resolves the ambiguity. Table 2.1 shows a few examples of such English words and Table 2.2 shows a few examples of Urdu words.

Table 2.1: Some Lexically Ambiguous English Words

fly	noun	an insect	fly	verb	I want to fly
use	noun	the use of a knife	use	verb	do not use a knife
can	noun	a can of juice	can	auxiliary	I can write now
novel	noun	book, story	novel	adjective	new, original
today	noun	today is eid	today	adverb	we'll go today

Table 2.2: Some Lexically Ambiguous Urdu Words

خطا	*xattaa*, mistake	noun	خطا	*xattaa*, to miss	verb
سونا	*saonaa*, gold	noun	سونا	*saonaa*, to sleep	verb
گلا	*galaa*, throat	noun	گلا	*galaa*, a softened/ cooked state	adjective
اتفاق	*aetefaaq*, unity	noun	اتفاق	*aetefaaq*, by coincidence	adverb
گانا	*gaanaa*, song	noun	گانا	*gaanaa*, to sing	verb
کھانا	*khaanaa*, food	noun	کھانا	*khaanaa*, to eat	verb

The lexical ambiguity in which a word has different meanings within the same lexical category is pure lexical in nature and this ambiguity cannot be resolved by the syntactic analysis. This property of words is often termed as *polysemy*. Semantic and contextual knowledge of the word usage is required for the ambiguity resolution. Table 2.3 lists some polysemic English words, while Table 2.4 shows some polysemic Urdu words.

Table 2.3: Some Polysemic English Words

bank	a financial institution	bank	a side of a river
table	a tabulated information	table	a wooden furniture
film	a movie, a picture	film	a layer, a coating
cricket	a game	cricket	an insect
mouse	a tiny animal	mouse	a computer instrument
ground	earth, soil, land	ground	reason, base

Table 2.4: Some Polysemic Urdu Words

ضد	*Jed*, opposite	fem. noun	ضد	*Jed*, stubbornness	fem. noun
صحت	*Sehat*, correctness	fem. noun	صحت	*Sehat*, health	fem. noun
تاریخ	*taareex*, date	fem. noun	تاریخ	*taareex*, history	fem. noun
کل	*kal*, tomorrow	masc. noun	کل	*kal*, yesterday	masc. noun
ہار	*haar*, necklace	masc. noun	ہار	*haar*, defeat	fem. noun
کان	*kaan*, ear	masc. noun	کان	*kaan*, mine, excavation	fem. noun
عرض	*AarJ*, width	masc. noun	عرض	*AarJ*, request	fem. noun

In addition to the above-mentioned types of lexical ambiguities in English, Urdu language, due to the nature of its script, has two more types of the lexical ambiguities. First type of lexical ambiguity normally arises due to the absence of diacritical marks in the written Urdu script. The diacritical marks represent vowel sounds and stops/pauses in Urdu. In written Urdu, these are omitted commonly and a reader of Urdu language uses the contextual knowledge to find the actual pronunciation of the given word in a sentence. The computational resolution of this kind of lexical ambiguity is a complex problem and beyond the scope of the syntactical analysis.

Table 2.5: Lexical Ambiguity in Urdu due to Absence of Diacritical Marks in Written Urdu

Written	Actual		
بل	بِل	*bel*, hole of insects	Noun
بکری	بِکرِی	*bekree*, sale	Noun
ان	اِن	*aen*, these	Pronoun
اس	اِس	*aes*, this	Pronoun
جلدی	جِلدِی	*jeldee*, of skin	Adjective
عالم	عالَم	*Aaalam*, world	Noun

Written	Actual	
بل	بَل	*bal*, power, strength
بکری	بَکرِی	*bakree*, goat
ان	اُن	*aon*, those
اس	اُس	*aos*, that
جلدی	جَلدِی	*jaldee*, quickly
عالم	عالِم	*Aaalem*, educated

There is another lexical ambiguity in Urdu due to two '*yey*' vowel shapes in Urdu, namely the big *yey*, ے, and the small *yey*, ی. When these '*yey*' appear as middle shape in a word, then both of these assume a single shape having two dots '*noqtah*' below. The ambiguity of two vowels sounds permit two different words to be written the same. To illustrate this ambiguity, some examples of such ambiguous words are shown in Table 2.6. These different words have different meaning but as a written word, these are the same. The same shape of these vowels represents a

consonant instead of a vowel, when it appears as a first character of a word or a phoneme. The sound of the consonant is represented by letter 'j' in the IPA table.

Table 2.6: Lexical Ambiguity in Urdu due to the same Middle Shape of two different Vowels

Noun	شیر	شِـیـر	*sheyr*, a lion	Noun	شیر	شِی ر	*sheer*, milk
Ques.	کیا	کِـیـا	*keyaa*, what	Verb	کیا	کِی ا	*keeaa*, did
Noun	بین	بَـیـن	*bayn*, whine	Noun	بین	بِی ن	*been*, musical instrument
Noun	چین	چَـیـن	*chayn*, calmness	Noun	چین	چِی ن	*cheen*, China
Noun	میرا	مِـیـرا	*meyraa*, mine	Noun	میرا	مِی را	*meeraa*, a name
Noun	خیر	خَـیـر	*xayr*, all right, fine	Noun	کھیر	کھِی ر	*kheer*, a sweet desert
Noun	فیس	فِـیـس	*feys*, face	Noun	فیس	فِی س	*fees*, fee
Noun	بیس	بِـیـس	*beys*, base	Adj.	بیس	بِی س	*bees*, twenty
Verb	بیک	بِـیـک	*beyk*, bake	Noun	بیک	بَـیـک	*bayk*, back

2.2.2 Syntactic or Structural Ambiguity

A sentence has syntactic or structural ambiguity if two or more structural interpretations can be assigned to it. If we consider translation from English to Urdu, then *attachment of prepositional phrases* with different syntactic units is one of the major reasons of syntactic ambiguity in English. The prepositional phrase can be attached to a noun to elaborate the noun phrase or with the main verb as an adjunct as shown in the following example sentences:

(1) I saw an astronomer *with a telescope*.

The English sentence shown in (1) has a prepositional phrase 'with a telescope' which may be attached either to the verb 'saw' to make phrase 'saw *something* with a telescope' or to the object noun phrase 'an astronomer' to make a noun phrase 'an astronomer with a telescope'. Due to attachment with different syntactic units, it results in the following two interpretations:

(1-a) میں نے خلاباز کو دیکھا جس کے پاس دوربین تھی
mayN ney xalaabaaz kao deykhaa jes key paas daorbeen thee❖1
I [[saw]$_V$ [[an astronomer]$_{NP}$ [with a telescope]$_{PP}$]$_{NP}$]$_{VP}$
I saw an astronomer, who is having a telescope; or

(1-b) میں نے خلاباز کو دوربین سے دیکھا
mayN ney xalaabaaz kao daorbeen sey deykhaa
I [[saw]$_V$ [an astronomer]$_{NP}$ [with a telescope]$_{PP}$]$_{VP}$
Using a telescope, I saw an astronomer.

❖1 The romanization / transcription system used throughout in this thesis for Urdu script is described in Appendix A.

(2) A teacher hit a student with an umbrella

Similarly, for the English sentence shown in (2), we may have the following two interpretations as shown in (2-a) and (2-b). In (2-a) 'student with an umbrella' is taken as a noun phrase, while in (2-b) 'with an umbrella' is attached to the verb 'hit' as an adjunct, which made the umbrella an instrument for hitting the student.

(2-a) استاد نے طالب علم کو، جس کے پاس چھتری تھی، مارا
aostaad ney ttaaleb Aelam kao, jes key paas chatree thee, maaraa
A teacher hit a student who had an umbrella;

or,

(2-b) استاد نے طالب علم کو چھتری سے مارا
aostaad ney ttaaleb Aelam kao chatree sey maaraa
A teacher hit a student by the use of an umbrella.

(3) Waseem cancelled a trip to Karachi to play cpicket

The sentence in (3) has two interpretations depending on the attachment of prepositional phrase 'to play cricket' with verb 'cancelled' or with noun 'trip'. If prepositional phrase is attached to 'cancelled' then we get the interpretation shown in (3-a) and the other interpretation is shown in (3-b).

(3-a) وسیم نے کراچی کا سفر کرکٹ کھیلنے کے لیے ملتوی کر دیا
waseem ney karaachee kaa safar karekeT kheylney key leeey moltawee kar deeaa
Waseem cancelled the trip to Karachi because he is to play cricket; or

(3-b) وسیم نے کراچی کا سفر، جو کرکٹ کھیلنے کے لیے تھا، ملتوی کر دیا
waseem ney karaachee kaa safar, jao karekeT kheylney key leeey thaa, moltawee kar deeaa
Waseem cancelled the trip which was for playing cricket at Karachi.

The ambiguity due to *attachment of complement structures* is shown in sentences (4) and (5).

(4) I forgot how good juice tastes.

(4-a) میں بھول گیا ہوں کہ اچھے جوس کا ذائقہ کیسا ہے
mayN bhool gayaa hooN keh ach.chey joos kaa Zaaeyqah kaysaa hay
I forgot [how [good juice] tastes]

(4-b) میں بھول گیا ہوں کہ کتنا اچھا جوس کا ذائقہ ہے
mayN bhool gayaa hooN keh ketnaa ach.chaa joos kaa Zaaeyqah hay
I forgot [[how good] juice tastes].

(5) Eating this often will make you fat.

(5-a) اسے اکثر کھانے سے تم موٹے ہو جاؤ گے
aesey aakthar khaaney sey tom maotey hao jaa-ao gey
[Eating this] [often] will make you fat

(5-b) اتنی دفعہ کھانے سے تم موٹے ہو جاؤ گے
aetnee dafAah khaaney sey tom maotey hao jaa-ao gey
[Eating] [this often] will make you fat.

The *ambiguity between gerund and participial adjective* results in different syntactic structures and therefore results in different interpretations, the examples of which are shown in (6) and (7) below:

(6) Visiting relatives can be boring.

(6-a) رشتہ داروں کے گھر جانے سے بوریت ہو سکتی ہے
reshtah daaraoN key ghar jaaney sey baoreeyat hao saktee hay
$[\text{Visiting}]_{V}$ $[\text{relatives}]_{N}$ [can be boring]; or

(6-b) گھر آنے والے رشتہ داروں سے بوریت ہو سکتی ہے
ghar aa.ney waaley reshtah daaraoN sey baoreeyat hao saktee hay
$[[\text{Visiting}]_{ADJ}$ relatives$]_{NP}$ [can be boring].

(7) Cleaning fluids can be dangerous.

(7-a) مائعات کو صاف کرنا نقصان دہ ہو سکتا ہے
maaAeyyat kao Saaf karnaa noqSaan deh hao saktaa hay
$[\text{Cleaning}]_{V}$ $[\text{fluids}]_{N}$ [can be dangerous]; or

(7-b) صفائی کے مائعات نقصان دہ ہو سکتے ہیں
Saafaaee key maaAeyyat noqSaan deh hao saktey hayN
$[[\text{Cleaning}]_{ADJ}$ fluids$]_{NP}$ [can be dangerous].

Modifier scope within noun phrase causes syntactic ambiguity as shown in phrases (8) and (9).

(8) impractical design requirements
غیر حقیقی ڈیزائن کی ضروریات ۔یا۔ ڈیزائن کی غیر حقیقی ضروریات
[impractical] [design requirements] -or- [impractical design] [requirements]

(9) plastic cup holder
پلاسٹک کپ کا ہولڈر ۔یا۔ پلاسٹک کا کپ ہولڈر
[plastic] [cup holder] -or- [plastic cup] [holder]

The syntactic ambiguity due to *conjunction scope* is shown in sentence (10):

(10) Small rats and mice can squeeze into holes or cracks in the wall.
چھوٹے چھچھوندر اور چوہے -یا- چھوٹے چھچھوندر اور چھوٹے چوہے
[Small [rats and mice]] can squeeze into holes or cracks in the wall.
[[Small rats] and [mice]] can squeeze into holes or cracks in the wall.

It is known that syntactic ambiguity exists in many sentences, which are not simple, and hence it makes parsing difficult. Different languages have different sources of syntactic ambiguity. Urdu does not have prepositions, instead it has postpositions. For most of the sentences, the position of post-positions in the Urdu sentence determines the syntactic unit to which it is to attach and thus enables to resolve the syntactic ambiguity. However, the syntactic problem may be seen in the following sentences of Urdu. Sentence (11) is ambiguous but sentence (12) is not, because position of post-positional phrase resolves the ambiguity.

(11) وسیم نے کرکٹ کھیلنے کے لیے کراچی کا سفر ملتوی کر دیا
waseem ney karekeT kheylney key leeey karaachee kaa safar moltawee kar deeaa
Waseem cancelled a trip to Karachi which was scheduled to play cricket.

(12) وسیم نے کراچی کا سفر کرکٹ کھیلنے کے لیے ملتوی کر دیا
waseem ney karaachee kaa safar karekeT kheylney key leeey moltawee kar deeaa
Waseem cancelled a trip to Karachi because he is to play cricket (say, at Lahore).

(13) وسیم نے اکرم کو عینک لگاتے ہوئے دیکھا
waseem ney aakram kao Aaynak lagaatey hooey deykhaa
Waseem while putting glasses on saw Akram, or
Waseem saw Akram who was wearing glasses.

2.2.3 Combined Lexical and Syntactic Ambiguity

The following sentence in English contains a word ‘dress’ that belongs to both ‘noun’ and ‘verb’ category and in addition to that it has two correct parse structures. Such a lexical ambiguity cannot be resolved by syntactic analysis alone.

(14) She [[made her [dress]$_N$] correctly]
She made her [[dress]$_V$ correctly]

Another source of ambiguity is distinction between a particle and a preposition as shown in example sentences (15) and (16). As a particle, resultant phrases are ‘dispenses with’ and ‘referred to’, while as a preposition resultant prepositional phrases are ‘with accuracy’ and ‘to student’s mistake’.

(15) A good pharmacist [dispenses with]$_V$ accuracy.
A good pharmacist dispenses [with accuracy]$_{PP}$.

(16) The teacher [referred to]$_V$ student's mistake.
The teacher referred [to student's mistake]$_{PP}$.

2.2.4 Semantic Ambiguity

When there is no syntactic or lexical ambiguity in a sentence yet the sentence has two different interpretations, then this kind of ambiguity is termed as semantic ambiguity. Semantic ambiguity also appears in sentences where the lexical ambiguity cannot be resolved by syntactic analysis and ambiguity resolution requires the knowledge of the semantic information for the resolution. The Urdu sentence in (17) has semantic ambiguity because either it can be interpreted as 'he ate the meal because the meal was ready' or as 'he ate the meal because he was hungry and ready for eating food'.

(17) اس نے کھانا کھا لیا کیونکہ وہ تیار تھا
aos ney khaanaa khaa leeaa keeoN-keh woh teyyaar thaa
He ate the meal because he/it was ready

If we compare sentences (18) and (19), then (18) has only one interpretation that we are ready for eating. The sentence (19) is similar to (18) both lexically and syntactically, but it could have two different interpretations. It may mean that we may start eating chickens, which are ready and cooked for us to eat. Second meaning of this sentence, like the meaning we get from sentence (18), is that chickens are ready to eat food and waiting, if we give them food the chickens will eat that food.

(18) ہم کھانے کے لیے تیار ہیں
ham khaaney key leeey teyyaar hayN
We are ready to eat

(19) مرغیاں کھانے کے لیے تیار ہیں
morgeeaN khaaney key leeey teyyaar hayN
The chickens are ready to eat food, *or*
The (cooked) chickens are ready (for someone) to eat

Sentence (20) has two interpretations. One is 'there is no women who can drive a car' and the second is 'not all women can drive a car, but some can drive a car'.

(20) ساری عورتیں گاڑی نہیں چلا سکتیں
saaree AorateyN gaaRee naheeN chalaa sakteeN
All women cannot drive a car

Sentence (21) has logical interpretation that 'each car is in a separate house, and there are as many houses as many cars are' but the sentence in (22) has logical interpretation that 'each car is in the same parking or there are many cars in one

parking'. The lexical and syntactic structure of these sentences is the same, but these require semantic or real world knowledge for interpretation.

(21) ہر گاڑی گھر میں کھڑی ہے
har gaaRee ghar meyN khaRee hay
Each car is parked in a house. (The cars are parked in the houses).

(22) ہر گاڑی پارکنگ میں کھڑی ہے
har gaaRee paarkeng meyN khaRee hay
Each car is parked in a parking. (The cars are parked in a parking).

2.2.5 Reference or Anaphoric Ambiguity

Anaphora is to refer to objects that have previously been mentioned in a discourse. The pronoun appearing in the sentence needs to bind with its antecedent in order to remove the ambiguity involved. Anaphora resolution is a challenging problem (Khan 1995).

(23) Akram was hungry and Ajmal was late from his work. He entered a restaurant.

The sentences in (23) have two pronouns 'his' and 'he', which need to refer to a noun. The pronoun 'his' may refer to 'Ajmal' which is the only masculine noun in the same clause. However, for the resolution of 'he' we need to refer to previous sentence. Both Ajmal and Akram are good candidates for binding, but if we go in semantics then only Akram could be referred to by 'he' in the second sentence, because Ajmal was already late from his office and has no reason to enter a restaurant. However, Akram was hungry and he had a reason to enter a restaurant. Still Ajmal could be a candidate for binding if he works in a restaurant.

(24) After Raheem proposed to Maria, he found a *nikah-khwan* and they got married. For the honeymoon, they went to Murree.

The sentences in (24) have three pronouns 'he', 'they' and 'they' which need binding with nouns. The first pronoun 'he' could be bound to Raheem easily as it is the only masculine noun. The pronoun 'they' refers to two or more nouns, so it could refer to all three nouns, i.e., Raheem, Maria and the '*nikah-khwan*' or to any two of them. If somehow we capture semantic knowledge that marriage is between a male and a female, then we are left with two combinations for binding with pronoun 'they' Raheem-Maria and Maria-'*nikah-khwan*'. Again, there is a question whether the same persons who got married went for honeymoon. It will require the world knowledge about the relationship between a marriage and a honeymoon. Thus binding of pronouns or anaphora resolution is deeply rooted into semantic knowledge base or ontology network available for a particular area under discussion of a language.

(25) Ahmad was washing his face. He saw him in the mirror.

The distinction between various pronoun categories, namely, personal, genitive, reflexive, demonstrative, and relative is useful for anaphora resolution. The sentences in (25) have two pronouns ‘he’ and ‘him’. The use of relative ‘him’ instead of reflexive ‘himself’ shows that ‘he’ and ‘him’ are referring to different persons. If ‘he’ has been somehow bound to a person ‘n’, then ‘him’ will not refer to the person ‘n’.

2.3 Historic Landmarks

Some of the historical landmarks related to Machine Translation (MT) are listed below (Hutchins and Somers 1997):

1629 René Descartes proposed an idea about unambiguous universal language based on logical principles.

1668 John Wilkins elaborated interlingua

1933 Russian Petr Smirnov-Troyanskii patented three stages for transforming, source word into base form and then to words into other-language equivalents

1939 Bell Labs demonstrated the first electronic speech-synthesizing device at the New York World's Fair

1949 Warren Weaver drafted his ideas on MT for peer review outlining the prospects of machine translation (MT)

1952 Yehoshua Bar-Hillel, organized the first MT conference at MIT

1954 Georgetown University & IBM collaborated for first public demonstration of Machine Translation, where 49 Russian sentences were translated into English using a 250-word vocabulary and six grammar rules.

1960 Bar-Hillel published a paper, in which he criticized and argued that due to ‘semantic barriers’ accurate translation systems are not possible

1964 Automatic Language Processing Advisory Committee (ALPAC) is formed by US Government sponsors to examine MT's feasibility

1966 ALPAC concludes that MT is slower, inaccurate and expensive. The outcome is a halt in federal funding for machine translation R&D

1967 L. E. Baum develops hidden Markov models, the mathematical backbone of continuous-speech recognition and statistical MT

1968 Peter Toma starts one of the first MT companies, Language Automated Translation System and Electronic Communications (LATSEC)

1969 Charles Byrne and Bernard Scott found Logos to develop MT systems

1970 Peter Toma develops SYSTRAN for Russian-English Translations

1976 SYSTRAN for English-French translations is developed

1982 Janet and Jim Baker found NEWTON, Massachusetts-based Dragon System

1983 The Automated Language Processing System (ALPS) is the first MT software for a microcomputer

1987 In Belgium, Jo Lernout and Pol Hauspie found Lernout & Hauspie

1988 Researchers at IBM's Thomas J. Watson Research Center revive statistical MT methods that equate parallel texts, then calculate the probabilities that words in one version will correspond to words in another

1991 The first translator-dedicated workstations appear, including STAR's Transit, IBM's Translation Manager, Canadian Translation Services' PTT, and Eurolang's Optimizer

1992 ATR-ITL founds the Consortium for Speech Translation Advanced Research (C-STAR), which gives the first public demonstration of 'phone translation' between English, German, and Japanese

1993 The German-funded Verbmobil project gets under way. Researchers focus on portable systems for face-to-face English-language business negotiations in German and Japanese.

BBN Technologies demonstrates the first off-the-shelf MT workstation for real-time, large-vocabulary (20,000 words), speaker-independent, continuous-speech-recognition software.

1994 Free SYSTRAN machine translation is available in select CompuServe chat forums

1997 AltaVista's Babel Fish offers real-time SYSTRAN translation on the Web

2.4 Machine Translation Architectures

According to the architecture or process, machine translation is divided into direct words translation, syntactic transfer, semantic transfer and interlingua based architectures. These form related levels in the form of a standard pyramid diagram of machine translation (Hutchins and Somers 1997) as shown in Figure 2.1.

2.4.1 Direct Words Transfer

In direct words transfer process, the words of the source language are directly translated to the target language words by means of bi-lingual dictionaries. This is the simplest form of word-to-word mapping form of machine translation process, which is suitable only for those languages, which are syntactically and semantically close to each other. For example, this method is suitable for machine translation between Urdu and Hindi languages. The English sentence shown in (26) may be translated by this method to an Urdu sentence as shown below:

(26) This is a book
yeh hay aeyk ketaab
یہ ہے ایک کتاب

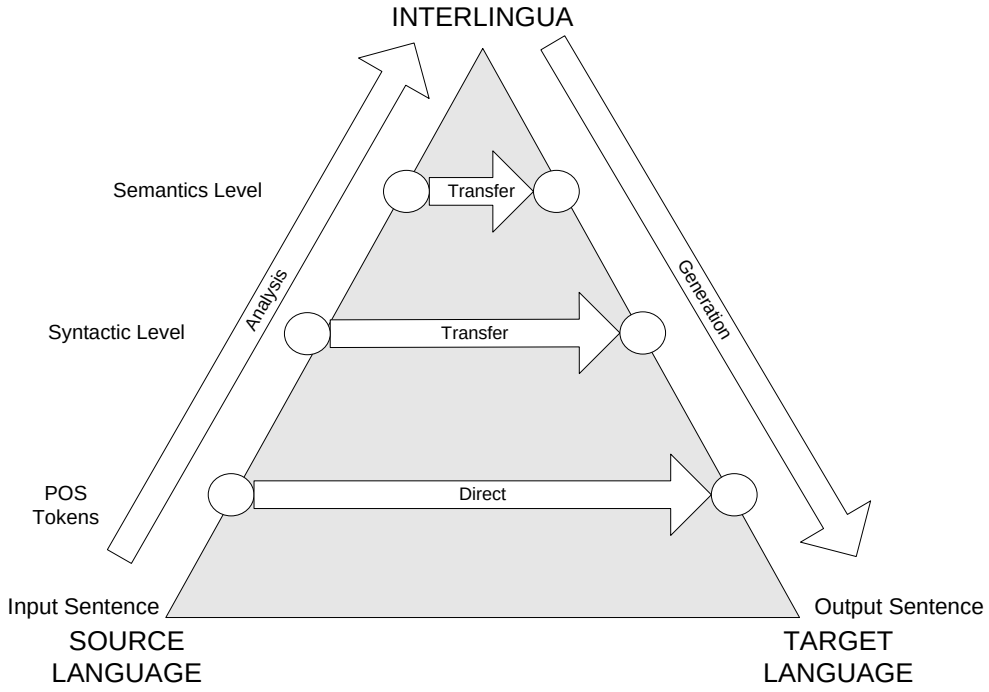


Figure 2.1: Machine Translation Architectures

2.4.2 Syntactic Transfer

In the syntactic transfer architecture, words of the source language are collected into syntactic categories, mostly in the form of trees and other representations like labeled bracketing or hierarchical matrices. These syntactic representations are then mapped to syntactic representations of the target language using mapping rules. The grammars used to map the syntactic tree of the source language to the target language are called 'transfer grammars'. Finally, the syntactic representation of the target language is mapped to a sentence in the target language. This type of transfer architecture, because of various syntactic differences, is needed for the machine translation between most of the natural languages. Although the Urdu translation of the sentence in (26) is acceptable but the correct syntactic translation of that sentence is achieved after the syntactic analysis as shown in (27). Moreover, the direct word translation process can translate only simple sentences.

(27) $[\text{This}]_{\text{SUB}}$ $[\text{is}]_{\text{VERB}}$ $[\text{a book}]_{\text{OBJ}}$
$[\textit{yeh}]_{\text{SUB}}$ $[\textit{aeyk ketaab}]_{\text{OBJ}}$ $[\textit{hay}]_{\text{VERB}}$
یہ ایک کتاب ہے

It is depicted in the pyramid figure that the difference between the source language and the target language is lesser at the syntactic level as compared with the

direct word transfer level. Therefore, the results of machine translation are expected to be better at syntactic level as compared to direct words transfer level.

2.4.3 Semantic Transfer

If the transfer between the source language and the target language is made after the semantic analysis of the source language has been performed, and semantic information in the form of a knowledge representation structure of the source language has been transferred to a semantic structure of the target language, then it can be seen from the pyramid diagram that difference between the source and the target languages at the semantic level is even lesser than the difference at the syntactic level. At the semantic level if we see the difference between the source and the target languages and the effort required to go to the next interlingua level, then we may conclude that machine translation at semantic level is acceptable for most of our MT requirements.

2.4.4 Interlingua

In an ideal MT process or architecture, the source language is fully translated to an intermediate language, called interlingua, which is supposed to represent every meaning of both source and target languages. As we go up the pyramid in Figure 2.1 the gap between source and target languages decreases, while the effort involved in analysis and generation increases. For most of the MT applications, it is found that syntactic or semantic transfer approach is acceptable.

2.5 Machine Translation Phases

The machine translation process is divided into two phases: the analysis phase and the generation phase.

2.5.1 Analysis

The tokenization, syntactic analysis and semantic analysis phase, up to the interlingua, shown in Figure 2.1, is the analysis phase of machine transfer. In this phase, the sentence is tokenized into words. The words are categorized into lexical categories known as part of speech, POS. The morphological analysis is performed to find various forms of the same word. The syntactic analysis is performed to find the structure of grouping of words into larger syntactic units, called phrases. The valid grouping of phrases to form a sentence is checked. The semantic analysis of the source language text is performed to extract meaning from the words and structural units of the text. For the interlingua process, the semantic structures are converted to interlingua.

2.5.2 Generation

The generation means conversion of a computational representation, i.e., interlingua, semantic structure or syntactic structure into a sentence in the target language. The grammar for this phase is called the 'generation grammar'. The generation is a reverse of analysis process as shown in Figure 2.1.

2.6 Machine Translation Paradigms

According to the handling or modeling of the problem, machine translation paradigms are broadly classified into linguistics based, non-linguistics and artificial intelligence based machine translation approaches. Recently, hybrid approaches, which are a combination of basic approaches, are becoming popular. Although the classification presented here does not have a clear boundary, and concepts seem to overlap, yet the given classification is based on the primary approach involved for accomplishing the machine translation.

2.6.1 Linguistics Based Approaches to Machine Translation

The approaches, which incorporate strong linguistic knowledge to drive the modeling process, are classified as linguistics based approaches to machine translation. These approaches heavily enforce universal grammatical features in the modeling of natural language grammars. Emphasis is on modeling of analysis, transfer and generation grammars based on knowledge that human posses about a language. There are many distinct theories for the modeling of grammars for various world languages, each one presents its own way of modeling language, and hence a separate route to machine translation. The modeling of Urdu language based on grammar theories will be discussed in the next chapter. Some of them, which are stronger and more popular in describing various natural language requirements, are briefly introduced here:

Transformation Based Linguistics Approaches

The transformation based linguistics approaches consider that there is a 'basic structure' of the sentences in a language and this 'basic structure' can be generated by context free grammar rules and the given lexicon. If there are other valid sentences in the language, then those can be transformed to basic structures using transformational grammar. There exist transformations in 'transformational linguistics' that can convert a normal sentence into a question sentence or into a passive sentence.

Initially presented by Chomsky, the earlier versions of transformational generative grammar (1960–1990) have changed significantly. Yet, the basic nature of transformational rules that map 'base/deep phrase structures' to 'surface phrase

structure' remains intact. The changes to framework are recorded (Chomsky 1993) as follows:

1955–1964	Early Transformational Grammar
1965–1970	The Standard Theory
1967–1974	Generative Syntax
1967–1980	The Extended Standard Theory
1980–date	Government and Binding Theory

Constraint Based Linguistics Approaches

The constraint based linguistics approaches apply constraints to context free grammar rules. Lexical Functional Grammar (LFG) (Bresnan 1982; Bresnan 2001) was developed in 1979 and still its initial concepts are well grounded. In LFG based approach of modeling natural languages, each node is attached with (optional) functional schemata in addition to lexical entries. These functional schemata allow generation of functional structures parallel to constituent structures by special mapping functions.

The Head Driven Phrase Structure Grammar (HPSG) (Sag, Wasow et al. 2004) considers features structures headed by a particular syntactic category. The feature structures interact and unify with each other using rules.

Rule Based Machine Translation (RBMT)

The Rule Based MT (RBMT) paradigm is associated with systems that rely on different linguistic levels of rules for translation between a source and a target language. The prototypical example is Rosetta (Rosetta 1994), an interlingual system which divides translation rules into two categories – The M-rules: which are meaning preserving rules, which map between syntactic trees to underlying meaning structures; and the S-rules: which are non-meaningful rules and map lexical items to syntactic trees. The former are used for compositional or regular phenomena and the latter are used for non-compositional or exceptional phenomena.

2.6.2 Non-Linguistics Approaches to Machine Translation

The main driving mechanism in these approaches is a non-linguistics approach. Although at some level these have to incorporate language features as they are modeling those, but main motivating theory is not well grounded in linguistics. Typically, these approaches utilize a large monolingual or bilingual text corpora to extract features using various computational algorithms for pattern recognition.

Statistics Based Machine Translation (SBMT)

The machine translation based on statistical analysis of parallel corpora of bilingual text falls into the category of Statistics Based Machine Translation (SBMT).

It utilizes conditional probability theory and particularly uses the famous Bayes' Rule to find conditional probabilities of word sequences for a sentence of a source language sentence to the corresponding word sequences for a sentence of the target language (Manning and Schütze 2003).

Example Based Machine Translation (EBMT)

The Example Based Machine Translation (EBMT) system employs the parallel corpora of the bilingual text to find a correspondence between the source and the target language sentences and phrases. It captures a database of example patterns of sentences and phrases of the source and the corresponding sentences and phrases of the target language. For translation it searches for the source language sentence pattern in the database, if found it gives translation using corresponding target language pattern available in the database.

2.6.3 Artificial Intelligence Based Approaches to Machine Translation

The main features of the AI based approach for MT include the application of semantic parsing (based on semantic categories, e.g. 'human', 'liquid', etc.), the building of semantic (or conceptual) representations of the meanings of texts, and the use of knowledge databases to assist in the interpretation of texts. Typically included in the latter are representations of conventional event schemata (e.g. what happens when going to a restaurant), normal inference patterns, and common sense expectations. It employs techniques, which primarily utilize established AI techniques like semantic networks, expert systems, neural networks, predicate logic. For AI persons language 'understanding' is a key to building a good MT system.

Knowledge Based Machine Translation (KBMT)

The system or network to represent 'knowledge' is the base for KBMT. The knowledge is extracted from the input sentences and used during analysis and generation phases. During 1980's at Carnegie Mellon University natural language understanding systems were developed with the help of AI community. AI community's effort to find language independent knowledge representations resulted in AI based interlingua for knowledge representation. They considered MT beyond pure linguistics information. Many attempts were made in various Universities around the world using this paradigm.

Neural Network Based Machine Translation

Work has been done with neural network technology for machine translation chores, such as, parsing, lexical disambiguation and learning of grammar rules. The incorporation of neural networks and connectionist approaches into machine

translation systems is a relatively new area of investigation. Most of the work carries out some tests with small vocabularies of the words and handles simple syntax. Handling large vocabularies and grammars significantly inflates the size of the neural networks and the training set, as well as the training time. In contrast with the other approaches described here, no realistic MT Systems have been built based solely on neural network technology. This technology is thus more of a technique than a system approach (Dorr 2000).

2.6.4 Hybrid Paradigms

Recent trend has been to make use of different mixes of goods in each paradigm and to avoid difficulties of each one of them. For example, the recent data oriented parsing technique (Bod, Scha et al. 2003) employs statistical techniques with linguistics grammars. Moreover, the statistical techniques are not good in analyzing long distance dependencies, while linguistics techniques have formulations for those. Similarly, example base machine translation has difficulties with complex sentence constructions (Dorr 2000).

2.6.5 Other Paradigms

Shake & Bake Machine Translation (Beaven 1992) and Generate & Repair Machine Translation (Naruedomkul and Cercone 2002) paradigms are similar to each other. The basic approach followed is not to spend much on analysis of the source language. After tokenization of the source language text, the text is transferred to the target language using direct words translation method by using bi-lingual dictionary. In shake (or generate) step the target language words are reordered in a new sequence under the generation grammar rules of the target language. The new words, like preposition or auxiliaries are added or word forms are replaced in the bake (or repair) step until a valid sentence is produced. If a valid target language sentence is not produced in the bake (or repair) step then shake and bake (generate and repair) continues until a valid sentence is produced.

2.7 MT Route Followed in this Thesis

This thesis does not develop a complete machine translation system, however, the computational grammar of Urdu developed in this work could be used in developing an MT system. For developing a computational grammar, the constraint based linguistics grammar development approach for the grammar-modeling of Urdu language is adopted due to the following main reasons:

Statistical language modeling techniques employ various sampling techniques on large corpora of textual data. When this research work was initiated, the Urdu text corpora were not available. Text corpora are the basic requirement for non-linguistics

based approaches. Recently Urdu text is becoming available through BBC, Jang newspaper websites in Unicode and books written in 'inpage' software, which can now be employed for statistical based analysis. Still a lot of work is needed to build parallel bilingual corpora of Urdu and English before statistical algorithms can be utilized.

The linguistics-based grammar modeling tries to capture the actual phenomena in the language as known to humans by studying various constructions in the language. The structure is studied by comparing different instances of valid and invalid sentences, which is a manual observation process. Based on various constructions a grammar rule is developed under the grammar theory. This manual comparison procedure of finding language structure is difficult to model, but if modeled it is expected to be more accurate and reliable.

The linguistics-based grammar development takes a long time for given language under consideration, but the phenomenon captured can be reused across the whole range of natural language processing applications. The statistics-based and example-based language modeling techniques, on the other hand, employ computational techniques instead of manual comparison to capture features necessary for the given application at hand for the given data, and porting this to other NLP applications and using other data reduces its accuracy significantly.

The constraint based lexicalist approaches handle wide variety of natural language phenomenon in a uniform manner without altering the 'surface structure' of the given sentences. These approaches are good for comparing structure of words and sentences in different languages. These could be used to build parallel grammars for different languages, which could be employed to achieve machine translation. The Lexical Functional Grammar based transfer between source and target languages at f-structure level is more reliable because it is close to interlingual approach.

LFG based grammar development need not change if the language pair for MT is changed, i.e., if we develop LFG grammar for MT between Urdu and English, the grammar will be the same if we add another language, say, German.

Chapter 3

GRAMMAR MODELING

The contemporary linguists' approach is that a sentence is *acceptable* if native speakers say it sounds good. Thus, if a majority of native people accept a sentence to be valid, then the sentence is considered good. In the view of formal language theorists, the sentence is *grammatical*, with respect to a grammar under consideration, if the grammar permits it by generating the parse tree of the sentence. The grammar should not only accept good sentences but also reject bad sentences. A grammar is good if it accepts good sentences, rejects bad sentences, has fewer rules and the parse tree generated by it is compact.

Mathematical modeling of the grammar of a natural language is one of the solutions for artificial comprehension by a machine. The mathematically simplest representation of a natural language grammar is a set of all the valid sentences in the given natural language. As infinite number of sentences can be generated for any given natural language, so this approach is clearly an infeasible solution. To make a large set of valid sentences will not only require a huge storage space but also searching for valid or invalid sentences will be time expensive. Thus, the solution is neither feasible for space nor for time requirements.

Next, we consider formal grammar theory proposed by Chomsky. The simplest formal grammar in the Chomsky hierarchy is the regular grammar (Hopcroft and Ullman 1979; Martin 1991). Although regular grammar can be used for modeling *morphotactics* for words in the lexicon of the natural language and thus can handle morphology requirements, but phrase structure and syntax is beyond the descriptive power of this class of grammars. The fact is proved in books of formal grammar theory under the heading 'pumping lemma for regular grammars'. Therefore, using regular grammar for modeling natural language syntax is similar to modeling a circle using a single straight line.

The languages defined by context-free-grammars (CFG) rules are one class higher in the Chomsky hierarchy from the class of regular-languages. The CFG's descriptive power is similar to modeling a circle using many straight lines, which means that CFG can model natural languages using large set of rules, but still it approximates the actual phenomenon. However, the CFG rewriting rules are fully capable of representing programming languages. Other problems of CFG based

modeling of natural languages will be given at the end of this section, after introducing some linguistics properties of natural languages. We start with the small fragment of phrase structure rules for the Urdu grammar based on CFG as shown in (28) and lexicon entries corresponding to this grammar are shown in (29):

(28)
$$\left.\begin{array}{lcl} S & \rightarrow & NP^* \; V \\ NP & \rightarrow & N \; CM \\ NP & \rightarrow & N \end{array}\right\}$$

(29)
N → حامد
N → کتاب
N → ناول

CM → نے

V → خریدی
V → خریدا

Each production in (28) consists of a rewrite rule. Each symbol on the left hand side of arrow (→) called non-terminals can be replaced with symbols on the right hand side of the arrow. The Kleene star (*) denotes zero or more repetitions. The Symbol S stands for sentence, NP for noun-phrase, CM for case-marker and V for verb. The verb V in Urdu is usually a derived form from the basic 'مصدر' (*maSdar*) form in Urdu using predefined Urdu rules of morphology. It contains information about tense, gender and number involved. In Urdu, it may be a complex-predicate construction (Butt 1995). The words or lexical items like حامد, کتاب, نے, and خریدی are terminals. Each non-terminal must be replaced with some terminal to generate a sentence in represented language. Using bottom-up parsing technique, the phrase structure tree (also called parse tree) of sentence is shown in (30). The resultant parsed tree is shown in Figure 3.1:

(30) حامد نے کتاب خریدی
Haamed ney ketaab xareedee
Hamid bought the book.

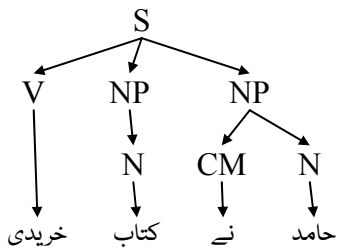


Figure 3.1: Phrase Structure of a Sentence '*Haamed ney ketaab xareedee*'

The shown parsed sentence in Figure 3.1 is grammatical as per reference grammar shown in (28) as well as according to the rules of traditional Urdu grammar.

Parse tree assigned proper grammatical categories to the respective lexical items. However, the same CFG rules can be used for the parsing of the incorrect sentence (31) as shown in parse tree Figure 3.2:

(31) *حامد نے ناول خریدی ❖[1]
**Haamed ney naawel xareedee*
Hamid bought the novel.

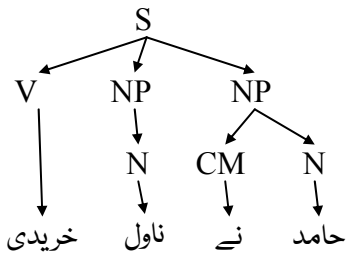


Figure 3.2: Phrase Structure of a Sentence '*Haamed ney naawel xareedee*'

To handle gender and number agreement through CFG we can change the grammar given in (28) by incorporating more specific categories of verbs and nouns as given in (32), which is not covering full agreement problem in Urdu, but just the object-verb agreement, without case marking:

(32)
$$\begin{cases} S \rightarrow NP^* \ NP\_sg\_masc \ V\_sg\_masc \\ S \rightarrow NP^* \ NP\_sg\_fem \ V\_sg\_fem \\ S \rightarrow NP^* \ NP\_pl\_masc \ V\_pl\_masc \\ S \rightarrow NP^* \ NP\_pl\_fem \ V\_pl\_fem \\ NP\_sg\_masc \rightarrow N\_sg\_masc \\ NP\_sg\_fem \rightarrow N\_sg\_fem \\ NP\_pl\_masc \rightarrow N\_pl\_masc \\ NP\_pl\_fem \rightarrow N\_pl\_fem \\ NP \rightarrow N \ CM \\ NP \rightarrow N \end{cases}$$

(33) N → حامد CM → نے V_sg_fem → خریدی
N_sg_fem → کتاب V_sg_masc → خریدا
N_sg_masc → ناول

(34) حامد نے ناول خریدا
Haamed ney naawel xareedaa
Hamid bought the novel.

❖[1] The asterisk symbol (*) is used to represent a 'grammatically incorrect' sentence or a syntactic unit.

The incorrect sentence (31) is corrected in (34) and the parse tree of correct sentence based on CFG, given in (32), and lexicon, given in (33), is shown in Figure 3.3. The parse tree of incorrect sentence cannot be generated for the modified CFG.

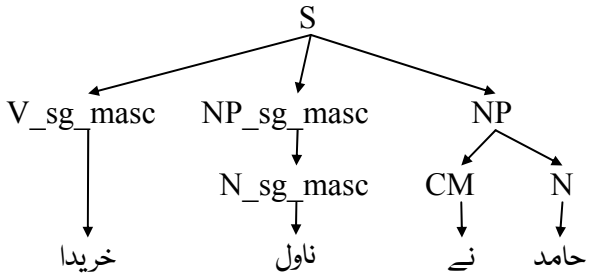


Figure 3.3: Phrase Structure using CFG in (32)

Just to handle object-verb gender-number agreement through CFG, we have to increase the number of CFG rules, whereas for LFG fewer rules and fewer part of speech (POS) categories are needed. Moreover, LFG overtly encodes linguistics information and enables manipulation and organization of linguistics phenomenon.

We have seen that CFG is useful in generating parse tree of grammatical sentences but it could also allow various other sentences that are grammatically incorrect. The verb agreement with the gender or number of a noun in the object position is not required in English language. While in Urdu, the verb form may change if the gender and number of a noun is different. The CFG needs many rules to take care of such grammatical functions, therefore, it is not preferred over LFG for the modeling of natural language grammar.

Now we go one class higher in Chomsky hierarchy (Hopcroft and Ullman 1979; Martin 1991) and model our Urdu grammar rules based on context-sensitive-grammar. Although specific examples are not presented here, but it has been shown (Luger and Stubblefield 1998) that context sensitive grammar requires a large set of rewriting rules making the parsing expensive and impractical to implement.

English grammar-modeling has various linguistics requirements, a few of these are as follows:

- Verb form needs to agree with third person subject noun in present tense.
- Verbs have different transitivity and require different number and type of complements or modifiers.
- Coordination between phrases requires phrases of the same nature.

Some of the Urdu grammar modeling requirements, in addition to the above-mentioned English modeling requirements, are as follows:

- Verb form needs to agree sometimes with subject noun and sometimes with object noun in various tenses/aspects.
- Nouns in Urdu bear gender; therefore, gender agreement with verb is also required, which also has dependency on tense/aspect.
- Noun-case agreement is required for perfective verb forms. The verb agrees with highest nominative noun phrase, if there is any nominative noun phrase in the sentence, otherwise verb gets default singular-masculine agreement.
- Nouns appear in different forms like nominative, oblique and vocative, which need agreement.
- Adjectives sometimes require agreement with nouns and sometimes they do not.
- Free phrase order may occur in Urdu sentences.

To accommodate the above-mentioned linguistic requirements in CFG, the following complexities are anticipated:

- To accommodate agreement requirements, the number of rules increases and so are grammatical categories. The analysis becomes cumbersome.
- The phrase structure does not represent linguistically motivated structure. The notions of grammatical functions like subject, direct and indirect object, etc., cannot be precisely represented.
- To accommodate free word order the number of permutations make number of rules even greater making implementation more ambiguous.

Therefore alternate constraint based lexicalist approaches for modeling Urdu grammar formalism are preferred, which are presented in the following sections. These approaches utilize CFG rules for parsing, but for agreement and other linguistic requirements, various rules and constraints are employed in a more efficient and natural way.

3.1 Lexical Functional Grammar (LFG)

The Lexical Functional Grammar (LFG) is an approach for modeling natural language grammar that has its ground in linguistics. The key features of LFG (Neidle; Wescoat; Bresnan 1982; Butt 1995; Bresnan 2001) are listed below:

1. Constraint based generative grammar – the constraints are applied to the phrase structure grammar.

2. Non-derivational surface oriented approach – which means that it has no transformations to change the actual structure of the sentence. It analyzes the actual order of words of the given sentence.
3. Multiple parallel levels of representations – which are related to each other through mappings.
4. Highly lexicalist theory – as it contains much of the functional information about words in the lexicon, e.g., the information of argument structure (called a-structure) of a predicate.
5. Constituent Structure – The phrase structure in the form of a tree diagram also called c-structure. It contains functional schemata attached to its nodes.
6. Unification based approach – in which features structure (called the f-structure) of mother is constructed by unification of its daughters by observing different attached constraints at each node of phrase structure.
7. Sentences' well-formedness – is indicated if the f-structure satisfies completeness, coherence and consistency conditions along with other constraint equations.

In this section, it is shown how modeling of sample Urdu grammar can take care of its linguistics characteristics using Lexical Functional Grammar. Lexical items are stored along with their syntactic category, functional schemata and the argument structure (called a-structure). Constituent structure (called c-structure) is formed from rules similar to CFG but having functional schemata attached to all the entries on the right hand side of each of the rule. The functional structure (called f-structure) can be constructed from the c-structure by using mapping functions.

3.1.1 Lexical Items and A-Structure

In LFG, lexical items are stored along with their syntactic category and functional schemata. The following list in (35) presents few of the lexical entries used as LFG based lexicon:

(35) حامد N (↑ PRED) = 'حامد'
(↑ PERS) = 3rd
(↑ NUM) = sg
(↑ GEN) = masc
کتاب N (↑ PRED) = 'کتاب'
(↑ NUM) = sg
(↑ GEN) = fem

ناول	N	(↑ PRED) = 'ناول'
		(↑ NUM) = sg
		(↑ GEN) = masc
خریدی	V	(↑ PRED) = 'خریدنا<(↑ SUBJ) , (↑ OBJ) >'
		(↑ TENSE) = Past
		(↑ OBJ NUM) = sg
		(↑ OBJ GEN) = fem
خریدا	V	(↑ PRED) = 'خریدنا<(↑ SUBJ) , (↑ OBJ) >'
		(↑ TENSE) = Past
		(↑ OBJ NUM) = sg
		(↑ OBJ GEN) = masc
نے	CM	(↑ CASE) = erg
		(SUBJ ↑)

The symbol ↑ refers to the predicate under which current entry is found. Each noun and verb entry has information about its number and gender. The verb entry has normal predicate form, e.g., *xareednaa* 'خریدنا' is the basic *maSdar* verb form for the singular masculine perfective form *xareedaa* 'خریدا' as well as for the singular feminine perfective verb form *xareedee* 'خریدی'. The angle brackets enclose the argument structure. The argument structure <(↑ SUBJ), (↑ OBJ)> for the predicate *xareednaa* 'خریدنا' indicates that the current predicate requires both subject and object noun phrases as required arguments.

3.1.2 C-Structure

The constituent structure (also called c-structure or phrase structure) is a parse tree, the rules for which are the same as CFG rules. However, these rules in LFG are attached with additional functional schemata with each token on the right hand side of the rules as shown in (36) below:

(36)
$$S \rightarrow \underset{(\uparrow SUBJ)=\downarrow}{NP} \quad \underset{(\uparrow OBJ)=\downarrow}{NP} \quad \underset{\uparrow=\downarrow}{V}$$
$$NP \rightarrow \underset{\uparrow=\downarrow}{N} \quad \underset{(SUBJ\ \uparrow)}{CM}$$
$$NP \rightarrow \underset{\uparrow=\downarrow}{N}$$

The symbol ↑ refers to f-structure of the mother node, while the symbol ↓ refers to f-structure of the current node. The resultant c-structure for the sentence (30) reproduced for clarity as sentence (37) is shown in Figure 3.4.

(37) حامد نے کتاب خریدی
Haamed ney ketaab xareedee
Hamid bought the book.

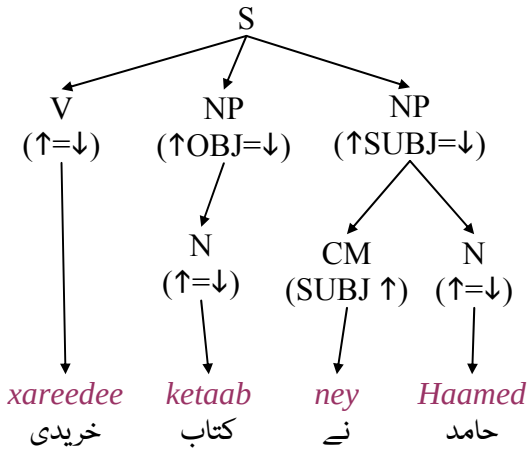


Figure 3.4: C-Structure of Sentence '*Haamed ney ketaab xareedee*'

3.1.3 F-Structure

This functional or feature structure representation, known as f-structure, is another level of LFG's syntactic representation of a sentence. It is considered language independent as it represents various features of a sentence with no reference to the actual surface and phrase structure of the sentence. The f-structure is represented using square brackets, [ ], which is an attribute-value matrix (AVM) containing entries as attribute-value pairs.

(38) $\begin{bmatrix} a_1 & v_1 \\ a_2 & v_2 \end{bmatrix}$

The attributes represent various universal grammatical functions and characteristics that are found across various natural languages. Each attribute must have a value, which may be (i) a simple value or (ii) another nested f-structure or (iii) set of values. The (39) shows these three types of attribute-values pairs.

(39) $\begin{bmatrix} a_1 & v_1 \\ a_2 & \begin{bmatrix} a_3 & v_3 \end{bmatrix} \\ a_4 & \begin{bmatrix} a_5 & \begin{bmatrix} a_6 & v_6 \end{bmatrix} \\ a_7 & v_7 \end{bmatrix} \\ a_8 & \left\{ \begin{bmatrix} a_9 & v_9 \end{bmatrix} \begin{bmatrix} a_{10} & v_{10} \end{bmatrix} \right\} \end{bmatrix}$

The attributes a_1, a_3, a_6, a_7, a_9 and a_{10} have simple values. The attributes a_2, a_4, and a_5 have f-structures as values. The attribute a_8 has a set of two f-structures as value. The set values are represented using curly brackets: '{' and '}'. The set can contain one or more values of simple or f-structure type.

A process in which two or more f-structures are combined to form a single f-structure is called unification. The operator $\uplus$ is used for unification operation. The unification contains attributes from each combining f-structures according to the following two rules:

Rule 1: If combining f-structures have different attributes, each attribute will be added to unified f-structure with the corresponding value.

(40) $$\begin{bmatrix} a_1 & v_1 & \\ a_2 & [a_3 & v_3] \end{bmatrix} \uplus [a_4 \quad v_4] = \begin{bmatrix} a_1 & v_1 & \\ a_2 & [a_3 & v_3] \\ a_4 & v_4 & \end{bmatrix}$$

Rule 2: If combining f-structures have one or more of the same attributes, each of these attributes will unify only if either (i) they have identical values or (ii) the attribute is of type set, which can hold different values of the same type.

(41) $$\begin{bmatrix} a_1 & v_1 & \\ a_2 & [a_3 & v_3] \end{bmatrix} \uplus [a_1 \quad v_1] = \begin{bmatrix} a_1 & v_1 & \\ a_2 & [a_3 & v_3] \end{bmatrix}$$

$$\begin{bmatrix} a_1 & \{v_1\} & \\ a_2 & [a_3 & v_3] \end{bmatrix} \uplus [a_1 \quad \{v_4\}] = \begin{bmatrix} a_1 & \begin{Bmatrix} v_1 \\ v_4 \end{Bmatrix} & \\ a_2 & [a_3 & v_3] \end{bmatrix}$$

However, the following unification shown in (42) results in inconsistent f-structure as the attribute a_1 has multiple values.

(42) $$\begin{bmatrix} a_1 & v_1 & \\ a_2 & [a_3 & v_3] \end{bmatrix} \uplus [a_1 \quad v_4] = \begin{bmatrix} a_1 & v_1 & \\ a_1 & v_4 & \\ a_2 & [a_3 & v_3] \end{bmatrix} \leftarrow \textit{inconsistent}\ \text{f-structure}$$

If there are nested f-structures, they may have the same attribute in the inner and outer f-structure, which may have the same or different values. The same attribute a_1 in (43) has different values v_1 and v_3, but is valid because the attribute is a member of the separate f-structures.

(43) $$\begin{bmatrix} a_1 & v_1 & \\ a_2 & [a_1 & v_3] \end{bmatrix} \uplus [a_1 \quad v_1] = \begin{bmatrix} a_1 & v_1 & \\ a_2 & [a_1 & v_3] \end{bmatrix}$$

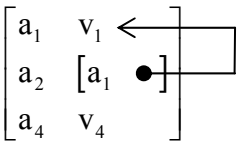
If there are nested f-structures, they may have the same attribute in the inner and outer f-structure having the same value. For example, attribute a_1 in (44) has the same values v_1. Usually, such a common value in an f-structure is shown only for one

attribute, while for the other attribute, it is represented using the same number in the box at both places or by drawing an arrow.

(44)
$$\begin{bmatrix} a_1 & v_1 \\ a_2 & [a_1 \quad v_1] \\ a_4 & v_4 \end{bmatrix}$$

By co-indexing:
$$\begin{bmatrix} a_1 & \boxed{1} v_1 \\ a_2 & [a_1 \quad \boxed{1}] \\ a_4 & v_4 \end{bmatrix}$$

By drawing an arrow:



3.1.4 Deriving F-Structure from C-Structure

Each c-structure can be mapped to the f-structure by employing the mapping function ϕ (Bresnan 1982; Butt 1995) and the unification process discussed above. The mapping function ϕ is shown in Figure 3.5 both in the form of equation and diagram.

$f_5 = f_3 \uplus [A \quad f_4]$ $\qquad$ $f_2 = f_0 \uplus f_1$

Figure 3.5: C-Structure to F-Structure Employing Mapping Function ϕ

To drive f-structure from c-structure we start from the leaf nodes. Each leaf node in c-structure is labeled with a unique number representing f-structure of the corresponding node. The leaf nodes get values of attributes from lexicon entries. For the c-structure shown in Figure 3.6, the N node will get attribute values from lexical entry for '*Haamed*', CM from '*ney*', N from '*ketaab*' and V from '*xareedee*'.

Each up arrow (↑) in Figure 3.6 is then replaced with numbered name of mother f-structure, while each down arrow (↓) is replaced with numbered name of the current node, and the result is shown in Figure 3.7. The values of leaf f-structures f_0, f_1, f_2 and f_3 constructed from LFG based lexicon shown in (35) are shown in (45) to (48)

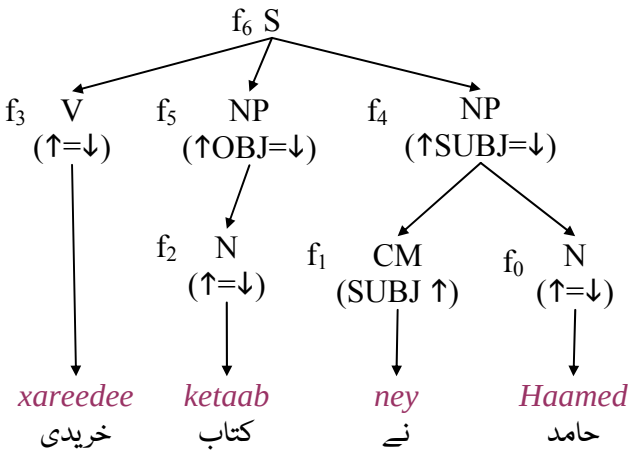


Figure 3.6: C-Structure Nodes Numbered from Leaves to Top

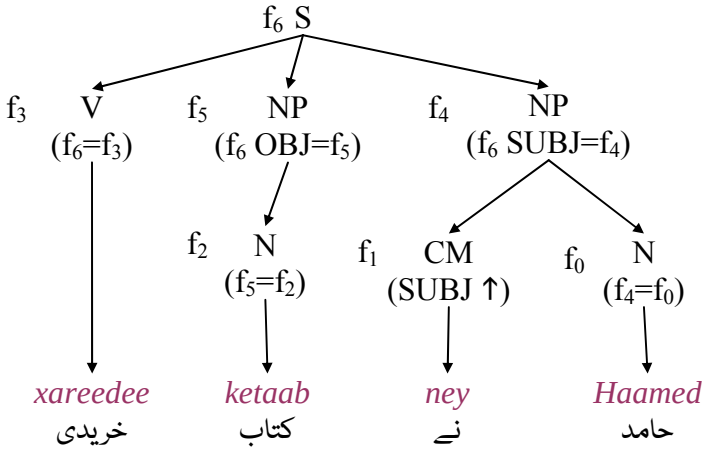


Figure 3.7: C-Structure Schemata with F-Structure Labels

(45) $$f_0 = \begin{bmatrix} \text{PRED} & \text{' } \textit{Haamed} \text{ '} \\ \text{PERS} & \textit{3rd} \\ \text{NUM} & \textit{sg} \\ \text{GEND} & \textit{masc} \end{bmatrix}$$

(46) $f_1 = [\text{CASE} \quad \textit{erg}]$ and a constraint that f_1 is the value of the attribute SUBJ of some mother node up in the hierarchy.

(47) $$f_2 = \begin{bmatrix} \text{PRED} & \text{' } \textit{ketaab} \text{ '} \\ \text{NUM} & \textit{sg} \\ \text{GEND} & \textit{fem} \end{bmatrix}$$

(48) $$f_3 = \begin{bmatrix} \text{PRED} & \text{' } \textit{xareednaa} \langle (\uparrow \ \text{SUB}), (\uparrow \ \text{OBJ}) \rangle \text{ '} \\ \text{TENSE} & \textit{past} \\ \text{OBJ} & \begin{bmatrix} \text{NUM} & \textit{sg} \\ \text{GEND} & \textit{fem} \end{bmatrix} \end{bmatrix}$$

The schemata equations in terms of f-structure labels or names, instead of up and down arrow notations derived from Figure 3.7 are shown in (49) and (50) and after unification f_4 is shown in (51), where the symbol $\uplus$ represents unification. The f-structure of sentence node S, which is f_6, solved using relation (50) is shown in (52). By substituting values of f-structures f_3, f_4, f_5 in (52), we get the final f-structure shown in Figure 3.8 that has been derived from the c-structure shown in Figure 3.4.

(49) $f_4 = f_0 \uplus f_1$
$f_5 = f_2$

(50) $f_6 = f_3$
$(f_6\ \text{SUBJ}) = f_4$
$(f_6\ \text{OBJ}) = f_5$

(51) $f_4 = f_0 \uplus f_1 = \begin{bmatrix} \text{PRED} & \text{'}\textit{Haamed}\text{'} \\ \text{PERS} & \textit{3rd} \\ \text{NUM} & \textit{sg} \\ \text{GEND} & \textit{masc} \\ \text{CASE} & \textit{erg} \end{bmatrix}$

(52) $f_6 = f_3 \uplus [\text{SUBJ}\quad f_4] \uplus [\text{OBJ}\quad f_5] = f_3 \begin{bmatrix} \text{SUBJ} & f_4 \\ \text{OBJ} & f_5 \end{bmatrix}$

$$\begin{bmatrix} \text{PRED} & \text{'}\textit{xareednaa}\ \langle(\uparrow\ \text{SUB}), (\uparrow\ \text{OBJ})\rangle\text{'} \\ \text{TENSE} & \textit{past} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'}\textit{Haamed}\text{'} \\ \text{PERS} & \textit{3rd} \\ \text{NUM} & \textit{sg} \\ \text{GEND} & \textit{masc} \\ \text{CASE} & \textit{erg} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'}\textit{ketaab}\text{'} \\ \text{NUM} & \textit{sg} \\ \text{GEND} & \textit{fem} \end{bmatrix} \end{bmatrix}$$

Figure 3.8: F-Structure derived from C-Structure

The derived f-structure must fulfill the consistency, completeness and coherence conditions for the well-formed sentences (Bresnan 2001; Dalrymple 2001)

3.1.5 Consistency Condition

For the grammatical sentence, the resultant f-structure must be consistent. During unification process if the sentence is grammatically incorrect, then the attribute from one part of the sentence will carry one value, while the same attribute from another part will carry different value, and upon unification, the values of that attribute will be inconsistent. For example, if one tries to form the f-structure of the incorrect sentence shown in (31), repeated in (53), where the attribute gender for the object of the verb is feminine in the lexicon and object itself is masculine, then the parse tree or c-structure of the incorrect sentence (53) will be generated without error as shown in Figure 3.9. However, while deriving f-structure from the c-structure, the unification will fail.

(53) *حامد نے ناول خریدی
**Haamed ney naawel xareedee*
Hamid bought the novel.

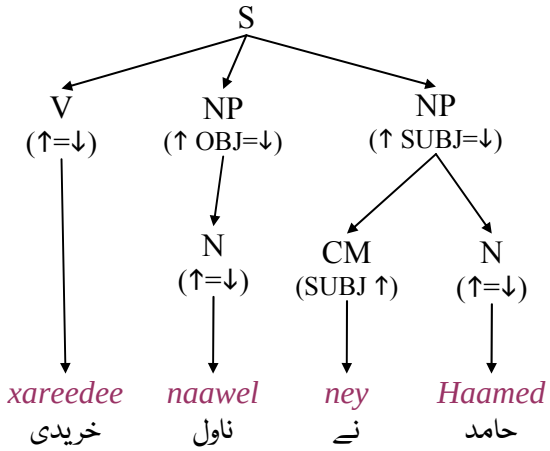


Figure 3.9: C-Structure of an Incorrect Sentence '*Haamed ney naawel xareedee*'

The attributes GEND of the object of the verb '*xareedee*' will get value 'fem'. A part of f-structure of the verb with gender attribute is shown in (54).

(54) [OBJ [GEND *fem*]]

However, the gender attribute of the noun '*naawel*' is masculine, which occupies the position of object. A part of f-structure for this noun is shown in (55).

(55) [OBJ [GEND *masc*]]

These f-structures attributes in (54) and (55) are clearly inconsistent because one attribute GEND has two different values 'masc' and 'fem' and therefore these f-

structures cannot unify, as shown in Figure 3.10, depicting the f-structure of the sentence. The source sentence (53) is rejected through consistency condition and declared grammatically incorrect, because the f-structure of a sentence in this case has inconsistent values for gender attribute. Similarly, other verb agreement requirements with object noun or with subject noun, like agreement for number, person and case could be checked.

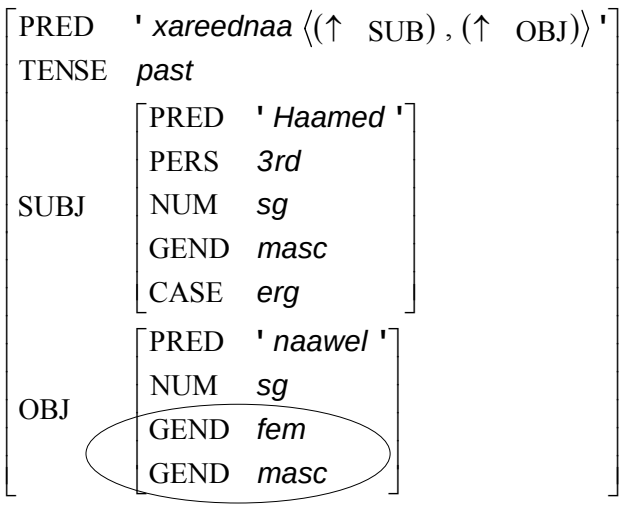


Figure 3.10: Inconsistent F-Structure of '*Haamed ney naawel xareedee*'

3.1.6 Completeness Condition

For the sentence to be grammatically well formed its f-structure must be complete, which means that the f-structure must contain all the grammatical functions mentioned in the argument structure of its attribute predicate (PRED) as attributes along with values in the same f-structure. For example, consider the following incomplete Urdu sentence shown in (56):

(56) *حامد نے خریدی
**Haamed ney xareedee*
Hamid bought

Because the predicate, *xareednaa* 'خریدنا', requires two values in its argument-structure, namely, the subject (SUBJ) argument and the object (OBJ) argument. Both attributes that appear in the argument-structure must be present in the f-structure of the sentence, along with respective values. The values of these attributes are f-structures. The OBJ argument is missing for (56) as shown in Figure 3.11, which depicts f-structure of an incomplete sentence. Grammatically, the sentence is ill-formed, because the requirement of the completeness condition for the f-structure is not fulfilled.

$$\begin{bmatrix} \text{PRED} & \text{' } \textit{xareednaa} \langle (\uparrow \ \text{SUBJ}) , (\uparrow \ \text{OBJ}) \rangle \text{ '} \\ \text{TENSE} & \textit{past} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{Haamed} \text{ '} \\ \text{PERS} & \textit{3rd} \\ \text{NUM} & \textit{sg} \\ \text{GEN} & \textit{masc} \\ \text{CASE} & \textit{erg} \end{bmatrix} \end{bmatrix}$$

Figure 3.11: Incomplete F-Structure of Sentence '*Haamed ney xareedee*'

3.1.7 Coherence Condition

For the sentence to be grammatically well formed its f-structure must be coherent. For f-structure to be coherent, it must not have superfluous grammatical functions attributes present in the f-structure, which are not mentioned in the argument structures of the its predicate, for example, consider an incoherent Urdu sentence shown in (57):

(57) * حامد جاگا کتاب

* *Haamed jaagaa ketaab*

Hamid woke up book

$$\begin{bmatrix} \text{PRED} & \text{'}\textit{jaagnaa} \langle (\uparrow \ \text{SUB}) \rangle \text{ '} \\ \text{TENSE} & \textit{past} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{Haamed} \text{ '} \\ \text{PERS} & \textit{3rd} \\ \text{NUM} & \textit{sg} \\ \text{GEND} & \textit{masc} \\ \text{CASE} & \textit{erg} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{ketaab} \text{ '} \\ \text{NUM} & \textit{sg} \\ \text{GEND} & \textit{fem} \end{bmatrix} \end{bmatrix}$$

Figure 3.12: Incoherent F-Structure of Sentence '*Haamed jaagaa ketaab*'

The f-structure of the incoherent sentence is shown in Figure 3.12. The predicate 'جاگنا<↑ SUBJ>' requires only one argument as 'subject' and other grammatical function 'object' is surplus to requirements in the resultant f-structure. As the coherence condition for f-structure is not met for the predicate, therefore the sentence is incoherent and grammatically ill-formed.

3.1.8 Constraint and Restriction Equations

In addition to above mentioned three conditions on the f-structure, there are various other types of constraints and restriction operations that can be applied to f-

structures (Bresnan 2001; Dalrymple 2001). These constraint equations and restrictions are also specified in the lexicon. The 'c' subscript with the '=' sign in the equation specifies that the equation is a constraint, as shown in (58). The f-structure must have these attribute and its value already present in the f-structure, which are specified in these constraint equations for the grammatically correct sentences. The constraint equations do not define new attributes for the f-structure as shown in (59) as compared with (61) where new attributes are defined.

(58) *xareedee* V (↑ PRED) = '*xareednaa*<(↑ SUBJ) , (↑ OBJ) >'
(↑ TENSE) = past
(↑ OBJ NUM) =c sg
(↑ OBJ GEN) =c fem

(59)
$$f_3 = \begin{bmatrix} \text{PRED} & \text{' } \textit{xareednaa} \ \langle (\uparrow \ \text{SUB}) , (\uparrow \ \text{OBJ}) \rangle \text{ '} \\ \text{TENSE} & \textit{past} \end{bmatrix}$$
$$\textit{with} \text{ constraint } \begin{bmatrix} \text{OBJ} & \begin{bmatrix} \text{NUM} & \textit{sg} \\ \text{GEND} & \textit{fem} \end{bmatrix} \end{bmatrix}$$

(60) *xareedee* V (↑ PRED) = '*xareednaa*<(↑ SUBJ) , (↑ OBJ) >'
(↑ TENSE) = past
(↑ OBJ NUM) = sg
(↑ OBJ GEN) = fem

(61)
$$f_3 = \begin{bmatrix} \text{PRED} & \text{' } \textit{xareednaa} \ \langle (\uparrow \ \text{SUB}) , (\uparrow \ \text{OBJ}) \rangle \text{ '} \\ \text{TENSE} & \textit{past} \\ \text{OBJ} & \begin{bmatrix} \text{NUM} & \textit{sg} \\ \text{GEND} & \textit{fem} \end{bmatrix} \end{bmatrix}$$

The choice whether to use defining equation or constraint equation in the lexicon depends on purpose at hand, for example, for the verb entry shown in (60) and the corresponding f-structure shown in (61), the number and gender attributes have been defined in the f-structure. In this case, object number and gender must be checked using consistency condition. However, this may fail the completeness test, as it will define attributes of object, which may not be present in the sentence. Therefore, in that situation, it is better to constrain the verb entry such that its object number is singular and its object gender is feminine, as shown in the verb entry (58) and the corresponding f-structure in (59). In this case the completeness test requires that there must be an object (from argument structure), and the object must satisfy constraint on the gender and the number (from constraint equations).

3.2 Transfer between English-Urdu F-Structures

F-structure is a syntactic representation of a sentence. At f-structure level of representation the structural difference between languages are minimal. The f-structure is theoretically considered language neutral. However, some differences arise due to differences in the structural requirements of the languages. The transfer of f-structures between two or more languages requires developing a parallel grammar. The parallel grammar project (Butt, Niño et al. 1999; Butt, King et al. 2002) is underway at various institutions around the world for the development of such parallel grammars for German, English, Danish, French, Japanese, Norwegian and Urdu languages. The minor differences between f-structures could also be sorted out algorithmically for a particular language pair and the transfer of text from one language to another, therefore, could be made between languages at f-structure level. For example, the f-structure of Urdu sentence in (37) is shown in Figure 3.8. If we consider direct transfer of predicate entries from Urdu to English the corresponding f-structure transferred to English is shown in Figure 3.13:

$$\begin{bmatrix} \text{PRED} & \text{' } \textit{buy} \ \langle (\uparrow \text{ SUBJ}) , (\uparrow \text{ OBJ}) \rangle \text{ '} \\ \text{TENSE} & \textit{past} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{Hamid} \text{ '} \\ \text{PERS} & \textit{3rd} \\ \text{NUM} & \textit{sg} \\ \text{GEND} & \textit{masc} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{book} \text{ '} \\ \text{NUM} & \textit{sg} \\ \text{GEND} & \textit{masc} \end{bmatrix} \end{bmatrix}$$

Figure 3.13: F-Structure Transferred to English from Urdu

It should be noticed that in English f-structure shown in Figure 3.13 the gender for the book is superfluous and it should be discarded. A sentence generated using English f-structure of Figure 3.13 would be:

(62) *Hamid bought book

The sentence in (62) is grammatically incorrect because English requires a determiner ‘a’ or ‘the’ before the noun ‘book’. This implies that there is some correctional mapping required when we transfer f-structure in one language to f-structure in another language. Therefore, when we map from Urdu to English (Rizvi and Hussain 2002), we may ignore gender for nouns but we need to check whether we have to add a determiner. If we are to add a determiner then proper type of determiner is required. Therefore, by adding the determiner ‘a’ and removing gender attribute

from the noun the correctly mapped f-structure is generated, which is shown in Figure 3.14. The generation of English sentence from this correctly mapped f-structure will result in correct English sentence, which is shown in (63):

$$\begin{bmatrix} \text{PRED} & \text{' } buy \langle (\uparrow \text{ SUBJ}) , (\uparrow \text{ OBJ}) \rangle \text{ '} \\ \text{TENSE} & past \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{' } Hamid \text{ '} \\ \text{PERS} & 3rd \\ \text{NUM} & sg \\ \text{GEND} & masc \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } book \text{ '} \\ \text{NUM} & sg \\ \text{SPEC} & a \end{bmatrix} \end{bmatrix}$$

Figure 3.14: Correctly Mapped English F-Structure from Urdu

(63) Hamid bought a book.

3.3 Free 'SOV' Phrase Order in Urdu

One of the classifications of languages is based on the phrasal order of subject (S), verb (V) and object (O) phrases. This S, V, O classification is mostly termed as 'word order', but more precisely it is 'phrase order' as these constituents of a sentence are basically phrases. English, Hebrew and Chinese are SVO languages. Arabic, Welsh, and Hawaiian are VSO languages. German, Japanese, Korean, Persian, Urdu and Hindi are SOV languages. For Urdu the order of phrases subject, verb, and object is actually quite free and these may occur in any order, although the SOV order is the most acceptable form. All the sentences shown in Table 3.1 are acceptable in Urdu language. The SOV order of sentences as shown in first sentences of Table 3.1 is the predominantly used order in Urdu language and sometimes mistaken as the only acceptable order. Urdu is able to exercise free word order phenomenon due to its strong case marking system, which disambiguates subject or object nouns appearing in the sentence.

Table 3.1: Free 'SOV' Phrase Order in Urdu

	Urdu Script	Roman Script	English
(64)	حامد نے کتاب خریدی	[*Haamed ney*]$_S$ [*ketaab*]$_O$ [*xareedee*]$_V$	Hamid bought the book.
	حامد نے خریدی کتاب	[*Haamed ney*]$_S$ [*xareedee*]$_V$ [*ketaab*]$_O$	
	کتاب حامد نے خریدی	[*ketaab*]$_O$ [*Haamed ney*]$_S$ [*xareedee*]$_V$	
	کتاب خریدی حامد نے	[*ketaab*]$_O$ [*xareedee*]$_V$ [*Haamed ney*]$_S$	
	خریدی کتاب حامد نے	[*xareedee*]$_V$ [*ketaab*]$_O$ [*Haamed ney*]$_S$	
	خریدی حامد نے کتاب	[*xareedee*]$_V$ [*Haamed ney*]$_S$ [*ketaab*]$_O$	

(65)	حامد نے حمید کو بلایا	[*Haamed ney*]$_S$ [*Hameed kao*]$_O$ [*bolaayaa*]$_V$	Hamid called Hameed.
	حامد نے بلایا حمید کو	[*Haamed ney*]$_S$ [*bolaayaa*]$_V$ [*Hameed kao*]$_O$	
	حمید کو حامد نے بلایا	[*Hameed kao*]$_O$ [*Haamed ney*]$_S$ [*bolaayaa*]$_V$	
	حمید کو بلایا حامد نے	[*Hameed kao*]$_O$ [*bolaayaa*]$_V$ [*Haamed ney*]$_S$	
	بلایا حمید کو حامد نے	[*bolaayaa*]$_V$ [*Hameed kao*]$_O$ [*Haamed ney*]$_S$	
	بلایا حامد نے حمید کو	[*bolaayaa*]$_V$ [*Haamed ney*]$_S$ [*Hameed kao*]$_O$	

$$\begin{bmatrix} \text{PRED} & \text{' } xareednaa \langle (\uparrow \text{ SUBJ}) , (\uparrow \text{ OBJ}) \rangle \text{ '} \\ \text{TENSE} & past \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{' } Haamed \text{ '} \\ \text{PERS} & 3rd \\ \text{NUM} & sg \\ \text{GEN} & masc \\ \text{CASE} & erg \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } ketaab \text{ '} \\ \text{NUM} & sg \\ \text{GEN} & fem \\ \text{CASE} & nom \end{bmatrix} \end{bmatrix}$$

Figure 3.15: F-Structure of Sentences in (64)

$$\begin{bmatrix} \text{PRED} & \text{' } bolaayaa \langle \uparrow \text{ SUBJ}, \uparrow \text{ OBJ} \rangle \text{ '} \\ \text{TENSE} & past \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{' } haamed \text{ '} \\ \text{PERS} & 3rd \\ \text{NUM} & sg \\ \text{GEN} & masc \\ \text{CASE} & erg \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } hameed \text{ '} \\ \text{NUM} & sg \\ \text{GEN} & masc \\ \text{CASE} & acc \end{bmatrix} \end{bmatrix}$$

Figure 3.16: F-Structure of Sentences in (65)

The resultant f-structure of a sentence in different phrase order listed in Table 3.1 is the same and is shown in Figure 3.15. It is the same because the case markers *ney* 'نے' and *kao* 'کو' are used to mark different nouns present in a sentence as subject and object, respectively, irrespective of the order in which they appear in a sentence (Rizvi and Hussain 2002). The unmarked noun case, having no case marker, is

nominative, i.e., 'nom'. The entries for case markers are shown in (66) which appear in proposed Urdu lexicon based on LFG.

(66) نے CM (↑ CASE) = erg
(SUBJ ↑)
کو CM (↑ CASE) = acc
(OBJ ↑)
the absence of case marker means that CASE = nom

3.4 Head Driven Phrase Structure Grammar (HPSG)

The Head Driven Phrase Structure (HPSG) is closely related to LFG in various features, but there are many differences between LFG and HPSG. HPSG has been greatly influenced by the Generalized Phrase Structure Grammar (GPSG). HPSG was formulated and proposed in two works of Carl Pollard and Ivan Sag (Pollard and Sag 1987; Pollard and Sag 1994), which remained reference books in the field until 2004. In 2004, a revised version of HPSG appeared (Sag, Wasow et al. 2004). The key features of HPSG 2004 are listed below:

1. Constraint based generative grammar – this means that constraints are applied to the phrase structure grammar.
2. Non-derivational surface oriented approach – it means that it has no transformations to change the actual structure of the sentence. It analyzes the actual order of words of the given sentence.
3. Unification based approach – the features of mother in a phrase structure are related to its daughters through unification, which is achieved by observing constraints and certain principles.
4. Highly lexicalist theory – contains information in the lexicon and this information is even richer than LFG.
5. Signs – feature structure are known as signs, which are attribute-value-matrices AVMs. Signs are nodes in a phrase structure rules.
6. Inheritance – signs follow an inheritance hierarchy. The sub-classes inherit attributes and their values from their super classes.
7. Head – each phrase is driven by a sign, known as head of the phrase.
8. Principles – various principles are applied during unification.

3.4.1 Signs and Inheritance

The attribute value matrices, AVMs, in HPSG are called signs (Sag, Wasow et al. 2004). Signs follow notion of object orientation. Each sign belongs to a specific type or class. A sign can be derived from another sign through inheritance. The derived sign inherits all the features of its base classes and can add more features to

the inherited features. Figure 3.17 shows an example of sign in HPSG; each sign has a particular type and contains feature-value pairs in the form of a matrix.

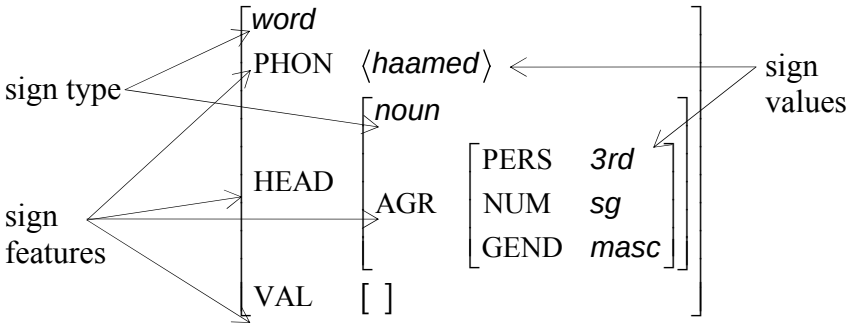


Figure 3.17: An Instance of AVM Sign in HPSG

Figure 3.18 shows a part of inheritance hierarchy of signs in HPSG. Each sign is a feature-structure, which contains feature-value pairs. The expression derived directly from feature-structure sign contains a HEAD attribute and a VAL (valance) attribute. Thus, word and phrase inherit HEAD and VAL features from expression. The word sign adds feature ARG-ST (argument-structure). The part-of-speech has AGR (agreement) feature, which is inherited to the derived classes like verb, noun.

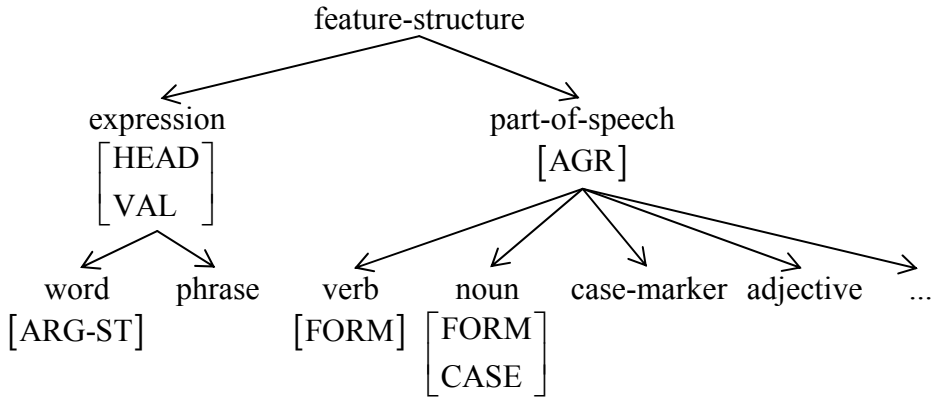


Figure 3.18: Part of Inheritance Hierarchy of Signs in HPSG

In HPSG, the features can take only specified type of values, unlike LFG where any type of value or f-structure or even set of values can be assigned to attributes. In HPSG, features or attributes cannot take any undefined value, for example, the HEAD feature can take values of type 'part-of-speech'. VAL feature can take values of type 'valance-category', which contains features COMPS (complements) and SPR (specifier). The HPSG is thus 'strictly typed' as compared with LFG.

3.4.2 Lexical Entries

The lexical entries of HPSG are quite large to display on paper. These contain phonetic, syntactic and semantic information related to a word. As an introduction here bare syntactic information is given. Lexical entries for three nouns in Urdu are shown in (67). First is a proper name, *Haamed* (Hamid). The AGR (agreement) feature of HEAD contains information about PERS (person), NUM (number) and GEND (gender). The noun has no valance requirements. The other two entries are for two nouns 'book' and 'novel' having gender masculine and feminine, respectively.

(67) Lexical Entries for Urdu Nouns in HPSG

$$\begin{bmatrix} word \\ \text{PHON} & \langle Haamed \rangle \\ \text{HEAD} & \begin{bmatrix} noun \\ \text{AGR} & \begin{bmatrix} agr\text{-}cat \\ \text{PERS} & 3rd \\ \text{NUM} & sg \\ \text{GEND} & masc \end{bmatrix} \end{bmatrix} \\ \text{VAL} & [\ ] \end{bmatrix}$$

$$\begin{bmatrix} word \\ \text{PHON} & \langle ketaab \rangle \\ \text{HEAD} & \begin{bmatrix} noun \\ \text{AGR} & \begin{bmatrix} agr\text{-}cat \\ \text{NUM} & sg \\ \text{GEND} & fem \end{bmatrix} \\ \text{FORM} & nom \end{bmatrix} \\ \text{VAL} & [\ ] \end{bmatrix}$$

$$\begin{bmatrix} word \\ \text{PHON} & \langle naawel \rangle \\ \text{HEAD} & \begin{bmatrix} noun \\ \text{AGR} & \begin{bmatrix} agr\text{-}cat \\ \text{NUM} & sg \\ \text{GEND} & masc \end{bmatrix} \\ \text{FORM} & nom \end{bmatrix} \\ \text{VAL} & [\ ] \end{bmatrix}$$

The feature GEND (gender) is additionally required in Urdu as compared with English for feature AGR in the type agr-cat (agreement-category). The feature FORM in the HEAD of nouns is also required in Urdu as compared to English, because in English nouns do not change form, while in Urdu noun appear in nominative, oblique

and vocative form. The form feature requires agreement with that of case marker that will be discussed later in Chapter 7. This feature FORM is not included in the agr-cat, because it requires separate agreement.

For words having HEAD of type verb, the HEAD feature contains agreement (AGR), FORM and CASE features. The values that verb FORM features take in Urdu are different from those of English. Similarly, CASE feature is additional from that of English. The CASE feature is not put as AGR value because CASE and AGR require separate agreements as shown in (68). The ergative CASE must match with the noun phrase of the SPR (specifier), while AGR (agreement) features of NUM (number) and GEND (gender) must match with the COMPS (complements) noun phrase.

(68) Lexical Entries for Urdu Verbs in HPSG

$$\begin{bmatrix} \textit{word} \\ \text{PHON} & \langle \textit{xareedaa} \rangle \\ \text{HEAD} & \begin{bmatrix} \textit{verb} \\ \text{AGR} & \boxed{1}\begin{bmatrix} \text{NUM} & \textit{sg} \\ \text{GEND} & \textit{masc} \end{bmatrix} \\ \text{CASE} & \boxed{2}\ \textit{erg} \\ \text{FORM} & \textit{perfect} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \begin{bmatrix} \text{NP} \\ \text{CASE} & \boxed{2} \end{bmatrix} \right\rangle \\ \text{COMPS} & \left\langle \begin{bmatrix} \text{NP} \\ \text{AGR} & \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix} \textit{word} \\ \text{PHON} & \langle \textit{xareedee} \rangle \\ \text{HEAD} & \begin{bmatrix} \textit{verb} \\ \text{AGR} & \boxed{1}\begin{bmatrix} \text{NUM} & \textit{sg} \\ \text{GEND} & \textit{fem} \end{bmatrix} \\ \text{CASE} & \boxed{2}\ \textit{erg} \\ \text{FORM} & \textit{perfect} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \begin{bmatrix} \text{NP} \\ \text{CASE} & \boxed{2} \end{bmatrix} \right\rangle \\ \text{COMPS} & \left\langle \begin{bmatrix} \text{NP} \\ \text{AGR} & \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

Lexical entries of HPSG take a large space to show on paper. In fact, each entry contains even more features in a fully specified HPSG entry. The fully specified entry is one, which shows values of all the features, even some of the features take default

values and may not be important for current discussion. There is a division of SYN (syntax) and SEM (semantic) features within each expression. The HPSG lexicon contains much information, even greater than lexical functional grammar.

3.4.3 Phrase Structure Rules

HPSG phrase structure rules are CFG based regenerative rules and thus can utilize the same CFG parsing algorithms. However, the terminal and non-terminal symbols used in CFG are not just symbols in HPSG but are AVM based signs, which contain syntactic and semantic information. There are some generalized rules and principles on phrase structures in HPSG, which restrict and control the formation of tree based on linguistic requirements such as agreement, transitivity, etc. Therefore, HPSG enforces control through feature structures and principles for well formed sentences.

In Head driven phrase structure grammar, as the name implies, one node in the phrase may act as the head node, which drives and controls the phrase. The head node may be any of the daughters in the phrase structure rule, know as 'head daughter'. Urdu is predominantly a head final language. In HPSG based rules for Urdu, head daughter is usually the last daughter. As shown in (69), verb (V) is head of sentence, post-position (P) is head of post-positional phrase and case marker (C) is head of case phrase (KP). The head daughter node is marked with capital letter '**H**'.

(69) $S \rightarrow NP^{*}\ \boxed{H}V$

$PP \rightarrow N\ \boxed{H}P$

The head daughter node is specified in order to satisfy agreement requirements of the phrase through Head Feature Principle or by co-indexing.

Head Feature Principle

The Head Feature Principle states:

(70) "In any headed phrase, the HEAD value of the mother and the HEAD value of the head daughter must be identical".
(Sag, Wasow et al. 2004)

The HEAD feature takes value of type part-of-speech (pos), which contains AGR (agreement) feature. With the use of Head Feature Principle (HFP), the agreement requirements of head daughter are transferred to the mother. By expanding symbols in (69) to signs, we get rules as shown in (71). The head value of mother is the same as that of head daughter marked with letter '**H**' by the use of HFP

(71) $\begin{bmatrix} phrase \\ \text{HEAD} \quad verb \end{bmatrix} \rightarrow \begin{bmatrix} phrase \\ \text{HEAD} \quad noun \end{bmatrix}^{*} \mathbf{H} \begin{bmatrix} word \\ \text{HEAD} \quad verb \end{bmatrix}$

$\begin{bmatrix} phrase \\ \text{HEAD} \quad pp \end{bmatrix} \rightarrow \begin{bmatrix} word \\ \text{HEAD} \quad noun \end{bmatrix} \mathbf{H} \begin{bmatrix} word \\ \text{HEAD} \quad pp \end{bmatrix}$

It is arguable in Urdu, if noun (N) and case marker (CM) make a case phrase (KP) such that CM is the head daughter as shown in (72) or these make a noun phrase (NP) such that noun is the head of phrase as shown in (73).

(72) KP → N H CM

$\begin{bmatrix} phrase \\ \text{HEAD} \quad cm \end{bmatrix} \rightarrow \begin{bmatrix} \text{word} \\ \text{HEAD} \quad noun \end{bmatrix} \mathbf{H} \begin{bmatrix} word \\ \text{HEAD} \quad cm \end{bmatrix}$

(73) NP → H N CM

$\begin{bmatrix} phrase \\ \text{HEAD} \quad noun \end{bmatrix} \rightarrow \mathbf{H} \begin{bmatrix} word \\ \text{HEAD} \quad noun \end{bmatrix} \begin{bmatrix} word \\ \text{HEAD} \quad cm \end{bmatrix}$

It is later shown in Chapter 7, that HEAD of both noun and case marker impart feature to mother HEAD, and although noun be marked as head daughter, the agreement of noun is selected by case marker and the resultant mother must have head value as noun as shown in (73). Based on the above-described consideration a modification in the HFP for Urdu is being proposed as shown in (74).

(74) "In any headed phrase, the HEAD value of the mother and the HEAD value of the head daughter must be identical, unless specified otherwise".

Valance Feature

The VAL (valance) feature is used to show that one grammatical category requires others for completion. Thus, transitivity requirements for the verbs, requirement of noun for adjectives, and determiner requirement for nouns are handled through the valance feature. The VAL feature contains two main features, the SPR (specifier) and COMPS (complements). In an English sentence, the specifier noun phrase of verb represents subject, while verb complements represent object requirements represented by verb transitivity. Since English is a SVO language, the verb splits subject and objects. In HPSG, the linear order is taken into account and if a noun phrase comes before a verb then it is taken as a subject, and other noun phrases which appear after verb are taken as objects and are also known as complements of verb in HPSG. The values of SPR and COMPS features are represented as lists so these can hold multiple values. A value or more in the valence list represents need of such an item for the completion, while empty list signals that there is no requirement for the completion.

(75) The Valance Principle
Unless the rule says otherwise, the mother's values of the VAL features (SPR and COMPS) are identical to those of the head daughter.
(Sag, Wasow et al. 2004)

The following Head-Complement and Head-Specifier rules are exception to the valance principle. Therefore, if the complements and/or specifier are found, the valance requirements of mother are satisfied, otherwise, mother will inherit the same requirement from the daughter according to 'the valance principle'.

Head Complement Rule

The head complement rule, in the form of regenerative rule, is shown in (76), which states that if a head daughter requires 'n' complements and all 'n' are identified as sisters to head daughter, then the complement requirements of the mother are satisfied.

(76) $$\begin{bmatrix} phrase \\ \text{VAL} \quad [\text{COMPS} \ \langle\ \rangle] \end{bmatrix} \rightarrow \mathbf{H} \begin{bmatrix} \text{VAL} \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle \boxed{1}, \ \cdots, \boxed{n} \rangle \end{bmatrix} \end{bmatrix} \ \boxed{1} \ \cdots \ \boxed{n}$$

However, if any one or more of 'n' complements is not found as sister to head daughter, then according to the valance principle (75), that will appear in the list of mother node and the resultant mother phrase is incomplete until its complements list is not empty.

Head Specifier Rule

The head specifier rule requires that item(s) specified in the list of SPR feature must be identified as sister to head daughter to satisfy the requirement and, thus, completing the mother phrase.

(77) $$\begin{bmatrix} phrase \\ \text{VAL} \quad [\text{SPR} \ \langle\ \rangle] \end{bmatrix} \rightarrow \boxed{1} \quad \mathbf{H} \begin{bmatrix} \text{VAL} \begin{bmatrix} \text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix}$$

The subject noun phrase acts as specifier for HEAD verb and determiner acts as specifier for HEAD noun. In Urdu, where phrase order of sentence daughter phrases is relatively free, SPR and COMPS features have no difference, but in order to keep the correspondence with English based HPSG, the SPR is used for subject and COMPS are used for objects and other noun phrases in Urdu.

3.4.4 Specifier Head Agreement Constraint

Verbs and common nouns in English HPSG are specified as shown in (78), which shows that AGR of verb or common noun must match AGR of its own

specifier. This constraint on specifier is known as 'Specifier Head Agreement Constraint', abbreviated as SHAC. Determiner is the specifier of the noun, the agreement of which is specified by the following this constraint.

(78)
$$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

In English, the verb specifier is actually a subject, SHAC is, therefore, representing agreement between a subject and a verb. Thus, this constraint is language specific. This constraint is valid only for those languages that have subject verb agreement. In Urdu language, the verb agrees with nominative subject, but if the subject is non-nominative then agreement may shift to object, and if object is non-nominative, then default singular masculine agreement is followed.

3.5 Selection of Grammar Theory

Lexical Functional Grammar (LFG) and Head-driven Pharse Structure Grammar (HPSG) – grammar-theories belong to the family of unification grammars. These apply constraints on phrase structure rules and both are lexicalist as these have much of the information specified in the lexicon. The LFG and HPSG resemble other formal grammars, such as Generalized Phrase Structure Grammar (GPSG), Tree Adjoining Grammar (TAG) or Categorical Grammar (CG) but these are considered more popular and robust. Moreover, LFG and HPSG have a regular yearly international conference with online reviewed conference proceedings on the internet. In contrast, other grammar theories have only a few papers.

The LFG based modeling is more language neutral because the underlying framework has no features or principles that are specific for a language. LFG can handle language with fixed word order and with free word order in the same manner, while HPSG is somewhat language specific. However, object oriented inheritance based hierarchy of HPSG with common system of rules is attractive for parallel grammar modeling.

LFG preserves the grammatical relations through different level of representations. Verb agreement for gender and number, transitive and in-transitive verbs, complex predicates, case marking and scrambling phenomenon in Urdu have been tested through LFG. The transfer of text between languages may be made at the f-structure level where the syntactic differences between languages are at the minimum and the f-structure level of LFG is quite language-neutral representation, therefore the grammar framework based on LFG is suitable for the successful syntactic Machine Translation between English and Urdu languages.

PART II

MORPHOLOGICAL ANALYSIS AND LEXICAL ATTRIBUTES

Chapter 4

URDU VERB CHARACTERISTICS AND MORPHOLOGY

Words are the building blocks of the grammar of a language. Morphology, also known as *Aelm-e-Sarf* 'علمِ صرف', is a branch of linguistics that deals with the internal structure of words. *Morphemes* are smallest building blocks that make words in a language. Morphemes express concepts or relationships. A morpheme that could be a meaningful whole word or a morpheme could be sequence of character(s), which is not directly meaningful until it is joined with another morpheme or a word. For example, car, table, anti–, re–, –s, –ing are morphemes. Morphemes which are not meaningful word, normally convey information about syntactic features, like number (singular, plural), tense (present, past, future) and gender (masculine, feminine). For example, in word 'flower' – the single morpheme 'flower' is recognized as the morph 'flower' to form the word 'flower'. However in word 'flowers' – the word morpheme and the plural morpheme are recognized as 'flower' and '–s' respectively, which combine to form the word 'flowers'. *Allomorphs* are the different forms of the same morpheme. For example, the plural morpheme in English has two allomorphs, –es, –s. The gerund form in English has three allomorphs, –ing (as in play–ing), –ing with e-deletion (as in sav–ing), and gemination (as in plan–ning, jog–ging).

Free morphemes are those that can stand on their own as individual words, like book, knock and soft. Bound morphemes are those that need to be attached to some 'host' morphemes to be realized as individual word. For example, the following affixes are bound morphemes, e.g., re–, –s, –ed, –ly, which cannot occur as standalone, but these impart meaningful information in words: reshape, books, knocked, softly.

If a word has various word forms and these word forms belong to a single grammatical category then these word forms are referred to as having the same 'lexeme'. For example, the words 'flower' and 'flowers' refer to the same noun 'flower'. The words 'run', 'runs', 'running' refer to the same verb 'run'. Thus, 'flower' and 'run' are lexemes for their respective forms. Free morphemes are thus usually lexemes.

Inflection morphology is the process of adding inflectional morphemes to a word. The inflectional morpheme adds some type of grammatical information, i.e.,

case, number, person, gender, mood, mode, tense and aspect. Inflectional morphology does not change grammatical category of the word and thus the inflected words refers to the same lexeme.

Derivational morphology, in contrast, adds derivational morphemes, which create a new word from an existing word, sometimes by simply changing grammatical category, i.e., changing a noun to a verb. Words generally do not appear in dictionaries with inflectional morphemes. However, they often do appear with derivational morphemes. For instance, English dictionaries list words 'readable' and 'readability', which has been derived from the root 'read'. However, most of English dictionaries do not list 'book' as one entry and 'books' as another. Similarly, English dictionaries do not list 'jump' and 'jumped' as two different entries.

Derivational morphology is thus the creation of new words out of other words and morphemes. The new words formed normally belong to a different part of speech, but not always. The words 'possible', 'possibly', 'impossible' are made by using derivational morphology. Similarly 'happy' and 'happiness', 'inform', 'informer' and 'information' are the words formed through derivational morphology.

Root is a lexical content morpheme having no affix. Root cannot be analyzed into further smaller meaningful parts. Root is common to set of all derived or inflected forms, when all of the affixes are removed. Root morpheme carries the main fraction of meaning, e.g., in words: disestablish, establishment, establishments, the word 'establish' is a root to which various derivational and inflection morphemes are attached.

A stem is the root or roots of a word, together with any derivational affixes, to which inflectional affixes can be added. For example, both 'tie' and 'untie' are stem, to which inflectional –s can be added to form 'ties' and 'unties'

Compounding is the formation of new words, which is made by combining two or more words. Each unit that combines in compounding is a lexeme in itself. Examples are: blackbird, firefighter, hardhat, water-hose, rubber-hose, and fire-hose.

Morphological analysis means finding information associated with the given word. For example, the word 'plays' is analyzed as noun 'play' in the plural form or as a verb 'play' which can be used with 3rd person, singular noun in present tense. Morphological generation is the reverse of analysis, which means given the information and the root word, generate the inflected or derived word.

4.1 Verb Transitivity and Valency

Verb Transitivity is the number of object noun phrases, in addition to subject noun phrase, required by a verb in order to make a well-formed sentence. At least one noun phrase, i.e., the subject, usually accompanies with the verb, which is not counted in the verb's transitivity. Similarly, other adverbial and post-positional phrases are

treated as adjuncts and are not counted in transitivity. The verbs requiring zero, one and two object noun phrases are termed as intransitive, transitive and ditransitive verbs respectively.

Verb Valency is the total number of arguments required by the verb. Thus, valency counts subject noun phrase, object noun phrases and other adverbial or post-positional phrases. Only those phrases are counted in valency, which are controlled by verb, thus adverbial or post-positional phrases that are not governed by the verb are treated as adjuncts. Valency is thus a general term that may apply to any other grammatical category, such as English noun, which requires a determiner and Urdu case marker, which requires a noun.

4.1.1 Intransitive Verb

Intransitive verb, *feAl laazem* (فعل لازم) describes a verb or clause that is unable to take a direct object. For intransitive verbs the work or event happening is only related to or caused by the subject or agent, *faaAel* (فاعل). The valency of intransitive verbs is one. Table 4.1 lists some of the intransitive verbs. We see that these mostly describe personal actions. These actions can be performed by oneself without any other thing needed for these actions to be performed, also these actions do not have direct effect on other things. Therefore, no object is required in a sentence formed by these verbs.

Table 4.1: Some Intransitive Verbs in Urdu

سَونا	*sao-naa*, to sleep	ہَنسنا	*hans-naa*, to laugh
کھانسنا	*khaans-naa*, to cough	رَونا	*rao-naa*, to weep
بَولنا	*baol-naa*, to speak	مَرنا	*mar-naa*, to die
دَوڑنا	*daoR-naa*, to run	گِرنا	*ger-naa*, to fell
جَاگنا	*jaag-naa*, to wake up	چھینکنا	*chheenk-naa*, to sneeze
چُھپنا	*chhop-naa*, to hide	آنا	*aa-naa*, to come
پَیدا ہونا	*paydaa hao-naa*, to be born	بَکنا	*bak-naa*, to splutter
چَلنا	*chal-naa*, to walk	بَھاگنا	*bhaag-naa*, to sprint
بَور ہونا	*baor hao-naa*, to feel bore	اُونگھنا	*aoongh-naa*, to doze
اُٹھنا	*aoTh-naa*, to rise up	تِلملانا	*telmelaa-naa*, to weary
بِلبلانا	*belbelaa-naa*, to mumble	اُکتانا	*aoktaa-naa*, to exhaust

4.1.2 Transitive Verbs

A transitive verb, *feAl motAadee* (فعل متعدی), is a verb that requires a direct object. The subject is the agent of the action being performed and direct object is undergoer, *mafAool* (مفعول), of that action. The valency of the transitive verb is two. In Table 4.2 some original transitive verbs have been listed, which are not derived morphologically from intransitive verbs.

Table 4.2: Some Transitive Verbs in Urdu

لکھنا	*lekh-naa*, to write	پڑھنا	*paRh-naa*, to read
چکھنا	*chakh-naa*, to taste	چھونا	*chhoo-naa*, to touch
سونگھنا	*soongh-naa*, to smell	پینا	*pee-naa*, to drink
بلانا	*bolaa-naa*, to call	دیکھنا	*deykh-naa*, to see
توڑنا	*taoR-naa*, to break	لیٹنا	*leyT-naa*, to lie
خریدنا	*xareed-naa*, to buy	بیٹھنا	*bayTh-naa*, to sit
بیچنا	*beych-naa*, to sell	پیدا کرنا	*paydaa kar-naa*, to give birth
بیلنا	*beyl-naa*, to squeeze	بور کرنا	*baor kar-naa*, to bore

4.1.3 Ditransitive

A ditransitive verb (فعل متعدی المتعدی) is a term, which describes a verb or clause that takes two arguments or objects. The original ditransitive verbs in Urdu are only few. Either most of the ditransitive verbs are morphologically derivable from the intransitive and transitive verbs or they are N-V compound verbs/ complex predicates. The valency of ditransitive verbs is three. Table 4.3 shows some original and compound ditransitive verbs.

Table 4.3: Some Original and Compound Ditransitive Verbs in Urdu

Original Ditransitive	
دینا	*dey-naa*, to give
لینا	*ley-naa*, to take
بتانا	*baat-naa*, to tell
بھیجنا	*bheyj-naa*, to send

Compound Ditransitive	
خرید دینا	*xareed dey-naa*, to buy & give
پیش کرنا	*peysh kar-naa*, to present
بھیج دینا	*bheyj dey-naa*, to send

4.2 Urdu Verb Morphology

Verb, *feAl* (فعل), is a word, which represents happening or doing of something. It is a predicate, which controls the type and the number of other constituents, like noun phrases and other complementary phrases present in a sentence.

4.3 Verb Forms

The same verb appears in different forms to show variations in happening of actions. The following are the forms of verb used in Urdu. Most of the verb forms have regular morphology. Some common verb forms are described in this section; other verb forms will be covered under discussion of tense, aspect and mood sections of this chapter.

4.3.1 Base or Root Form

The morpheme of Urdu verbs that do not change between different morphological forms is called a root form also known as base form. The easiest way

to recognize a root is to separate the suffix *–naa* from the dictionary form of a verb (the infinitive form). The remaining portion of an infinitive form is a root, also called *maadah* (مادہ) verb and the root form can be used to make other forms of verb by adding suffixes through morphology rules.

4.3.2 Causative Stem Forms

In Urdu and Hindi, it is well known that causative verbs are morphologically formed by the addition of suffixes to root form (Abdul-Haq 1991; Bhatt and Embick 2003; Butt 2003). The causative formation normally increases the valency or transitivity of a verb. The higher valency transitive and ditransitive verb forms, known as the causative verb forms or transitivitized verb forms are derived from lower valency verb roots by adding suffixes: *–aa*, *–waa* to the root form of the original verb. The causative verb forms are called stem forms, because all the morphology that can be applied to base or root form can also be applied to stem forms to make other forms of the verb. The causative stems are sort of new verbs as these have different, although related, meaning to the original root verbs. By causative verbs, an agent causes or forces someone, known as a patient or an intermediate agent, to do some action or change of state. Thus Urdu has a morphological causative formulation as compared to English, which engages idiomatic use of verbs like 'make', 'get', 'have', 'let' or 'help' for causatives.

To certain root forms, we can add suffix *–aa* to form causative form 1. Similarly, to certain root forms we can add suffix *–waa* to form causative form 2, using the morphology rules shown in (79). There are verb roots to which both causative forms morphemes could be appended.

(79) CausativeForm 1 = RootForm + *–aa*
CausativeForm 2 = RootForm + *–waa.*

Table 4.4: Some Divalent Verbs Derived from Univalent Verbs

Univalent Verbs		Derived Divalent Verbs	
دوڑنا	*daoR-naa*, to run	دوڑانا	*daoR-aa-naa*, to make someone run
چَلنا	*chal-naa*, to walk	چَلانا	*chal-aa-naa*, to make someone walk
ہنسنا	*hans-naa*, to laugh	ہنسانا	*hans-aa-naa*, to make someone laugh
گِرنا	*ger-naa*, to fell	گِرانا	*ger-aa-naa*, to make someone fell
چُھپنا	*chhop-naa*, to hide oneself	چُھپانا	*chhop-aa-naa*, to hide something/ someone
اٹھنا	*aoTh-naa*, to stand up, to rise	اُٹھانا	*aoTh-aa-naa*, to pick up, to raise

The above-mentioned causative formation rules can be used with many original verbs in Urdu to form higher valency causative verbs by adding the suffix *–aa* to the root form. A few of causative verbs are listed in Table 4.4. It may be seen that derived

divalent verbs, although have related meaning to the one from which they are derived, but their actual meaning and argument structures are different.

Moreover, the above-mentioned rule for transitivization of verbs is regular for most of the verb roots. However, the root of verb is changed in some cases, especially when the root form ends in a vowel or *–aag*. Examples of irregular morphology are shown in Table 4.5 below; see how root of verb is changed.

Table 4.5: Some Divalent Verbs Derived Irregularly from Univalent Verbs

Univalent Verbs		Divalent Verbs	
رونا	*rao-naa*, to weep	رلانا	*rol-aa-naa*, to make someone weep
سَونا	*sao-naa*, to sleep	سُلانا	*sol-aa-naa*, to make someone sleep
سِینا	*see-naa*, to sew	سِلانا	*sel-aa-naa*, to get something stitched
جاگنا	*jaag-naa*, to wake up	جگانا	*jag-aa-naa*, to help someone awake
بَھاگنا	*bhaag-naa*, to sprint	بَھگانا	*bhag-aa-naa*, to make someone sprint

To drive ditransitive/ trivalent verbs from intransitive/ univalent verbs the suffix *–waa* is added to root form of a verb. For some verbs, verb root has irregular form for making trivalent verb by the addition of suffix *–waa*. In Table 4.6, both regular and irregular formation of trivalent verbs, derived from univalent verbs, is shown.

Table 4.6: Some Trivalent Verbs Derived from Univalent Verbs

Univalent Verbs		Derived Trivalent Verbs	
گِرنا	*ger-naa*, to fell	گِروانا	*ger-waa-naa*, to make someone fell by someone
چَلنا	*chal-naa*, to walk	چَلوانا	*chal-waa-naa*, to make someone walk by someone
ہنسنا	*hans-naa*, to laugh	ہنسوانا	*hans-waa-naa*, to make someone laugh by someone
اٹھنا	*aoTh-naa*, to standup	اُٹھوانا	*aoTh-waa-naa*, to make someone pick someone
…	…	…	…
سَونا	*sao-naa*, to sleep	سُلوانا	*sol-waa-naa*, to make someone sleep by someone
رونا	*rao-naa*, to weep	رلوانا	*rol-waa-naa*, to make someone weep by someone
دوڑنا	*daoR-naa*, to run	دڑوانا	*doR-waa-naa*, to make someone run by someone
…	…	…	…
جاگنا	*jaag-naa*, to wake up	جگوانا	*jag-waa-naa*, to help someone awake by someone
بَھاگنا	*bhaag-naa*, to sprint	بَھگوانا	*bhag-waa-naa*, to make someone sprint by someone

To drive ditransitive/ trivalent verbs from transitive verbs, the same suffixes *–aa*, *–waa* is added to the root form of the verb. There are many verbs formed by adding suffix *–waa* to the root form of divalent verb, which take four arguments and thus function as tetravalent verbs. Table 4.7 shows regular and irregular formation of trivalent verbs from divalent verbs. Table 4.8 shows regular and irregular formation of tetravalent verbs from divalent verbs.

Table 4.7: Some Trivalent Verbs Derived from Divalent Verbs

Divalent Verbs		Derived Trivalent/Ditransitive Verbs	
پڑھنا	*paRh-naa*, to read	پڑھانا	*paRh-aa-naa*, to make someone read something
لکھنا	*lekh-naa*, to write	لکھانا	*lekh-aa-naa*, to make someone write something
…	…	…	…
سینا	*see-naa*, to sew	سلوانا	*sel-waa-naa*, to make someone sew something
بلانا	*bolaa-naa*, to call/invite	بلوانا	*bol-waa-naa*, to make someone call someone
…	…	…	…
پینا	*pee-naa*, to drink	پلانا	*pel-aa-naa*, to make someone drink something
کھانا	*khaa-naa*, to eat	کھلانا	*khel-aa-naa*, to make someone eat something
دیکھنا	*deykh-naa*, to see	دکھانا	*dekh-aa-naa*, to make someone see something
		دکھلانا	*dekhl-aa-naa*, to make someone see something
سونگھنا	*soongh-naa*, to smell	سونگھانا	*soongh-aa-naa*, to make someone smell something
چکھنا	*chakh-naa*, to taste	چکھانا	*chakh-aa-naa*, to make someone taste something
سننا	*son-naa*, to listen	سنانا	*son-aa-naa*, to make someone listen something
سمجھنا	*samjh-naa*, to understand	سمجھانا	*samjh-aa-naa*, to get someone understand something

Table 4.8: Some Tetravalent Verbs Derived from Divalent Verbs

Divalent Verbs		Derived Tetravalent Verbs	
پڑھنا	*paRh-naa*, to read	پڑھوانا	*paRh-waa-naa*, to make someone read something through someone
لکھنا	*lekh-naa*, to write	لکھوانا	*lekh-waa-naa*, to make someone write something through someone
سننا	*son-naa*, to listen	سنوانا	*son-waa-naa*, to make someone listen something through someone
پینا	*pee-naa*, to drink	پلوانا	*pel-waa-naa*, to make someone drink something through someone
کھانا	*khaa-naa*, to eat	کھلوانا	*khel-waa-naa*, to make someone eat something through someone
دیکھنا	*deykh-naa*, to see	دکھوانا	*dekh-waa-naa*, to help someone see something through someone
سونگھنا	*soongh-naa*, to smell	سونگھوانا	*soongh-waa-naa*, to make someone smell something through someone
چکھنا	*chakh-naa*, to taste	چکھوانا	*chakh-waa-naa*, to help someone taste something through someone
سننا	*son-naa*, to listen	سنوانا	*son-waa-naa*, to make someone listen something through someone
سمجھنا	*samjh-naa*, to grasp	سمجھوانا	*samjh-waa-naa*, to make someone grasp something through someone

4.3.3 Infinitive Form

The dictionary form of the verb in Urdu is infinitive form, called *maSdar* (مصدر), which contains suffix –*naa*. The infinitive form acts as a verbal-noun and it can be used in place of a noun. The normal infinitive form ends in masculine suffix –*naa*. The suffix or morphemes for feminine infinitive form and oblique infinitive form are –*nee*, –*ney* respectively. The infinitive appears in the masculine, the feminine and the oblique forms as shown in Table 4.9. It is worth to note here in Table 4.9 that

feminine infinitive form does not appear for intransitive verbs, because feminine form is only used for object agreement and intransitive verbs do not allow object to be associated with them. With all root forms or stem forms of the verb, we can use the following rules to generate infinitive forms of verb.

(80) InfinitiveForm = StemForm + *–naa*
InfinitiveForm = StemForm + *–nee*
InfinitiveForm = StemForm + *–ney*

Table 4.9: Infinitive Forms for Few Urdu Verbs

Root	English	Transitivity	Masculine	Feminine	Oblique
ہنس *hans*	laugh	intransitive	ہنسنا *hans-naa*	x	ہنسنے *hans-ney*
بول *bol*	speak	intransitive	بولنا *bol-naa*	x	بولنے *bol-ney*
سو *sao*	sleep	intransitive	سونا *sao-naa*	x	سونے *sao-ney*
پڑھ *paRh*	read	transitive	پڑھنا *paRh-naa*	پڑھنی *paRh-nee*	پڑھنے *paRh-ney*
خرید *xareed*	buy	transitive	خریدنا *xareed-naa*	خریدنی *xareed-nee*	خریدنے *xareed-ney*
دیکھ *deykh*	look	transitive	دیکھنا *deykh-naa*	دیکھنی *deykh-nee*	دیکھنے *deykh-ney*
دے *dey*	give	ditransitive	دینا *dey-naa*	دینی *dey-nee*	دینے *dey-ney*

4.3.4 Repetitive Form

The repetitive form, called *aestemraaree* (استمراری), and also known as habitual form or imperfect form (*na-tamaam* – ناتمام), is formed by adding suffixes to root or stem forms like: *–taa*, *–tee*, *–tey*, *–teeN*. Where the first part of suffix, *–t*, represents repetitive form, while the remaining portion represents gender and number agreement morphemes. The repetitive form represents the repetitive aspect of the verb, for an action, which is repeated, and it is normally used in present and past tenses. With all root forms of verb or causative stem forms of verbs, we can use the following rules to generate repetitive forms:

(81) RepetitiveForm = StemForm + *–taa*
RepetitiveForm = StemForm + *–tee*
RepetitiveForm = StemForm + *–tey*
RepetitiveForm = StemForm + *–teeN*

Although combining words is a topic of syntax that will be covered later, yet it is worth to note here that ‘feminine plural repetitive form’ is never used in combination with ‘feminine plural auxiliary’. It means if a sentence has ‘feminine plural’ subject and for agreement requirements if ‘feminine plural auxiliary’ is used

then 'feminine singular repetitive verb form' is used instead of plural form. Compare sentences (82) and (83), which require subject-agreement which is 'feminine plural', and the sentence that take 'feminine singular' verb form with 'feminine plural' auxiliary verb is correct while that uses 'feminine plural' verb form is incorrect. However, the 'feminine plural' verb form is used without auxiliary verb when there is a series of sentences in a narration. An example is shown in (84).

Table 4.10: Repetitive Forms for Few Urdu Verbs

Root	English	Transitivity	Masculine Singular	Feminine Singular	Masculine Plural	Feminine Plural
ہنس *hans*	laugh	intransitive	ہنستا *hans-taa*	ہنستی *hans-tee*	ہنستے *hans-tey*	ہنستیں *hans-teeN*
بول *bol*	speak	intransitive	بولتا *bol-taa*	بولتی *bol-tee*	بولتے *bol-tey*	بولتیں *bol-teeN*
سو *sao*	sleep	intransitive	سوتا *sao-taa*	سوتی *sao-tee*	سوتے *sao-tey*	سوتیں *sao-teeN*
پڑھ *paRh*	read	transitive	پڑھتا *paRh-taa*	پڑھتی *paRh-tee*	پڑھتے *paRh-tey*	پڑھتیں *paRh-teeN*
خرید *xareed*	buy	transitive	خریدتا *xareed-taa*	خریدتی *xareed-tee*	خریدتے *xareed-tey*	خریدتیں *xareed-teeN*
دیکھ *deykh*	look	transitive	دیکھتا *deykh-taa*	دیکھتی *deykh-tee*	دیکھتے *deykh-tey*	دیکھتیں *deykh-teeN*
دے *dey*	give	ditransitive	دیتا *dey-taa*	دیتی *dey-tee*	دیتے *dey-tey*	دیتیں *dey-teeN*

(82) انجم اور بتول کتابیں خریدتی تھیں

[anjom aor batool] *ketaab-eeN* *xareed-tee* *th-eeN*

[Anjom and Batool].*fem.pl* Book-*fem.pl* buy-*repeat.fem.sg* be.*past-fem.pl*

Anjom and Batool were used to buy books.

(83) * انجم اور بتول کتابیں خریدتیں تھیں

* *[anjom aor batool]* *ketaab-eeN* *xareed-teeN* *th-eeN*

[Anjom and Batool].*fem.pl* Book-*fem.pl* buy-*repeat.fem.pl* be.*past-fem.pl*

(84) انجم اور بتول کتابیں خریدتیں اور انہیں بیگ میں رکھ لیتیں

[anjom aor batool] ketaab-eeN xareed-teeN aor aonheyN bayg meyN rakh ley-teeN

[Anjom and Batool].*fem.pl* book-*fem.pl* buy-*repeat.fem.pl* and those-*pl* bag-*fem.pl* put-*base* take-*repeat.fem.pl*

Anjom and Batool were used to buy books and to put those in a bag.

4.3.5 Perfective Form

The perfective form is formed by just adding number and gender agreement suffix, *–aa*, *–ee*, *–ey*, *–eeN* , to the root or stem form. For verb roots that end in vowels, the morphology is not regular. The regular perfective verb forms are shown in Table 4.11 and the irregular perfective verb forms are shown in Table 4.12. With most

of the root forms or stem forms, the following rules can be used to generate perfective forms.

(85) PerfectiveForm = StemForm + *–aa*
PerfectiveForm = StemForm + *–ee*
PerfectiveForm = StemForm + *–ey*
PerfectiveForm = StemForm + *–eeN*

However, for causative stem forms the rule for singular masculine perfective form needs to add morpheme *–yaa* instead of regular morpheme *–aa* for root form. The morpheme *–yaa* is irregular form of the perfective morpheme *–aa* and this requires special phonological rules (Kaplan and Kay 1994) and may be handled using Xerox tool 'TWOLC', which is short for 'two level rule compiler'.

(86) PerfectiveForm = StemForm + *–yaa*

Table 4.11: Regular Perfective Forms for Few Urdu Verbs

Root	English	Transitivity	Masculine Singular	Feminine Singular	Masculine Plural	Feminine Plural
ہنس *hans*	laugh	intransitive	ہنسا *hans-aa*	ہنسی *hans-ee*	ہنسے *hans-ey*	ہنسیں *hans-eeN*
بول *bol*	speak	intransitive	بولا *bol-aa*	بولی *bol-ee*	بولے *bol-ey*	بولیں *bol-eeN*
پڑھ *paRh*	read	transitive	پڑھا *paRh-aa*	پڑھی *paRh-ee*	پڑھے *paRh-ey*	پڑھیں *paRh-eeN*
خرید *xareed*	buy	transitive	خریدا *xareed-aa*	خریدی *xareed-ee*	خریدے *xareed-ey*	خریدیں *xareed-eeN*
دیکھ *deykh*	look	transitive	دیکھا *deykh-aa*	دیکھی *deykh-ee*	دیکھے *deykh-ey*	دیکھیں *deykh-eeN*

Table 4.12: Irregular Perfective Forms for Few Urdu Verbs

Root	New Root	English	Transitive	Masculine Singular	Feminine Singular	Masculine Plural	Feminine Plural
سو *sao*	x	sleep	intransitive	سویا *sao-yaa*	سوئی *sao-ee*	سوئے *sao-ey*	سوئیں *sao-eeN*
رو *rao*	x	weep	intransitive	رویا *rao-yaa*	روئی *rao-ee*	روئے *rao-ey*	روئیں *rao-eeN*
جا *jaa*	گَ *ga*	Go	intransitive	گیا *ga-yaa*	گئی *ga-ee*	گئے *ga-ey*	گئیں *ga-eeN*
کھا *khaa*	x	Eat	transitive	کھایا *khaa-yaa*	کھائی *khaa-ee*	کھائے *khaa-ey*	کھائیں *khaa-eeN*
کَر *kar*	کِی *kee*	Do	transitive	کِیا *kee-aa*	کِی *kee*	کِیے *kee-ey*	کِیں *kee-N*
سِی *see*	x	Sew	transitive	سِیا *see-aa*	سِی *see*	سِیے *see-ey*	سِیں *see-N*
دے *dey*	دِی *dee*	Give	ditransitive	دِیا *dee-aa*	دِی *dee*	دِیے *dee-ey*	دِیں *dee-N*
لے *ley*	لِی *lee*	Take	ditransitive	لِیا *lee-aa*	لِی *lee*	لِیے *lee-ey*	لِیں *lee-N*

4.3.6 Subjunctive Form

The subjunctive form is formed by just adding person and number agreement suffix, *–ooN*, *–ey*, *–ao*, *–eyN*, to the root or stem form. The gender variation has no effect on the subjunctive form. The subjunctive mood expresses feelings, opinions, suggestions, desires, hopes, wishes. It is used to explain unclear, imaginary events and future happenings. The subjunctive form is used with appropriate future auxiliary to make future tense. With most of the Urdu root and stem forms, the following rules may be used to generate subjunctive forms:

(87) SubjunctiveForm = StemForm + *–ao*
SubjunctiveForm = StemForm + *–ooN*
SubjunctiveForm = StemForm + *–ey*
SubjunctiveForm = StemForm + *–eyN*

Table 4.13: Subjunctive Forms for Few Urdu Verbs

Root	English	Valency	Form 1	Form 2	Form 3	Form 4
ہنس *hans*	laugh	1	ہنسوں *hans-ooN*	ہنسو *hans-ao*	ہنسے *hans-ey*	ہنسیں *hans-eyN*
بول *bol*	speak	1	بولوں *bol-ooN*	بولو *bol-ao*	بولے *bol-ey*	بولیں *bol-eyN*
پڑھ *paRh*	read	2	پڑھوں *paRh-ooN*	پڑھو *paRh-ao*	پڑھے *paRh-ey*	پڑھیں *paRh-eyN*
خرید *xareed*	buy	2	خریدوں *xareed-ooN*	خریدو *xareed-ao*	خریدے *xareed-ey*	خریدیں *xareed-eyN*
دیکھ *deykh*	look	2	دیکھوں *deykh-ooN*	دیکھو *deykh-ao*	دیکھے *deykh-ey*	دیکھیں *deykh-eyN*

It is worth to note that the perfective morpheme *–eeN* and the subjunctive morpheme *–eyN* are ambiguous in Urdu script because these are written with the same characters but the pronunciation of these are different because of the difference in two vowel sounds, i.e., between the '*baRee yey*' and the '*chhotee yey*':

(88) انہوں نے کتابیں خریدیں
aonhooN=ney ketaab-eyN xareed-eeN
They.*pron.pl=erg* Book-*fem.pl* buy-*perf.fem.pl*
They bought books.

(89) آئیں! کتابیں خریدیں
Come! ketaab-eyN xareed-eyN
Come! Book-*fem.pl* buy-*subj.form4*
Come! Let us buy books.

4.3.7 Imperative Form

The imperative form is formed by adding number agreement suffix, –, *–ao*, *–eyN*, *–eeay* to the root or stem form. The imperative verb form (فعل امر و نہی) is used in

imperative mood, which is normally used for second persons. With most of the Urdu root and stem forms, the following rules may be used to generate imperative forms:

(90) ImperativeForm = StemForm
ImperativeForm = StemForm + *–ao*
ImperativeForm = StemForm + *–eyN*
ImperativeForm = StemForm + *–eeey*

Table 4.14: Imperative Forms for Few Urdu Verbs

Root	English	Valency	Frank (or Rude)	Formal (or Familiar)	Polite (or Respect)	More Polite (or Request)
ہنس *hans*	laugh	1	ہنس *hans*	ہنسو *hans-ao*	ہنسیں *hans-eyN*	ہنسیئے *hans-eeey*
بول *bol*	speak	1	بول *bol*	بولو *bol-ao*	بولیں *bol-eyN*	بولیے *bol-eeay*
پڑھ *paRh*	read	2	پڑھ *paRh*	پڑھو *paRh-ao*	پڑھیں *paRh-eyN*	پڑھیئے *paRh-eeey*
خرید *xareed*	buy	2	خرید *xareed*	خریدو *xareed-ao*	خریدیں *xareed-eyN*	خریدیئے *xareed-eeey*
دیکھ *deykh*	look	2	دیکھ *deykh*	دیکھو *deykh-ao*	دیکھیں *deykh-eyN*	دیکھیئے *deykh-eeey*

4.4 Verb Morphology Representation

Morphology can be represented on computer using minimal acyclic deterministic finite state automata (Mihov; Daciuk 1998) and by using lexical transducers (Karttunen 1994; Beesley and Karttunen 2003). A good work on Urdu morphology using finite state transducers has been done (Hussain 2004). Figure 4.1 shows a finite state network for Urdu verb forms. This network accounts for regular verb morphology, which can be used for most of the Urdu verbs. For irregular verb morphology, the same network may be used with little modifications in root and suffixes. Initially the verb root is categorized into four types of roots. Root 1 and root 2 forms are converted to causative form 1 by the addition of suffix *–aa*, root 2 and root 3 forms are converted to causative form 2 by the addition of suffix *–waa*. Therefore, root 2 is convertible to both causative forms, and root 4 form is not convertible to either of causative forms. The causative forms and root 4 form act as a stem form to which other suffixes are added. From stem form, we can make five verb forms namely: infinitive, perfective, repetitive, subjunctive and imperative forms. Each of these forms is further divided for gender, number, person and honor-mood depending upon the morpheme used. Thus for each stem we end up with 19 forms, and for a verb having 3 stem forms we have a total 60 forms of a verb as shown in Table 4.15 for the network shown in Figure 4.1.

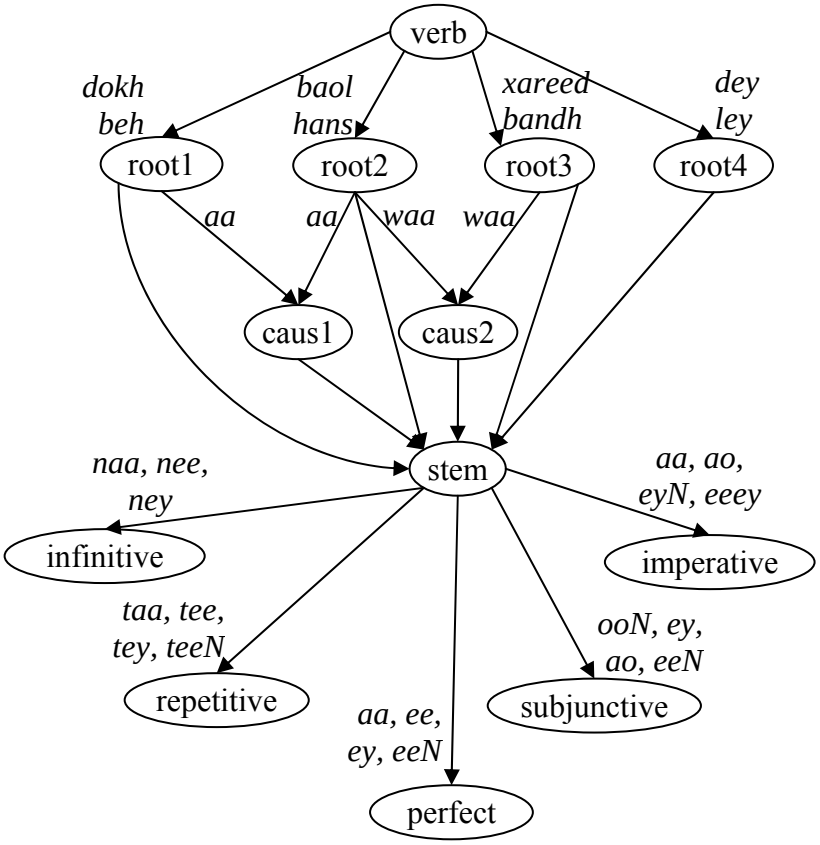


Figure 4.1: Finite State Network for Urdu Verb Morphological Forms

Table 4.15: Sixty Forms of Verb 'Read' in Urdu

Root Form	پڑھ *paRh-*			
Infinitive	پڑھنی *paRh-nee*	پڑھنے *paRh-ney*	پڑھنا *paRh-naa*	
Repetitive	پڑھتیں *paRh-teeN*	پڑھتی *paRh-tee*	پڑھتے *paRh-tey*	پڑھتا *paRh-taa*
Perfective	پڑھیں *paRh-eeN*	پڑھی *paRh-ee*	پڑھے *paRh-ey*	پڑھا *paRh-aa*
Subjunctive	پڑھیں *paRh-eyN*	پڑھوں *paRh-ooN*	پڑھے *paRh-ey*	پڑھو *paRh-ao*
Imperative	پڑھیئے *paRh-eeey*	پڑھیں *paRh-eyN*	پڑھو *paRh-ao*	پڑھ *paRh-*

Causative Stem 1	پڑھا *paRh-aa*			
Infinitive	پڑھانی *paRh-aa-nee*	پڑھانے *paRh-aa-ney*	پڑھانا *paRh-aa-naa*	
Repetitive	پڑھاتیں *paRh-aa-teeN*	پڑھاتی *paRh-aa-tee*	پڑھاتے *paRh-aa-tey*	پڑھاتا *paRh-aa-taa*
Perfective	پڑھائیں *paRh-aa-eeN*	پڑھائی *paRh-aa-ee*	پڑھائے *paRh-aa-ey*	پڑھایا *paRh-aa-yaa*
Subjunctive	پڑھائیں *paRh-aa-eyN*	پڑھاوں *paRh-aa-ooN*	پڑھائے *paRh-aa-ey*	پڑھاو *paRh-aa-ao*
Imperative	پڑھایئے *paRh-aa-eeey*	پڑھائیں *paRh-aa-eyN*	پڑھاو *paRh-aa-ao*	پڑھا *paRh-aa*

Causative Stem 2	پڑھوا *paRh-waa*			
Infinitive	پڑھوانی *paRh-waa-nee*	پڑھوانے *paRh-waa-ney*	پڑھوانا *paRh-waa-naa*	
Repetitive	پڑھواتیں *paRh-waa-teeN*	پڑھواتی *paRh-waa-tee*	پڑھواتے *paRh-waa-tey*	پڑھواتا *paRh-waa-taa*
Perfective	پڑھوائیں *paRh-waa-eeN*	پڑھوائی *paRh-waa-ee*	پڑھوائے *paRh-waa-ey*	پڑھوایا *paRh-waa-yaa*
Subjunctive	پڑھوائیں *paRh-waa-eyN*	پڑھوں *paRh-ooN*	پڑھوائے *paRh-aa-ey*	پڑھواو *paRh-waa-ao*
Imperative	پڑھوایئے *paRh-waa-eeey*	پڑھوائیں *paRh-waa-eyN*	پڑھواو *paRh-waa-ao*	پڑھوا *paRh-waa*

These sixty forms are shown with complete morphological information in Table 4.16. For verb forms mostly there is no ambiguity, however the subjunctive morpheme *–eyN* has three different tags sets and the same morpheme appears also in imperative form. Similarly, the root form and the imperative rude form are the same due to the existence of a null morpheme for the imperative rude form. The verb forms are considered different if they lie in different categories, i.e., infinitive, perfective, repetitive, subjunctive and imperative forms in the categorization shown in Table 4.16. Therefore, the subjunctive verb that ends with morpheme *–eyN* having three different tags is considered one form, while imperative form having the same morpheme is considered a separate form.

Table 4.16: Sixty Forms of Verb 'Read' with Morphological Information

Sr. No.	Urdu script	Transliteration	Morphological Information
1	پڑھ	*paRh-*	paRhnaa+V+root
2	پڑھنا	*paRh-naa*	paRhnaa+V+inf+masc
3	پڑھنی	*paRh-nee*	paRhnaa+V+inf+fem
4	پڑھنے	*paRh-ney*	paRhnaa+V+inf+obl
5	پڑھتیں	*paRh-teeN*	paRhnaa+V+repeat+fem+pl
6	پڑھتی	*paRh-tee*	paRhnaa+V+repeat+fem+sg
7	پڑھتے	*paRh-tey*	paRhnaa+V+repeat+masc+pl
8	پڑھتا	*paRh-taa*	paRhnaa+V+repeat+masc+sg
9	پڑھیں	*paRh-eeN*	paRhnaa+V+perf+fem+pl
10	پڑھی	*paRh-ee*	paRhnaa+V+perf+fem+sg
11	پڑھے	*paRh-ey*	paRhnaa+V+perf+masc+pl
12	پڑھا	*paRh-aa*	paRhnaa+V+perf+masc+sg
13	پڑھیں	*paRh-eyN*	paRhnaa+V+subj+1st+pl paRhnaa+V+subj+2nd+polite paRhnaa+V+subj+3rd+pl
14	پڑھوں	*paRh-ooN*	paRhnaa+V+subj+1st+sg
15	پڑھے	*paRh-ey*	paRhnaa+V+subj+3rd+sg
16	پڑھو	*paRh-ao*	paRhnaa+V+subj+2nd+formal
17	پڑھیئے	*paRh-eeay*	paRhnaa+V+subj+2nd+request
18	پڑھیں	*paRh-eyN*	paRhnaa+V+impr+2nd+polite
19	پڑھو	*paRh-ao*	paRhnaa+V+impr+2nd+formal

Sr. No.	Urdu script	Transliteration	Morphological Information
20	پڑھ	*paRh-*	paRhnaa+V+impr+2nd+frank
21	پڑھا	*paRh-aa*	paRhnaa+V+root +caus1
22	پڑھانا	*paRh-aa-naa*	paRhnaa+V+caus1+inf+masc
23	پڑھانی	*paRh-aa-nee*	paRhnaa+V+caus1+inf+fem
24	پڑھانے	*paRh-aa-ney*	paRhnaa+V+caus1+inf+obl
25	پڑھاتیں	*paRh-aa-teeN*	paRhnaa+V+caus1+repeat+fem+pl
26	پڑھاتی	*paRh-aa-tee*	paRhnaa+V+caus1+repeat+fem+sg
27	پڑھاتے	*paRh-aa-tey*	paRhnaa+V+caus1+repeat+masc+pl
28	پڑھاتا	*paRh-aa-taa*	paRhnaa+V+caus1+repeat+masc+sg
29	پڑھائیں	*paRh-aa-eeN*	paRhnaa+V+caus1+perf+fem+pl
30	پڑھائی	*paRh-aa-ee*	paRhnaa+V+caus1+perf+fem+sg
31	پڑھائے	*paRh-aa-ey*	paRhnaa+V+caus1+perf+masc+pl
32	پڑھایا	*paRh-aa-yaa*	paRhnaa+V+caus1+perf+masc+sg
33	پڑھائیں	*paRh-aa-eyN*	paRhnaa+V+caus1+subj+1st+pl paRhnaa+V+caus1+subj+2nd+polite paRhnaa+V+caus1+subj+3rd+pl
34	پڑھاوں	*paRh-aa-ooN*	paRhnaa+V+caus1+subj+1st+sg
35	پڑھائے	*paRh-aa-ey*	paRhnaa+V+caus1+subj+3rd+sg
36	پڑھاو	*paRh-aa-ao*	paRhnaa+V+caus1+subj+2nd+formal
37	پڑھایئے	*paRh-aa-eeay*	paRhnaa+V+caus1+subj+2nd+request
38	پڑھائیں	*paRh-aa-eyN*	paRhnaa+V+caus1+impr+2nd+polite
39	پڑھاو	*paRh-aa-ao*	paRhnaa+V+caus1+impr+2nd+formal
40	پڑھا	*paRh-aa*	paRhnaa+V+caus1+impr+2nd+frank
41	پڑھوا	*paRh-waa*	paRhnaa+V+root+caus2
42	پڑھوانا	*paRh-waa-naa*	paRhnaa+V+caus2+masc
43	پڑھوانی	*paRh-waa-nee*	paRhnaa+V+caus2+fem
44	پڑھوانے	*paRh-waa-ney*	paRhnaa+V+caus2+obl
45	پڑھواتیں	*paRh-waa-teeN*	paRhnaa+V+caus2+repeat+fem+pl
46	پڑھواتی	*paRh-waa-tee*	paRhnaa+V+caus2+repeat+fem+sg
47	پڑھواتے	*paRh-waa-tey*	paRhnaa+V+caus2+repeat+masc+pl
48	پڑھواتا	*paRh-waa-taa*	paRhnaa+V+caus2+repeat+masc+pl
49	پڑھوائیں	*paRh-waa-eeN*	paRhnaa+V+caus2+perf+fem+pl
50	پڑھوائی	*paRh-waa-ee*	paRhnaa+V+caus2+perf+fem+sg
51	پڑھوائے	*paRh-waa-ey*	paRhnaa+V+caus2+perf+masc+pl
52	پڑھوایا	*paRh-waa-yaa*	paRhnaa+V+caus2+perf+masc+sg
53	پڑھوائیں	*paRh-waa-eyN*	paRhnaa+V+caus2+subj+1st+pl paRhnaa+V+caus2+subj+2nd+polite paRhnaa+V+caus2+subj+3rd+pl
54	پڑھوں	*paRh-ooN*	paRhnaa+V+caus2+subj+1st+sg
55	پڑھوائے	*paRh-aa-ey*	paRhnaa+V+caus2+subj+3rd+sg
56	پڑھواو	*paRh-waa-ao*	paRhnaa+V+caus2+subj+2nd+formal
57	پڑھوایئے	*paRh-waa-eeay*	paRhnaa+V+caus2+subj+2nd+request
58	پڑھوائیں	*paRh-waa-eyN*	paRhnaa+V+caus2+impr+2nd+polite
59	پڑھواو	*paRh-waa-ao*	paRhnaa+V+caus2+impr+2nd+formal
60	پڑھوا	*paRh-waa*	paRhnaa+V+caus2+impr+2nd+frank

Figure 4.2 shows Acyclic Deterministic Finite State Automata (ADFSA) for few Urdu words having root forms: *hans*, *bol*, *paRh*, *deykh* and *xareed*. Each node

represents state and arrow represents transition. The hollow nodes represent intermediate states, while filled nodes represent final states. The characters starting with a dot represent grammatical information, while those starting with no dot represent normal characters.

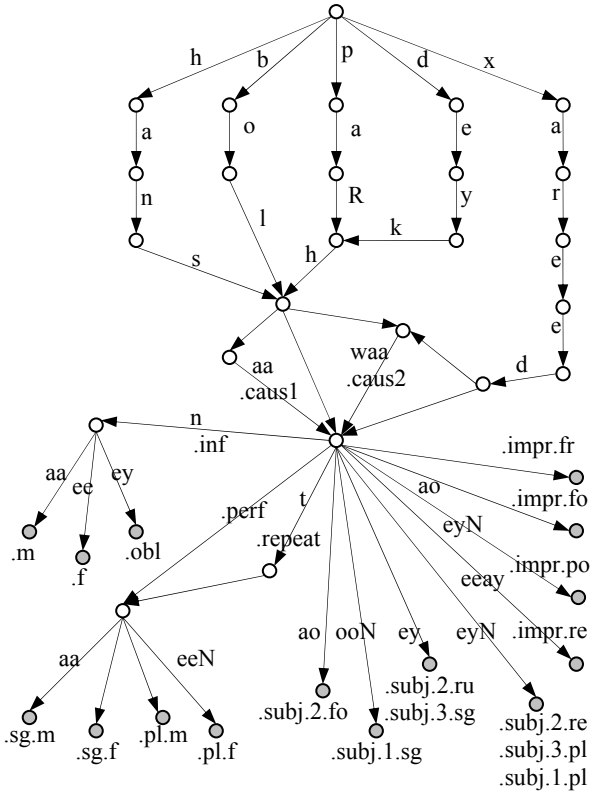


Figure 4.2: Acyclic Deterministic Finite State Automata Representing Various Morphological Forms of Few Urdu Verbs

4.5 Tense

Tense (زمانہ) tells about the location in time at which an event occurs or a state changes. It is mainly divided into three categories: present; past and future. It is a grammatical category which is either marked on the verb itself or it can be marked on the accompanying auxiliary or helping verbs. Tense refers to the time of the event or state denoted by the verb in relation to the time of utterance.

The tense can be represented in terms of Reichenbachian relations (Butt 2003). It defines three temporal points: the time of utterance/speech (S), the reference time (R), and the event time (E). These three points generate two relationships, one between S and R time (S/R), which is contextually determined relation, and another between R and E time (R/E), which is intrinsic relation. The temporal points of these relationship may occur simultaneous, as S and R are in the Present Tense, or may be ordered sequentially, as in the tenses with perfect aspect, E occurs before R (E < R),

regardless of the relationship (S/R). This allows for perfect aspect in the past, present and future tenses. Table 4.17 lists tenses in Reichenbachian concept relations.

Table 4.17: Tenses in Reichenbachian Concept Relations

Tense	Reichenbachian Relations
Present Tense	E ⇔ R and R ⇔ S
Present Perfect Tense	E < R and R ⇔ S
Past Tense	E ⇔ R and R < S
Past Perfect Tense	E < R and R < S
Future Tense	E ⇔ R and R > S
Future Perfect Tense	E < R and R > S

Tense in Urdu is represented by verb auxiliaries. Table 4.18 shows Urdu auxiliaries for present, past and future tenses.

Table 4.18: Auxiliaries for Representing Tense in Urdu

Auxiliary		Tense	Person	Gender	Number	Honor Form
hooN	ہوں	Present	1st	masc, fem	sg	–
hay	ہے	Present	2nd	masc, fem	–	frank
hao	ہو	Present	2nd	masc, fem	–	formal
hayN	ہیں	Present	2nd	masc, fem	–	polite
hay	ہے	Present	3rd	masc, fem	sg	–
hayN	ہیں	Present	1st, 3rd	masc, fem	pl	–
thaa	تھا	Past	1st, 3rd	masc	sg	–
thay	تھے	Past	1st, 3rd	masc	pl	–
thaa	تھا	Past	2nd	masc	–	frank
thay	تھے	Past	2nd	masc	–	formal, polite
thee	تھی	Past	1st, 3rd	fem	sg	–
theeN	تھیں	Past	1st, 3rd	fem	pl	–
thee	تھی	Past	2nd	fem	–	frank, formal
theeN	تھیں	Past	2nd	fem	–	polite
gaa	گا	Future	1st, 3rd	masc	sg	–
gay	گے	Future	1st, 3rd	masc	pl	–
gaa	گا	Future	2nd	masc	–	–
gay	گے	Future	2nd	masc	–	formal, polite
gee	گی	Future	1st, 3rd	fem	sg, pl	–
gee	گی	Future	2nd	fem	sg, pl	frank, formal, polite

The auxiliaries for tense have complex dependence on person, number and gender as shown in Table 4.18. The present auxiliaries are the same for masculine and feminine gender, in other words, these do not have dependency on gender. For second person, Urdu has honorific forms like frank (or rude), formal (or familiar) and polite (or respect). In the case of second person, most of the time the same auxiliary is used for singular or plural person, therefore the 'number' is not significant.

In negative present tense, sometimes, present auxiliary is dropped.

4.6 Aspect

The aspect expresses features about duration, repetition and/or completion of an event without reference to its actual location in time. The action of a verb is either complete, *tamaam* (تمام), termed as 'perfect', or it may be incomplete, *naa-tamaam* (نا تمام). The incomplete form is known as 'imperfect', 'progressive', or 'continuous' form, *jaaree* (جاری). Both verb inflections and auxiliaries are utilized in Urdu to describe aspect. It has been described that repetitive and perfective morphemes are directly marked on the verb, the Urdu auxiliaries also show aspect in other cases. Table 4.19 lists some Urdu aspect auxiliaries along with related features, these auxiliaries require subject agreement.

Table 4.19: Some Urdu Aspect Auxiliaries; Subject Agreement

Auxiliary		Aspect	Person	Gender	Num	Honor Form
chokaa	چکا	Perfect	1st, 3rd	masc	sg	–
chokee	چکی	Perfect	1st, 3rd	fem	sg, pl	–
chokey	چکے	Perfect	1st, 3rd	masc	pl	–
chokaa	چکا	Perfect	2nd	masc	–	frank
chokey	چکے	Perfect	2nd	masc	–	formal, polite
chokee	چکی	Perfect	2nd	fem	–	frank, formal, polite
rahaa	رہا	Progressive	1st, 3rd	masc	sg	
rahee	رہی	Progressive	1st, 3rd	fem	sg, pl	
rahey	رہے	Progressive	1st, 3rd	masc	pl	

Perfective aspect in Urdu can be expressed by either the use of perfective verb auxiliary and or by perfective verb-morpheme. There are other aspect auxiliaries in Urdu like *chalaa*, *jaa*, *rahaa*, *lagaa*, etc. that show duration and repetition related aspects. The following aspects will be discussed, with some syntactic details, in Chapter 8.

- Perfective aspect
- Progressive aspect
- Repetitive aspect
- Inceptive aspect

4.7 Mood

The verb mood describes the relationship of a verb with respect to purpose and actual happening. Languages mostly differentiate various moods by inflecting the verb form. The verb mood of a verb expresses a fact (indicative mood), a command (imperative mood), a question (interrogative mood), a wish (optative mood), or a conditionality (subjunctive mood). This aspect is shown by using modality. A modal auxiliary is used in English to show the mood. The following moods are commonly expressed in Urdu texts.

- Declarative mood
- Permissive mood
- Prohibitive mood
- Imperative mood
- Capacitive mood
- Suggestive mood
- Compulsive mood
- Dubitative/Presumptive mood
- Subjunctive mood

The subjunctive and imperative moods in Urdu have verb morpheme to represent mood. While other moods utilize separate auxiliaries to represent mood, like, *looN*, *saktaa*, *chaaheeey*, *hooN gaa*, *paRaa*, *deeaa*, etc. These moods will be discussed under syntax in more details along with syntactic examples in Chapter 8.

4.8 Attribute–Values for Urdu Verbs

In this Chapter, various Urdu verb forms and characteristics are described. The attributes and the values, which attributes can take to represent verb types and characteristics, are summarized in Table 4.20. These attribute-values are useful in describing Urdu verbs for the morphological and syntactical analysis.

Table 4.20: Attribute–Values for Urdu Verbs

Attribute	Values	Comments
VFORMSTM	root, causative1, causative2	verb stem forms
VFORM	infinitive, perfective, repetitive, imperative, subjunctive	verb forms
VFORMINF	absolute, oblique	infinitive verb forms
VFORMSUB	S1, S2, S3, S4	subjunctive verb forms
VFORMIMP	frank, formal, polite, request	imperative verb forms
GENDER	masculine, feminine	gender attribute
NUMBER	singular, plural	number attribute
PERSON	first, second, third	person attribute
TENSE	present, past, future	tense
ASPECT	perfect, repetitive, progressive, inceptive	aspect
MOOD	declarative, imperative, subjunctive, capacitive, presumptive, compulsive, permissive, prohibitive, suggestive	mood
VOICE	active, passive	voice
BASELANG	Arabic, Persian, Hindi, Turkish, English	base language

Urdu verb's lexical attributes and their respective values are associated with words, however, these are also syntactically important at a sentence level, because these are useful in various syntactic agreement requirements.

Chapter 5

URDU NOUN CHARACTERISTICS AND MORPHOLOGY

Noun (اسم) is a word, which is the name of something, i.e., name of a person, an animal, a place, a thing, a situation, a time, or a concept, etc.

Initial classification of nouns is into proper and common (improper) nouns. Proper noun (اسم خاص، معرفه) is the name of particular person, place or thing, like: Zafar, Lahore, Kohinoor, etc. Common noun (اسم عام، نکره) is the general name for any person, place or thing, like: boy, city, diamond, etc.

Attribute	Values	Comments
NCLASS	proper, common	Noun Classes

Common nouns are further classified with respect to concept they are representing, into state nouns, group nouns, spatial nouns, temporal nouns, instrumental nouns, etc.

Attribute	Values	Comments
NCONCEPT	state, group, spatial, temporal, instrument	Concepts represented by common nouns

Abstract Nouns or Status nouns (اسم کیفیت) – depict some state, characteristic or theme. These are usually used in declarative sentences telling some state or news about someone. *Group Nouns* – (اسم جمع) represent a group or collection of multiple nouns and look like that their number is plural, but their syntactic use in the sentence is singular. *Spatial Nouns* – (اسم ظرف مکاں) refer to location in space. *Temporal Nouns* – (اسم ظرف زماں) refer to location in time. *Instrumental Nouns* – (اسم آلہ) refer to instrument. Some examples of common noun categories are shown in Table 5.1.

Another classification of common nouns is mass and count nouns. Mass noun (اسم مادہ - اسم جنس) is the same for part or whole of something, e.g., a small amount of water is called water and similarly whole sea contains water. The mass nouns are not counted. Some examples of mass and count nouns are shown in Table 5.2.

Table 5.1: Few Urdu Common Nouns
(a) Abstract (b) Group (c) Spatial (d) Temporal (e) Instrumental

(a) Few Abstract Nouns in Urdu

دوستی	*daostee*, friendship	صحت	*sehat*, physical condition
لڑکپن	*laRkpan*, boyhood	جلن	*jalan*, soreness
نرمی	*narmee*, softness	چال چلن	*chaal chalan*, reputation
گرمی	*garmee*, hotness	گھبراہٹ	*ghabraahaT*, uneasiness
درد	*dard*, pain	دیوانہ پن	*deewaanah pan*, mental illness

(b) Few Group Nouns in Urdu

فوج	*faoj*, army	قطار	*qttaar*, queue
انجمن	*aanjoman*, society	جھنڈ	*jhonD*, cluster
میلہ	*meylah*, fair/ festival	گروپ	*garoop*, group

(c) Few Spatial Nouns in Urdu

گھر	*ghar*, home	ددھیال	*dadheyaal*, father's family
میدان	*meydaan*, play land	پن گھٹ	*pan ghaT*, a place to get water
سبزہ زار	*sabzazar*, a green field	بیٹھک	*beyThak*, sitting room
پرستان	*parestaan*, fairyland	تارگھر	*taar ghar*, telegraph office

(d) Few Temporal Nouns in Urdu

پانچ گھنٹے	*paanch ghanTey*, five hours	دو دِن	*dao din*, two days
صبح	*SobaH*, morning	رات	*raat*, night
کل	*kal*, tomorrow/ yesterday	شام	*shaam*, evening
پرسوں	*parsooN*, a day after tomorrow/ a day before yesterday	ڈیڑھ بجے	*DeyRh bajey*, half past one

(e) Few Instrument Nouns in Urdu

چاقو	*chaaqoo*, knife	جھاڑو	*jhaaRoo*, wisp/ swab/ mop
قلم	*qalam*, pen	نہرنا	*naharnaa*, nail cutter
کلہاڑی	*kolhaaRee*, axe	ہتھوڑا	*hathaoRaa*, hammer

Table 5.2: Few Mass and Count Nouns in Urdu

Mass Nouns		Count Nouns	
آٹا	*aaTaa*, flour	مکان	*makaan*, house
پانی	*paanee*, water	پلنگ	*palang*, bed
چینی	*cheenee*, sugar	گھڑی	*ghaRee*, watch
دال	*daal*, grams	باغ	*baaG*, garden

However, mass nouns may adopt plural and oblique forms, if we want to refer to number of different kinds of mass nouns. Like in (91), *daal* contains plural morpheme *–eyN* to refer to different kinds of grams (or pulses) used to make *haleem*, a special Asian dish.

(91) میں نے ساری دالیں ڈال کر حلیم پکائی ہے
meyN=ney saaree daal-eyN Daal kar haleem pakaaee hay
I=*erg* all gram-*pl* put having haleem.*sg.fem* cook.*perf.sg.fem*
I, having put all the grams, cooked *haleem*.

5.1 Urdu Noun Characteristics

The basic characteristics associated with Urdu nouns are: (1) gender; (2) number; (3) form; and (4) case, which are briefly discussed below.

5.1.1 Gender

Nouns in Urdu bear masculine and feminine gender (Mustafa 1973; Abdul-Haq 1991; Schmidt 1999). This gender is realistic for animate nouns, which have natural gender classification, but for inanimate nouns, this gender classification is unrealistic and artificial, because they do not have natural gender. This tradition of assigning gender to inanimate nouns has come in Urdu from its ancestors languages. Gender of such nouns in some languages is neutral, which is a realistic classification. Some gender classification for Urdu nouns is shown in Table 5.3.

Table 5.3: Gender for Some Urdu Nouns

مذکر	*moZakar*, Masculine	مونث	*maoanas*, Feminine
ناول	*naawel*, novel	کتاب	*ketaab*, book
قلم	*qalam*, pen	پنسل	*pensel*, pencil
مکان	*makaan*, house	دوکان	*daokaan*, shop
آدمی	*aadmee*, man	عورت	*Aaorat*, woman

There is no general rule in Urdu to find the gender classification for inanimate nouns. Usually huge, heavy, powerful, dominant and bigger things are masculine, while smaller, weak and lighter are feminine. Normally, 'bigger nouns' (اسم مکبر) are masculine, while 'smaller nouns' (اسم مصغر) are feminine as shown in Table 5.4.

Table 5.4: Few Smaller and Bigger Nouns

Bigger Noun (اسم مکبر)		Smaller Noun (اسم مصغر)	
مذکر	*moZakar*, Masculine	مونث	*maoanas*, Feminine
رسا	*ras-aa*, thick rope	رسی	*ras-ee*, thin rope
گولا	*gaol-aa*, big spherical subject	گولی	*gaol-ee*, small spherical thing
جال	*jaal*, net	جالی	*jaal-ee*, small net
پگ	*pag*, big special cap	پگڑی	*pagR-ee*, small special cap
پگڑ	*paggaR*, big special cap	پگڑی	*pagR-ee*, small special cap
گھڑیال	*ghaR-ee-aal*, clock	گھڑی	*ghaR-ee*, watch
دیگچہ	*deygch-aa*, big pan	دیگچی	*deygch-ee*, small pan

On the basis of morphology, it is very difficult to make rules to distinguish gender that encompass all combinations; however there are some general rules for nouns in Urdu that have special gender morpheme attached as suffix. For those nouns that do not have any morpheme marking for gender, the gender must be acquired from the dictionaries. Hindi based nouns ending with suffixes *–aa*, *–ah* are generally singular masculine, while those ending with *–ey*, are generally plural masculine.

However, Arabic based nouns, ending with suffix *–h*, are mostly singular feminine. Nouns ending with Persian suffixes *–pan*, *–pa* are masculine. Table 5.5 shows examples of nouns with masculine gender suffixes.

Table 5.5: Nouns with Masculine Gender Suffixes

لڑکا	*laRk-aa*, boy
مرغا	*morG-aa*, rooster
قرضہ	*qarJ-ah*, loan
روپیہ	*raopey-ah*, rupee
بڑھاپا	*boRhaap-aa*, old age

لڑکے	*laRk-ey*, boys
بکرے	*bakr-ey*, (male) goats
روپے	*raop-ey*, rupees
بچپن	*bach-pan*, childhood

Similarly, the (Hindi based) nouns ending in suffixes *–ee*, *–eeaa* are generally singular feminine and the nouns ending in suffixes *–eeaN*, *–eyN* are generally plural feminine. Arabic-based nouns adopted in Urdu ending with suffixes *–at*, *–aa* are feminine. Persian based nouns adopted in Urdu ending with suffixes *–gaah*, *–ee*, *–gee*, *–haT*, *–aawaT* are feminine. Various examples of feminine nouns with the above-mentioned endings are shown in Table 5.6.

Table 5.6: Nouns with Feminine Gender Suffixes

لڑکی	*laRk-ee*, girl
مرغی	*morG-ee*, hen
دنیا	*don-eeaa*, world
چڑیا	*ceR-eeaa*, sparrow
عبادت گاہ	*Aebaadat-gaah*, place of worship
زندگی	*zenda-gee*, life
روکاوٹ	*raok-aawaT*, obstacle

لڑکیاں	*laRk-eeaN*, girls
مرغیاں	*morG-eeaN*, hens
کتابیں	*ketaab-eeN*, books
چڑیاں	*ceR-eeaN*, sparrows
دوستی	*daost-ee*, friendship
گھبراہٹ	*ghabraa-haT*, discomfort
مسکراہٹ	*moskraa-haT*, smile

5.1.2 Number

Urdu nouns like English have two dimensions of number: singular and plural. Unlike Arabic or Sanskrit, it has no category for ‘dual’ nouns.

5.1.3 Form

Normal form in which Urdu nouns are listed in dictionaries is known as ‘nominative’ form. Urdu nouns appear in ‘oblique’ form if they are followed by a postposition. The nouns that refer to humans, and sometimes other animate nouns, have another form used to call or address person(s), this form is called ‘vocative’. Table 5.7 lists some noun forms. In the literature (Mohanan 1990; Arsenault 2002), such noun forms are sometimes referred to as a type of ‘case’ but these morphological forms does not represents a case by themselves. Another form is more commonly called the ‘case’ which attach with nouns at syntactic level briefly discussed in next section, and with more details it is discussed in section 7.3 . Therefore, to distinguish two case types in Urdu this work will refer to first type as noun ‘form’.

Table 5.7: Noun Forms for Few Urdu Words

Nominative (NOM) قائم		Oblique (OBL) محرف		Vocative (VOC) ندائی	
لڑکا	*laRkaa*, boy	لڑکے	*laRkey*, boy	لڑکے	*laRkey*, boy
لڑکی	*laRkee*, girl	لڑکی	*laRkee*, girl	لڑکی	*laRkee*, girl
لڑکے	*laRkey*, boys	لڑکوں	*laRkaoN*, boys	لڑکو	*laRkao*, boys
لڑکیاں	*laRkeeaN*, girls	لڑکیوں	*laRkeeoN*, girls	لڑکیو	*laRkeeo*, girls
مرغا	*morGaa*, rooster	مرغے	*morGey*, rooster	لوگو	*laogao*, people
کمرہ	*kamrah*, room	کمرے	*kamrey*, room	بچو	*bach.chao*, children

5.1.4 Case

The case markers that follow nouns in the form of post positions cannot be handled at lexical level through morphological suffixes and are thus needed to be handled at syntactic level (Butt and King 2002). Table 5.8 lists case markers in Urdu along with example sentences.

Table 5.8: Case Markers in Urdu

Case	Case Marker		Example Sentence
Ergative (agent/subject)	*ney*	نِے	لڑکے نے کتاب خریدی *laRkey ney ketaab xareedee* The boy bought a book.
Dative (indirect object)	*kao*	کَو	میں نے لڑکے کو کتاب دی *mayN ney laRkey kao ketaab dee* I gave the book to the boy.
Accusative (direct object)	*kao*	کَو	لڑکے نے کتاب کو خریدا *laRkey ney ketaab kao xareedaa* The boy bought the book.
Instrumental	*sey*	سِے	لڑکے نے پنسل سے لکھا *laRkey ney pensel sey lekhaa* The boy wrote with the pencil
Ablative (agent in passive)	*sey*	سِے	لڑکے سے خط لکھا گیا *laRkey sey xat lekhaa gayaa* The letter is written by the boy
Locative	*meyN*	مِیں	لڑکا کمرے میں ہے *laRkaa kamrey meyN hay* The boy is in the room.
Locative	*par*	پَر	کتاب میز پر ہے *ketaab meyz par hay* The book is on the table.

If there is no case marker, then the case is ‘nominative’.

Nominative		لڑکا کتاب خریدے گا *laRkaa ketaab xareedey gaa* The boy will buy a book

5.2 Noun Morphology

This work divides Urdu nouns into five categories based on difference in morphemes and associated syntactic information.

The category 1 nouns are animate nouns that end with morpheme *–aa* or *–ah*. More specifically the morphology of this category is applicable in daily life usage to those animate nouns that are used for humans but sometimes this morphology is also used with other animate nouns in a narration or a story. In this category although there are eight morphemes, i.e., *–aa* (or *–ah*), *–ey*, *–ooN*, *–ao*, *–ee*, *–ee–aN*, *–ee–ooN*, *–ee–ao* but total noun forms are ten based on different tags as shown in Table 5.9 (a).

Table 5.9: Noun Morphology in Urdu

(a) Category 1 Noun Morphology in Urdu

	Morphological Tags	لڑکا boy		بچہ child		بکرا masc. goat	
1	+masc+sg	لڑکا	*lark-aa*	بچہ	*bach.ch-ah*	بکرا	*bakr-aa*
2	+masc+sg+obl	لڑکے	*lark-ey*	بچے	*bach.ch-ey*	بکرے	*bakr-ey*
3	+masc+pl	لڑکے	*lark-ey*	بچے	*bach.ch-ey*	بکرے	*bakr-ey*
4	+masc+pl+obl	لڑکوں	*lark-ooN*	بچوں	*bach.ch-ooN*	بکروں	*bakr-ooN*
5	+masc+pl+voc	لڑکو	*lark-ao*	بچو	*bach.ch-ao*	بکرو	*bakr-ao*
6	+fem+sg	لڑکی	*lark-ee*	بچی	*bach.ch-ee*	بکری	*bakr-ee*
7	+fem+sg+obl	لڑکی	*lark-ee*	بچی	*bach.ch-ee*	بکری	*bakr-ee*
8	+fem+pl	لڑکیاں	*lark-ee-aN*	بچیاں	*bach.ch-ee-aN*	بکریاں	*bakr-ee-aN*
9	+fem+pl+obl	لڑکیوں	*lark-ee-ooN*	بچیوں	*bach.ch-ee-ooN*	بکریوں	*bakr-ee-ooN*
10	+fem+pl+voc	لڑکیو	*lark-ee-ao*	بچیو	*bach.ch-ee-ao*	بکریو	*bakr-ee-ao*

(b) Category 2 Noun Morphology in Urdu

	Morphological Tags	Mango	Novel	Letter	Plane	Question
1	+masc+sg	آم *aam*	ناول *naawel*	خط *xatt*	جہاز *jahaaz*	سوال *jahaaz*
2	+masc+sg+obl					
3	+masc+pl					
4	+masc+pl+obl	آموں *aam-ooN*	ناولوں *naawel-ooN*	خطوں *xatt-ooN*	جہازوں *jahaaz-ooN*	سوالوں *jahaaz-ooN*

(c) Category 3 Noun Morphology in Urdu

	Morphological Tags	Book	Table	Talk	Road	Socks
1	+fem+sg	کتاب *ketaab*	میز *meyz*	بات *baat*	سڑک *baat*	جراب *joraab*
2	+fem+sg+obl					
3	+fem+pl	کتابیں *ketaab-eyN*	میزیں *meyz-eyN*	باتیں *baat-eyN*	سڑکیں *baat-eyN*	جرابیں *joraab-eyN*
4	+fem+pl+obl	کتابوں	میزوں	باتوں	سڑکوں	جرابوں

(d) Category 4 Noun Morphology in Urdu

	Morphological Tags	Lock	Food	Door	Room	Birdcage
1	+masc+sg	تالا *taal-aa*	کھانا *khaan-aa*	دروازہ *darwaaz-aa*	کمرہ *kamar-aa*	پنجرہ *penjr-aa*
2	+masc+sg+obl	تالے *taal-ey*	کھانے *khaan-ey*	دروازے *darwaaz-ey*	کمرے *kamar-ey*	پنجرے *penjr-ey*
3	+masc+pl					
4	+masc+pl+obl	تالوں	کھانوں	دروازوں	کمروں	پنجروں

(e) Category 5 Noun Morphology in Urdu

	Morphological Tags	Chair	Staircase	Key	Car	Bread
1	+fem+sg	کرسی *kors-ee*	سیڑھی *seeRh-ee*	چابی *chaab-ee*	گاڑی *gaaR-ee*	روٹی *raot-ee*
2	+fem+sg+obl					
3	+fem+pl	کرسیاں *kors-eeaN*	سیڑھیاں *seeRh-eeaN*	چابیاں *chaab-eeaN*	گاڑیاں *gaaR-eeaN*	روٹیاں *raot-eeaN*
4	+fem+pl+obl	کرسیوں	سیڑھیوں	چابیوں	گاڑیوں	روٹیوں

The category 2 nouns are inanimate masculine nouns that do not end in masculine gender morpheme *–aa* or *–ah*. The singular, plural and singular-oblique forms of these nouns are the same, but their plural-oblique form has morpheme *–ooN*. Table 5.9 (b) lists some category 2 nouns along with gender, number and obliqueness information tags.

The category 3 nouns are inanimate feminine nouns that do not end in feminine gender morpheme. The singular and singular-oblique forms of these nouns are the same, the plural form has morpheme *–eyN*, their plural-oblique form has morpheme *–ooN*. Table 5.9 (c) lists some category 3 nouns along with gender, number and obliqueness information tags.

The category 4 nouns are inanimate masculine nouns that end in masculine gender morpheme *–aa* or *–ah*. Their singular form has morpheme *–aa* or *–ah*, their singular-oblique and plural forms have morpheme *–ey* and their plural-oblique form has morpheme *–ooN*. Table 5.9 (d) lists some category 4 nouns along with gender, number and obliqueness information tags.

The category 5 nouns are inanimate feminine nouns that end in feminine gender morpheme *–ee*. Their singular and singular-oblique forms have morpheme *–ee*, the plural forms have morpheme *–ee–aN* and the plural-oblique form has morpheme *–ee–ooN*. Table 5.9 (e) lists some category 5 nouns along with gender, number and obliqueness information tags.

5.3 Adjective Morphology

Adjectives in Urdu come before noun to which they modify and these are required to agree with noun form in gender, number and obliqueness if they have morpheme to represent these features. If adjectives do not have morpheme to identify gender, number and obliqueness features then theoretically they do not require to agree, but in practice to handle both categories of adjectives in a uniform manner this work assumes that these have these features which are not morphologically visible. Therefore, this work divides Urdu adjectives into two categories: one having morphology for agreement with noun as shown in Table 5.10 (a) and the other has no morphology to agree with nouns as shown in Table 5.10 (b). However, both adjective categories have gender, number and obliqueness features to satisfy noun-adjective agreement equations.

Table 5.10: Adjective Morphology in Urdu

(a) Category 1 Adjective Morphology in Urdu

	Morphological Tags	اچھا good	نیلا blue	ہرا green	تازہ fresh	تیسرا third	کڑوا harsh
1	+masc+sg	اچھا *ach.ch-aa*	نیلا *neel-aa*	ہرا *har-aa*	تازہ *taaz-ah*	تیسرا *teesr-aa*	کڑوا *kaRw-aa*
2	+masc+sg+obl	اچھے *ach.ch-ey*	نیلے *neel-ey*	ہرے *har-ey*	تازے *taaz-ey*	تیسرے *teesr-ey*	کڑوے *kaRw-ey*
3	+masc+pl						
4	+masc+pl+obl						
5	+fem+sg	اچھی *ach.ch-ee*	نیلی *neel-ee*	ہری *har-ee*	تازی *taaz-ee*	تیسری *teesr-ee*	کڑوی *kaRw-ee*
6	+fem+sg+obl						
7	+fem+pl						
8	+fem+pl+obl						

(b) Category 2 Adjective Morphology in Urdu

	Morphological Tags	گول round	سرخ red	لال red	باسی old	شریر naughty	محنتی hard worker
1	+masc+sg	گول *gaol*	سرخ *sorkh*	لال *laal*	باسی *baasee*	شریر *shareer*	محنتی *meHnatee*
2	+masc+sg+obl						
3	+masc+pl						
4	+masc+pl+obl						
5	+fem+sg						
6	+fem+sg+obl						
7	+fem+pl						
8	+fem+pl+obl						

5.4 Attribute–Value Tags for Urdu Nouns

In this Chapter, noun types and characteristics related to Urdu nouns has been reviewed. The summary of attributes and various values, which attributes can take,

are listed in Table 5.11. These attributes and associated values may be helpful for the morphological and syntactical analysis for Urdu nouns.

Table 5.11: Attribute–Values for Urdu Nouns

Attribute	Values	Comments
N-CLASS	common, proper	noun class
N-CONCEPT	abstract, group, spatial, temporal, instrumental, animate	noun semantic concept
N-TYPE	mass, count	noun type
GENDER	masculine, feminine	noun gender
N-FORM	nominative, oblique, vocative	noun form
NUMBER	singular, plural	noun number
CASE	nominative, ergative, dative, accusative, instrumental, locative, travel, infinitive, participant, temporal	noun case
PERSON	first, second, third	person
BASELANG	Arabic, Persian, Hindi, Turkish, English	base language

Urdu noun's attributes and their respective values are lexical in nature as these are associated with words, however, these are also syntactically important, because these are useful in various syntactic agreement requirements.

Chapter 6

ALGORITHMS FOR LEXICON IMPLEMENTATION

6.1 Introduction

This chapter reviews the various algorithms and methods for efficient storage and retrieval of lexicon. The chapter has been organized into two main parts: In the first part lexicon is implemented and tested using hash tables with least consideration of morphology and therefore all word forms are stored separately in the hash table. The hash table storage is efficient to access but requires more space in memory. In second part lexical transducers, which are specialized finite state automata, are considered for the storage of Urdu lexicon. These are efficient both in time and space but require morphological analysis of language data.

6.2 Storage of Urdu Lexicon

Lexicon is the base for many natural language processing applications. Researchers and developers of machine translation (MT) systems are also concerned with the efficient storage and retrieval of the lexical information. This is especially critical when a small MT system is developed to a full-scale MT system in order to process real world texts that need larger and richer lexicons containing large subject domains.

This section reviews the approaches for the Urdu lexicon implementation. The easiest workable method for the storage and retrieval of a word list could be to list the lexical information in a text file. However, many NLP applications like machine translation, spelling checker, speech synthesizer, etc. require continuous consultation with the lexicon for each word. Raw lexicon as a simple word list is expensive both for the search time and storage space. The lexicon's word lookup algorithm must be highly efficient. The unsorted list of words has the worst-case time efficiency of the order of O(N), where N is the number of words stored in the lexicon. This is certainly not acceptable, when the value of N is high. The search efficiency of word list can be improved to $O(\log_2 N)$, e.g., using a sorted list and applying binary search or converting the word list to a binary search tree (Cormen, Leiserson et al. 1994; Knuth 1998). Although the increase in search efficiency to $O(\log_2 N)$ is significant, more improvement is required for the NLP applications under consideration.

High word lookup efficiency of the order of O(1), close to perfect hashing, can be achieved using hash tables with appropriate hash functions. Hashing results in a simpler and acceptable lexicon design at the cost of some extra space. A compact representation for lexicon can be a character tree structure, called a trie. Lexicon storage using trie reduces word search time as well as storage space as compared to simple word list. Further enhancement in efficiency is achieved by converting the trie into directed acyclic word graph (Ciura and Deorowicz 2001). The directed acyclic word graph could be utilized for automatic separation of word stems from prefixes. Further compressed form of which is a directed acyclic word graph (DAWG). The search time efficiency for DAWG is O(L), where L is the average length of words in lexicon. Simple DAWG can be used with spell checking application (Ciura and Deorowicz 2001), but for MT application morphology information is also required. For MT application a specialized form of DAWG called lexical transducer is a better choice (Beesley and Karttunen 2003). Lexical transducer is a form of DAWG, which maps input surface word form to lexical word form and vice versa. Urdu language morphology rules are inherited from many languages, like Sanskrit, Arabic, Persian, etc. which make full morphology-based design of the Urdu lexicon containing inflected as well as derivative words is a difficult task. A comparative study shows that lexical transducer implementation, due to morphological analysis requirement, is relatively more complex than hashing but it is efficient for both search time and storage space requirements.

6.3 Storage in a Hash Table

Hashing is one of the solutions for a large dictionary problem. Although, using hash tables, the retrieval of data is very fast – in one or few steps, but the main problem with hashing is that all strings with full-length are needed to be stored, which requires more memory space. The perfect hash function is the one in which no collisions occur. It means no repeating hash values should arise for different words. It is difficult to find such a perfect hash function, however, a function that is close to perfect hashing can be found. Hash functions are classified by the way they generate hash values from data. In addition-method, the hash value is computed by traversing through each character of the word and continually incrementing an initial hash value. The calculation done on the element value is usually in the form of a multiplication by a prime number. In bitwise-shift hashing, similar to addition-method every character of the word in the data string is used to construct the hash, but the value is calculated through bitwise left and right shifting, the shift value is normally a prime number. Some string hashing functions have been implemented and tested for the English word list and for the Unicode-based Urdu word list, where dictionary sizes varying from 17,000 words to 75,000 words are used with details shown in Table 6.1. The

results listed in Table 6.2 show that we can achieve high search efficiency close to perfect-hashing requirements. A basic algorithm to calculate hash value is as follows, other algorithms are modification to this basic algorithm:

1: Initialize hash value
2: For each character in a word, do step 3, 4
3: Multiply character value with some number P_i
4: Add result of step 3 to hash value
5: Normalize hash value to fit to size of hash table

Some of the hash functions are listed here which are used for hashing Urdu lexical entries. The simple hash function based on addition method is:

```
hash = 0;
for(i=0; i<word.Length; i++)
  hash = hash*29 + (word[i]-'A');
hashIndex = hash % hashSize;
```

Another hash function known as RS hash function discussed by Robert Sedgwick (Sedgwick 1988) based on addition method is:

```
b = 378551;  a = 63689;  hash = 0;
for(i=0; i<word.Length; i++) {
  hash = hash*a+Word[i];
  a = a*b;
}
hashIndex = hash & 0x7FFFFFF;
```

We present some of the rotative hash functions. The JS hash function developed by Justin Sobel (Partow) based on bitwise shifting is:

```
hash = 1315423911;
for(int i = 0; i <word.Length; i++) {
hash ^= ((hash << 5) + Word[i] ;
hash += (hash >> 2));
}
hashIndex = (hash & 0x7FFFFF);
```

The Daniel J. Bernstein gave DJB hash function (http://cr.yp.to/papers.html) based on bitwise shifting is:

```
hash = 5381;
for(int i=0; i < word.Length; i++)
hash = ((hash << 5) + hash) + Word[i];
hashIndex = (int) (hash & 0x7FFFFFF);
```

AP Hash function developed by Arash Partow (Partow) based on bitwise shifting is:

```
hash = 0;
for(int i=0; i < word.Length; i++)
if ((i & 1) == 0)
  hash^=((hash<<7)^Word[i]^(hash>>3));
else
  hash^=(~((hash<<11)^Word[i]^(hash>>5)));
```

Table 6.1: Dimensions of Lexicon Files for Hash Table Storage

	English 1	English 2	Urdu 1	Urdu 2
Words	74317	25017	17476	49427
Hash Table Size	131071	32749	32749	65521
Average Word Length	8.5	7.2	13.4	10.9

As shown in Table 6.1, two Unicode based text files containing Urdu word list and two text files containing English word list are used for testing the hashing functions. Table 6.1 shows number of words in each file, hash table (HT) size, and average word length in each file.

There are other hashing functions that are not included in this study as the focus of the study was comparison of hashing as an alternate approach for lexicon implementation. For choosing a hash table size, largest prime number smaller than 2^{m+1} was used with a condition that $2^{m-1} < N < 2^m$, where N is the total number of words and m is an integer exponent.

Table 6.2: Average Word Lookup Searches in a Hash Table

	Dictionary/ Word List			
Hash Function	English 1	English 2	Urdu 1	Urdu 2
Simple	1.7	5.5	1.6	2.8
RS	1.7	2.8	1.6	2.6
JS	1.7	3.3	1.6	2.7
ELF	6.4	4.8	1.6	3.3
DJB	1.7	2.8	1.6	3.0
AP	1.7	3.0	1.6	2.6

The results are shown in Table 6.2 for the average number of word lookup searches required to find a word in a word list, which is implemented as a hash table. The results show that there are not many collisions and values of average access time per word are close to the perfect hashing value of one. This average word search time is calculated by accessing, one by one, all the words in the dictionary file. The linear open addressing method is used for collision resolution in this study.

6.4 Storage using Lexical Transducer

Lexical transducers are a better solution for representing lexicon of a language, especially for those languages that are known to have more inflectional morphology. To build a lexical transducer in order to store lexicon a through morphological analysis of the language is needed. The purpose of morphological analysis is to define

the word stems, prefixes and suffixes as well as a set of rules – known as morphotactics. The morphotactics tell how to combine the roots, stems, prefixes and suffixes with each other to make meaningful words. For example, there are two words to represent Muslim: the '*moslem*' (مسلم) and '*mosalmaan*' (مسلمان). To make antonym non-Muslim we can use prefix '*Gayr*' (غير) with '*moslem*' to make '*Gayr moslem*' (غير مسلم), but we cannot use it to make '*Gayr mosalmaan*' (غير مسلمان). These rules that govern which affix can be joined with which stem are known as morphotactics.

The following subsections cover some basic definitions and simpler data structures used to introduce lexical transducers and to define a method for automatically separating stems from affixes.

6.4.1 Trie – Tree Structure

A trie, or a character tree, is one of the solutions to store a lexicon with less storage space requirement as compared to string storage using a linear list, binary search tree or hash tables. A trie is a tree for storing strings in which there is one node for every common prefix. The strings are stored in extra leaf nodes.

Definition: For a given set of strings $S = \{A_1, A_2, \ldots, A_N\}$, where each string A_i contains characters from the given set of alphabets $A = \{ a, b, c, \ldots, z \}$; the trie for the given set S is defined recursively as:

$$\text{Trie}(S) = \{\text{Trie}(S\backslash\alpha_1),\ \text{Trie}(S\backslash\alpha_2),\ \cdots,\ \text{Trie}(S\backslash\alpha_r)\}$$

where $S\backslash\alpha_j$ means the subset of S consisting of strings that start with α_j, stripped of their initial letter α_j; recursion is halted when S is empty resulting in an empty trie.

6.4.2 Finite State Automata

Definition: A deterministic finite state automata is a 5–tuple: $D = \langle \Sigma, S, s, F, \mu \rangle$, where:

Σ:	is a finite set of alphabets
S:	is a finite set of states
$s \in S$:	is the starting state
$F \subseteq S$:	is the set of final states
$\mu(r, a)$:	$S \times \Sigma \rightarrow S$ is a transition function, where $r \in S$, $a \in \Sigma$. If we consider $\sigma \in \Sigma^*$, then μ is extended over $S \times \Sigma^*$ using induction: $\mu(r, \varepsilon) = r$, and $\mu(r, \sigma a) = \mu(\mu(r, \sigma), a)$, in case $\mu(r, \sigma)$ and $\mu(r, a)$ are defined, otherwise $\mu(r, \sigma a)$ is undefined

A finite state machine with at most one transition for each symbol and state combination is a deterministic finite state automaton (DFSA).

Definition: An automaton is acyclic, when for $\forall r \in S$ and $\forall \sigma \in \Sigma^{+}$, there is $\mu(r, \sigma) \neq r$.

The language of acyclic finite state automata is finite. By merging all equivalent sub-tries of a full trie into one, we can get acyclic DFSA.

A trie can be compressed to a minimal acyclic finite-state automata, which is also known as directed acyclic word graph (DWAG) by using algorithms (Daciuk 1998; Ciura and Deorowicz 2001). A directed acyclic graph represents the suffixes of a given string in which each edge is labeled with a character. The characters along a path from the root to a node make the substring, which the node is representing.

Definition: The deterministic finite state automata $D = \langle \Sigma, S, s, F, \mu \rangle$ is called minimal, for a given language $L(D) \subseteq \Sigma^{*}$, when for every other deterministic finite state automata $D' = \langle \Sigma, S', s', F', \mu' \rangle$ having language $L(D') = L(D)$, there exists the inequality $|S| \leq |S'|$, where $|S|$ represents number of states. This means that a minimal DFSA has minimum number of states for the given language. For a non-empty language, it is minimal if and only if every state is reachable from the starting state, from every state a final state is reachable, and there are no different equivalent states. There always exists a unique minimal automation for a given language (Daciuk 1998).

6.4.3 Implementation of Word Insertion

The algorithm for the word insertion in a trie, presented in Appendix B, is implemented and tested for English and Urdu word lists. Insertion of the word list (92) into a trie:

(92) بکرا، بکری، بکریاں، بکرے، لڑکا، لڑکی، لڑکیاں، لڑکے، لڑکیو

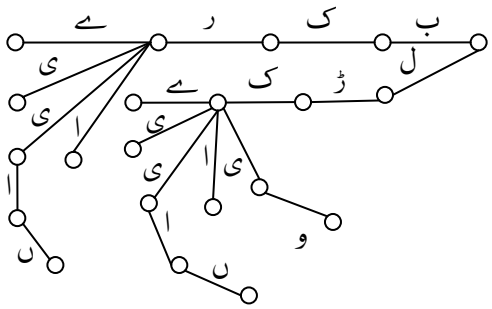


Figure 6.1: A Trie for Representing Urdu Words

In the trie shown in Figure 6.1, each path from root to a leaf represents a single word and the branching in tree represents successive characters. Trie is certainly an improvement over plain string storage, but it can be observed that paths compression is possible by representing common suffixes as one path instead of many paths with the same suffix. Result, then, of course will be a graph instead of a tree. The terms

states and transitions from automata theory will be used instead of nodes and branches from the graph theory. In Figure 6.1, four paths [ا – ی – ے – ں ی ا] can be compressed to one with one final state denoted as ◎, instead of a simple circle ○, which represents an ordinary state. This simple form of DFSA has one start state and one final state.

One class of algorithms for acyclic DFSA construction is by minimization of the trie (Daciuk 1998; Ciura and Deorowicz 2001) and another class of algorithms is for directly building acyclic DFSA from the given set of strings (Mihov; Daciuk 1998). Inserting the same word list (92) results in an acyclic DFSA, which is shown in Figure 6.2. We can see that the automaton created using above algorithm is both deterministic and acyclic. It has a single start state (with no in-coming transition) and a single final state (with no out-going transition).

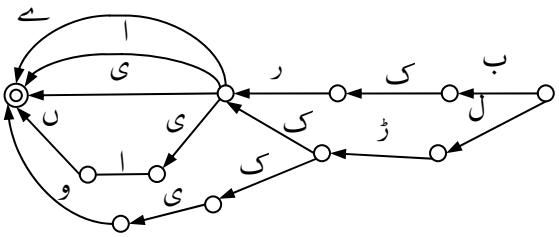


Figure 6.2: An acyclic DFSA for Urdu Words

The algorithm to construct the DFSA with no duplicate state is implemented. The resulting FSA is minimal, acyclic and deterministic. Although there are some algorithms available for unsorted words (Daciuk 1998), but the algorithm implemented was for sorted words, which is given in Appendix B. For minimal acyclic DFSA, we could have more than one final state. Therefore states are divided into two classes: First, the terminal final state (TFS), having no out-going transition. There is only one TFS in an automaton. Second, the intermediate final state (IFS) that can have out-going transitions as well as in-coming transitions. There can be many IFS in an automaton. So each state is stored with a flag that tell whether it is a final state or not. The minimal acyclic DFSA is shown in Figure 6.3.

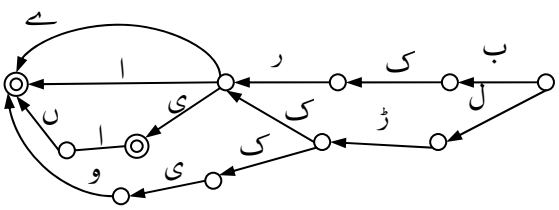


Figure 6.3: A Minimal Acyclic DFSA for Urdu Words

6.4.4 Affix Recognition by minimal acyclic DFSA

Morphological analysis of a language could be performed automatically if we could find morphological roots, stems, prefixes and suffixes for a given language. In this section, it is demonstrated that minimal acyclic DFSA can be used for automatic stemming application. Given the word list of a language containing all morphological forms, we can find prefixes and suffixes strings starting from intermediate final state (IFS) to the terminal final state (TFS). The IFS with many incoming branches shows that this final state is shared by many words, and therefore it is a good candidate for the identification of an affix. The following algorithm is proposed for finding suffixes from a list of words having all word forms.

1: create minimal acyclic DFSA from the list of sorted words.
2: for each state in the DFSA
3: if an IFS state has many incoming branches
4: mark it as Suffix State
5: find suffix string from Suffix state to TFS

The parts of words from start state to suffix state are candidates for being stems. For finding prefixes, the same algorithm is used, but first all strings are reversed and finally each suffix found is also reversed, which results in the required prefix.

Implementation and testing of algorithm showed that although correct identification of affixes and stems is carried out but there is also noticeable false detection and therefore more work on the algorithm may be made to reduce or remove false detection. Only those prefixes and suffixes are to be retained for inclusion to lexical transducer that has moderate frequency in the list of words and at the same time, they have morphological significance.

6.5 Lexical Transducers

A lexical transducer is a specialized finite state automaton that maps inflected surface forms to lexical forms and vice versa (Karttunen; Karttunen 1994; Beesley and Karttunen 2003). A surface form is the form of word that appears in a sentence, while lexical form is that form which is stored in a morphology-based lexicon.

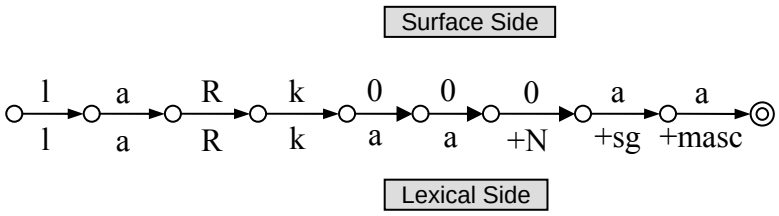


Figure 6.4: A Path in a Lexical Transducer for Urdu Noun '*laRkaa*'

If the input "l a R k 0 0 0 a a" to the transducer, shown in Figure 6.4, is given from surface side, where 0 represents a null value, the output of the transducer from the lexical side will be "l a R k a a +N +sg +masc" from the lexical side and vice versa. This could be written in the form of equation as shown in (93).

(93) *laRkaa*+N+sg+masc:*laRkaa*

Nouns, adjectives and verbs are morphologically open classes of words, some morphological properties of which have been discussed in previous chapters.

6.6 Conclusions

In this Chapter, few algorithms for lexicon implementation are discussed. The simple word list, due to large search time, is definitely not an acceptable solution for real MT applications. Table 6.2 shows the time efficiency for hash table implementation of a lexicon. The average number of word lookup searches for Urdu words ranges from 1.6 to 3.3 for different hash functions, which is acceptable and can be further improved by using better collision resolution strategy than linear open addressing. The advantages of hash table lexicon implementation are the fast access time, lesser morphological knowledge requirement and easier inclusion of non-morphological word attributes. The only disadvantage is more space requirements, which is not a big issue for current desktop computing standards.

Trie and directed acyclic word graphs have search time proportional to the length of words and have lesser space requirements. Table 6.2 shows that average Urdu word length is less than 15 characters, therefore for successful word search we need about 15 comparisons, while the unsuccessful search in these branching structures is even faster. These structures are useful for spell checking and automatic stemming applications. Lexical transducer based lexicon implementation is best suited for both search time and storage space requirements. However, the knowledge of stems and affixes as well as morphotactics must be available for the lexical transducer implementation through morphological analysis.

PART III

SYNTACTICAL ANALYSIS AND MODELING

Chapter 7

MODELING URDU NOMINAL SYNTAX BY IDENTIFYING CASE MARKERS AND POSTPOSITIONS

In Chapter 3 and Chapter 4, morphological analysis of various verb forms, noun forms and adjective forms in Urdu, and various attributes associated with different morphemes have been analyzed and listed. These lexical attributes obtained through morphology are very useful for the syntactic analysis based on the 'Lexical Functional Grammar'. In the approach used in this research, morphology variations are handled by using finite state transducers (Karttunen 1994; Beesley and Karttunen 2003). Given the various word forms, the finite state transducers, extract useful grammatical information from the word morphemes. In LFG, these lexical attributes extracted by finite state transducers become feature-value pairs at the feature-structure level. To assign syntactic attributes values extracted by finite state transducers, a form of mapping table is used. For example, GEND attribute may get values MASC or FEM, if the finite state tags have value +Masc or +Fem for the word under consideration. When constituent-structure nodes unify, these attributes at leaf node, which contain attributes obtained from lexical entries, get unify to generate overall f-structure.

In this Chapter, the NP structure is analyzed and its syntactic combination with various case-markers/ postpositions in Urdu is distinguished. A Noun Phrase (NP) in Urdu is characterized by a rich case-marking system, which makes possible its free phrase order. The case markers and postposition are similar in nature and it is not easy to find a definition, which clearly separates the two. In this Chapter, an approach to distinguish various classes of case-markers and postposition has been introduced. The term 'case marker' or 'case clitic' is generally used for a word, which appears with a noun or a noun phrase such that the resultant phrase is a case marked noun phrase. While for a postposition, the resultant phrase is a postpositional phrase that acts as an adjunct to verb phrase. Some terms are defined below which may be referred in this chapter.

Transitivity refers to the number of objects a verb requires or takes in a grammatically well-formed clause or a sentence. The argument structure of a verb always contains subject and zero, one or two objects. The transitivity refers only to objects present in the argument structure of a verb. A subject is treated as a specifier

of the verb, while the object noun phrases appear in complement position in grammar modeling theories like X-bar and HPSG. Urdu, in contrast, has a flat phrase structure with rich case marking system, which allows relatively free order of phrase structure of sentence daughter phrases, and the verb is sister to subject noun phrase. The specifier and verb phrase thus do not appear in Urdu as in English.

Valency refers to the total number of arguments controlled by a predicate. Thus verb valency counts all the arguments of the verb including subject, objects, oblique case marked noun phrases and complement phrases. Valency is more relevant for analysis of Urdu verb's argument structures presented in this chapter for causative verbs and for other cases, which are marked with marker '*sey*'.

Thematic role is the semantic relationship between a predicate (e.g. a verb) and an argument (e.g. the noun phrases) of a sentence. There are different thematic roles available in the literature and different authors agree on different roles. The more widely used thematic roles are briefly reviewed here.

Agent is the one who deliberately performs the action, the one who is the principal cause of action and/or the one that controls the event, e.g., '*Hamid* ate the apple'. **Experiencer** is the one who gets affect of sensory, emotional or abstract input or the one who is unconsciously participating in the event, e.g., '*Anjom* is shocked', and '*Hamid* fears heights'. **Beneficiary** is the one who benefits from the action, e.g., 'The teacher teaches *Anjom*', and 'The teacher gave *Anjom* the book'. **Theme or Patient** is the role of the undergoer of an action, e.g., 'The boy crushed *the snake*', and 'The teacher gave Anjom *the book*'. **Instrument** is a thing used to carry out the action, e.g., 'Hamid cut the apple with *the knife*'. **Location** is the place in space and time where the action occurs, e.g., 'Hamid plays cricket in *the park*'. **Goal** is the person or place towards which action is directed, e.g., 'Hamid is going to *the school*', 'He writes a letter to *her*'. **Source** is the person or place from where the action is initiated, e.g., 'The rain is coming from *the west*', and 'He received a letter from *the principal*'.

Thematic hierarchy presents relative prominence among various thematic roles. The '>' sign means that role on left side has more prominence than on right side. There are variations in the literature, however the more acceptable (Bresnan 2001; Dalrymple 2001) is given in (94).

(94) agent > beneficiary > experiencer/recipient > instrument > patient/goal/theme > locative

These thematic roles are mapped to the grammatical functions in the argument structure of verbs. The mapping of grammatical functions and thematic roles is called *linking* or *mapping theory*. There are many approaches for mapping with theoretical

details (Butt 2005), however, usually agent and experiencer roles are mapped to subjects; patient and theme roles are mapped to objects; and goal/beneficiary are mapped to indirect objects. Locative, instrument, source and goal roles fill oblique arguments or they are attached as adjuncts as summarized below.

subject	–	agent, experiencer
object	–	patient, theme
indirect object	–	goal, beneficiary
oblique arguments	–	instrument, locative, source, goal

This chapter presents the data and analysis to show that the role of case marker '*sey*' is quite diverse and it adopts various grammatical functions or thematic roles in the argument structure of different verbs. The role of '*sey*' is described as versatile, and it is treated as the 'instrumental case' which adopts different roles (Mohanan 1990; Butt and King 2002). The marker '*sey*' marks subjects, objects, instruments, time and space nouns, post-positional phrases, adverbial phrases, etc. The analysis presented in this chapter shows that semantic considerations simplify classification of these roles. It is also shown that the marker '*sey*' marks '*indirect subjects*', for causative form 2 verbs. At the end, the chapter includes evidence of Urdu tetravalent causative verbs and presents a model for their handling.

7.1 Classification of Case Markers and Postpositions

For languages with case marking, mostly, the 'case marker' is morphologically attached at the lexical level. The Urdu-Hindi noun changes its form at the lexical level which is sometimes referred to as a case (Mohanan 1994; Arsenault 2002). Other case-markers in Urdu-Hindi that help in mapping the verb argument structure appear as syntactic unit. To distinguish between syntactic case marking, morphological case marking and other post-positions, it is proposed that these may be classified based on the way these are handled or according to their function. The case marking and postposition system in Urdu/Hindi have been divided into five categories: (i) noun form, (ii) core case markers, (iii) oblique case markers, (iv) possession markers and (v) 'pure' post-positions. This division of case markers into these categories is primarily based on the difference in computational modeling required in each case. The division of case markers may be based on morphological (lexical), structural (syntactic) and on functional (semantics) reasons. Therefore, the division presented in this work borrows heavily from the division of case markers presented by (Butt and King 1999), which includes lexical, structural, semantic and quirky case. However, the divsion presented in this work separates possession marking and also includes use of semantic features to help distinguish core and oblique verb arguments. Figure 7.1

shows hierarchical structure of the case markers and post-positions in Urdu and Hindi, which are explained in the following sections.

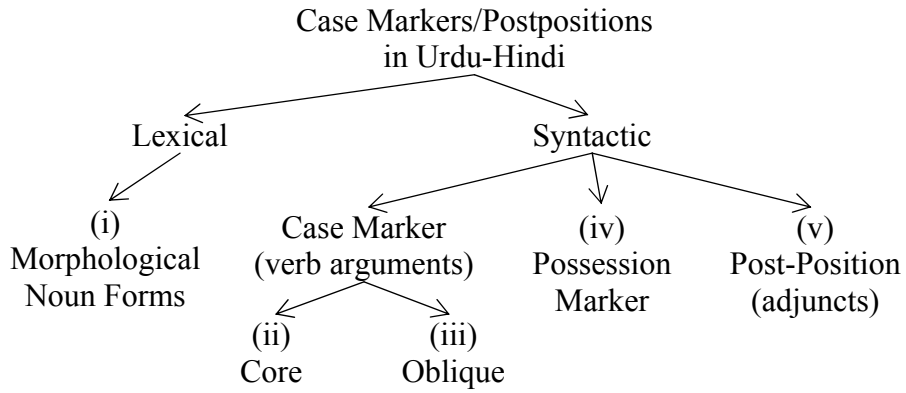


Figure 7.1: Classification of Case-Markers/ Postpositions in Urdu-Hindi

7.1.1 Noun Forms

Nouns in Urdu/Hindi appear in nominative, oblique and vocative morphological forms as shown in Figure 7.1. The syntactic test which employ coordination show that these noun suffixes like –*ey* in the oblique noun forms cannot be used in coordinated structures (Butt and King 2002) as shown in (95). The suffix is tightly coupled with the word as a unit, and this suffix cannot be taken common in the coordination. These suffixes are, therefore, lexical in nature and need to be handled morphologically at lexical level, while other case markers and postposition can be coordinated and those are therefore syntactic in nature. The example (96) shows that the ergative marker '*ney*' can be used in a coordinated structure.

(95) (a) گھوڑے اور بکرے
ghor-ey aor bakr-ey
horse-*sg.masc.obl* and goat-*sg.masc.obl*
horses and goats

(b) *گھوڑ اور بکرے
**ghor aor bakr -ey*
*horse and goat *-sg.masc.obl*
horses and goats

(96) (a) گھوڑے نے اور بکرے نے
ghor-ey=ney aor bakr-ey=ney
horse=*erg* and goat=*erg*
horses and goats

(b) گھوڑے اور بکرے نے
ghor-ey aor bakr-ey =ney
horse and goat =*erg*
horses and goats

The lexical suffixes do not play direct role in linking or mapping to the verb argument structure, as only noun form cannot tell which grammatical function noun may adopt. The oblique form is used with case markers and postpositions, which impart verb categorization features. However, the vocative form is used as 'subject' in the imperative mood. As the vocative form is governed by the verb in the imperative mood, therefore it is the only example of 'lexical case' in Urdu or Hindi. The

nominative form appears in the absence of case marker or postposition. These have already been discussed in the section on morphology, and are being reproduced in Table 7.1.

Table 7.1: Noun Forms in Urdu

Nominative (NOM) قائم		Oblique (OBL) محرف		Vocative (VOC) ندائی	
لڑکا	*laRkaa*, boy	لڑکے	*laRkey*, boy	لڑکے	*laRkey*, boy
لڑکی	*laRkee*, girl	لڑکی	*laRkee*, girl	لڑکی	*laRkee*, girl
لڑکے	*laRkey*, boys	لڑکوں	*laRkaoN*, boys	لڑکو	*laRkao*, boys
لڑکیاں	*laRkeeaN*, girls	لڑکیوں	*laRkeeoN*, girls	لڑکیو	*laRkeeo*, girls
مرغا	*morGaa*, rooster	مرغے	*morGey*, rooster	لوگو	*laogao*, people
کمرہ	*kamrah*, room	کمرے	*kamrey*, room	بچو	*bach.chao*, children

7.1.2 Core Case Markers

The core case markers are those that assign nouns a universal grammatical relation like subject, object and indirect object. These core grammatical relations in a sentence are directly controlled by verbal predicate and these help noun find a position in the argument structure of the verb. These are counted in verb transitivity as well as in valency of the verbal predicate. These core case markers will be discussed in more details later in this chapter. The case marker and corresponding grammatical relation is summarized as follows:

no marker – subject, object
‘نے’, *ney* – subject
‘کو’, *kao* – object, indirect object, subject
‘سے’, *sey* – subject, object

7.1.3 Oblique Case Markers

The oblique case markers are those that assign noun the oblique grammatical relation associated with a semantic role, these are governable by verbal predicate through its argument structure. The noun phrase marked with an oblique case is not an optional phrase in a sentence, as its presence is predictable from the argument structure of a verb, in contrast to an optional post-positional phrase, which is not predictable from the argument structure of the verb. As English do not have a case marking system, the oblique arguments of the verbal predicate are treated as prepositional phrases. In languages with strong case marking, like Urdu, the oblique arguments may be treated as case marked rather than ‘simple’ postpositional phrases. For some Australian languages, such as Warlpiri, case marked oblique phrases have been observed (Nordlinger 1998). Few markers that act as the oblique case markers are:

'سے', *sey*	instrument, space, time, etc.
'میں', *meyN*	in
'پر', *par*	on, at

The oblique case marked noun phrases are controlled by the argument structure of the verb and therefore these are counted in the valency of the verb. However, these are not counted in the transitivity of the verb. The verbs '*nekaal-naa*' (to take out) and '*rakh-naa*' (to put), '*Daal-naa*' (to put in) are transitive verbs but the argument structure of these verbs contains three arguments, as shown in (97), which means that the valency of these verbs is three. For verb '*nekaal-naa*' (to take out) one subject, one source location and one object is required, while for verb '*rakh-naa*' (to put) one subject, one destination location and one object is required. Two examples of oblique case markers in Urdu are shown in (98) and (99) as follows. These source or destination locations are not just bare locations in the form of post positions, because if we use destination location with '*nekaal-naa*' and source location with '*rakh-naa*' then the sentence will not be acceptable as shown in (100) and (101)

(97) *nekaal-naa*< 'agent', 'source location', 'patient'>
rakh-naa< 'agent', 'destination location', 'patient'>

(98) لڑکے نے فرج سے پانی نکالا

laRk-ey=ney	*ferej=sey*	*paanee*	*nekaal-aa*
boy-*sg.masc=erg*	fridge=*source*	water=*nom*	take out-*perf.sg.masc*

The boy took the water out from the fridge.

(99) آدمی نے کمرے میں سامان رکھا

aadmee=ney	*kamrey=meyN*	*saamaan*	*rakh-aa*
man-*sg.masc=erg*	room=*dest*	luggage	put-*perf.sg.masc*

The man put the luggage in the room.

(100) *لڑکے نے فرج میں پانی نکالا

**laRk-ey=ney*	*ferej=meyN*	*paanee*	*nekaal-aa*
boy-*sg.masc=erg*	fridge=*dest*	water=*nom*	take out-*perf.sg.masc*

*The boy took out the water in the fridge.

(101) *آدمی نے کمرے سے سامان رکھا

**aadmee=ney*	*kamrey=sey*	*saamaan*	*rakh-aa*
man-*sg.masc=erg*	room=*source*	luggage	put-*perf.sg.masc*

*The man put the luggage from the room.

However, for few liquid objects, sometimes the verb '*nekaal-naa*' may be used with destination location and the sentence is well formed without mentioning a source

location as shown in sentence (102), but in these cases a source location is *semantically implied* to be known. The destination location is an adjunct in this case.

(102) لڑکے نے کپ میں چائے نکالی

laRk-ey=ney *cap=meyN* *(X=sey)* *chaaey* *nekaal-ee*

boy-*sg.masc=erg* cup=*dest* *X=source* tea=*nom* take out-*perf.sg.fem*

A boy 'took out' tea in a cup (from a teapot).

7.1.4 Possession Marking

The possession marking is represented by genitive markers (or postposition, as it is called sometimes) is different from case markers for the following features:

1. The possession markers appear between two nominals and cannot form a 'noun phrase' by combining with just one nominal
2. The possession markers change form to agree in gender and number with the second nominal
3. The possession markers assign that first nominal is the possessor of second nominal
4. The possession markers are not controlled by a verbal predicate and therefore do not directly mark a grammatical function

Four characteristics mentioned above suggest that a 'genitive' or 'possession' marker is distinct from a case marker. Therefore, for these markers a new term 'possession marker' instead of 'genitive case marker' is being proposed. This distinction is especially useful in analyzing the syntactic structure represented by 'possession marker' as shown in section 7.5. There are three possession markers in Urdu, which require first nominal in the oblique form and gender-number agreement with second nominal.

Possession Marker	Gender	Number
'کا', *kaa*	masc	sg
'کی', *kee*	fem	–
'کے', *key*	masc	pl

7.1.5 Postpositions

The pure postpositions are those that are not controlled by verbal predicates and a sentence is complete in its meaning with or without postpositional phrases. Postpositional phrases are optional in the sense that these are not controlled by the argument structure of the verb. These, therefore, are counted neither in the transitivity nor in the valency of a verb. A larger list of postpositions in Urdu is given in Chapter 10. Semantic features of nouns, as employed for for case markers, are also important for better machine translation of the postpositional adjunct phrases from one natural

language to another natural language. A few postpositions, which acts as adjuncts in Urdu, are listed below:

'میں', *meyN*	in
'پر', *par*	on
'کے لیے', *key leeey*	for

(103) آدمی کمرے میں کھانا کھا رہا ہے

aadmee	*kamrey=meyN*	*khaanaa*	*khaa rahaa hay*
man-*sg.masc=nom*	room=*loc.*	food	eat-*sg.masc.prog*

The man is eating food in the room.

For example, the sentence in (103) is complete, even if the postpositional phrase '*kamrey meyN*' (in the room) is omitted. The postpositional phrases add information to the event happening but are not directly related to the arguement structure of the verbal predicate. There may be zero or more postpositional phrases, which appear as a set of adjuncts to a verbal predicate.

7.2 Urdu Case Marking Phrase Structure

In HPSG, a word in a phrase is designated as 'head' of the phrase and each phrase is recognized through its head. For example, the head of verb phrase is a verb and the head of a postpositional phrase is a postposition itself. This is an interesting debate that the head of a 'case marked noun phrase' is a case marker or a noun itself and there is another debate that whether case marker selects noun or noun selects case marker in a 'case marked noun phrase'. One approach (Butt and King 2002) is that case marker (K) functions as the head of Case Phrase (KP). The structure of phrase in (104) is shown in Figure 7.2 (a), where it is assumed that oblique marking on nouns (the singular oblique morpheme *–ey*) is the result of the complement-head relationship between the K and the NP. The NP is required to be in oblique form whenever there is an unconcealed K head. However, not all NPs contain surface morpheme *–ey* to show obliqueness. Many nouns have no apparent oblique form for singular nouns, but these have oblique morpheme *–ooN* for plural nouns. For the noun phrases that have no morpheme to show oblique form, nominative form is used for oblique form.

Another way to analyze case marking in Urdu is to assume that the noun in oblique form requires a case marker and the resultant phrase is a NP instead of a KP. In this representation, the head of phrase is a noun as shown diagrammatically in Figure 7.2 (b).

(104) لڑکے نے

laRk-ey=ney

boy-*sg.obl.masc=erg*

In fact, handling case marked structures is complicated as the case marked NP (or KP) in Urdu synthesize syntactic features from both a noun and a case marker.

(a) Case Phrase (KP) (b) Noun Phrase (NP)

Figure 7.2: Case Phrase verses Noun Phrase

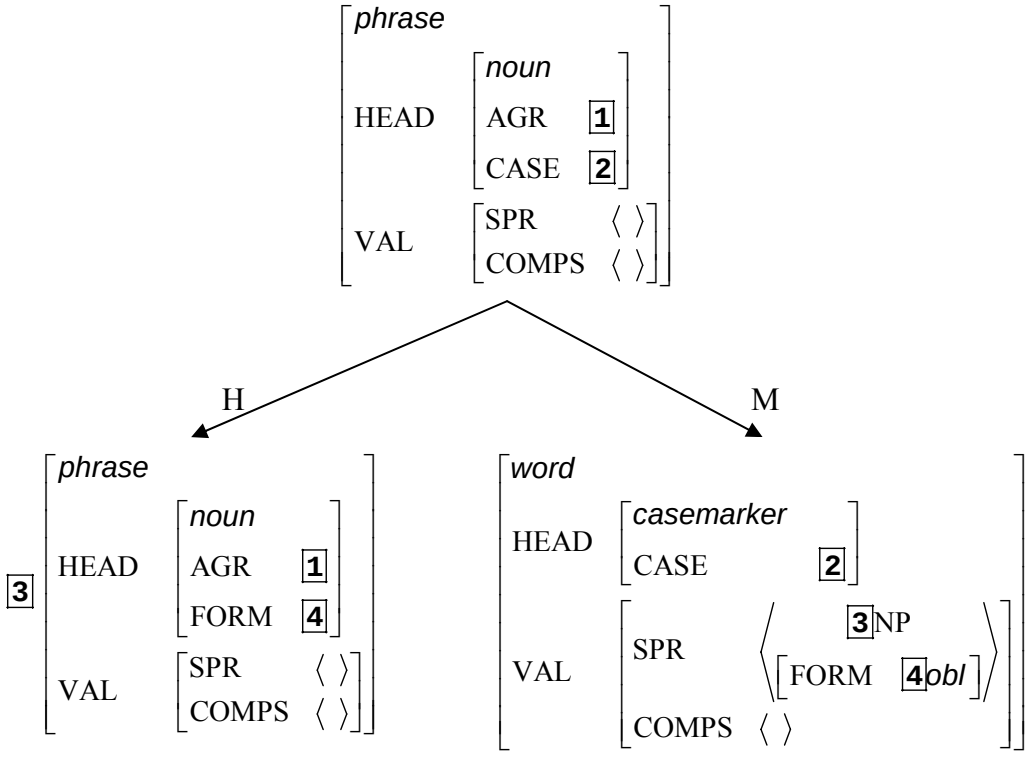


Figure 7.3: Case Marking in Urdu: Proposal 1

HPSG based phrase structure rule shown in Figure 7.3 is being proposed. The head daughter (H) is a noun or a noun phrase. The mother NP gets agreement (AGR) features like gender, number from the head daughter (H) and gets CASE feature from the case marker (M). The number 1 in box with the AGR feature of the mother and the daughter noun phrase describes that these values are the same. Similarly, the boxed number 2 expresses that CASE value of mother NP is required to match with the value of the same attribute of the case marker M. The noun phrase is proposed in

this rule as the specifier of case marker, which means that whenever there is an overt case marker, the noun or noun phrase numbered 3 is required. In this rule, the case marker selects noun but the resultant phrase is a noun phrase as the head of the phrase is designated a noun phrase. With the restriction using number 4 the attribute FORM of the specifier of the case marker must match with the same attribute of the noun phrase, which fills the specifier slot. This is necessary for the noun-case agreement requirement that the oblique form of a noun (or a noun phrase) is needed with case markers.

(105) لڑکے نے نے *
**laRk-ey=ney=ney*
*boy-*sg.obl.masc=erg=erg*

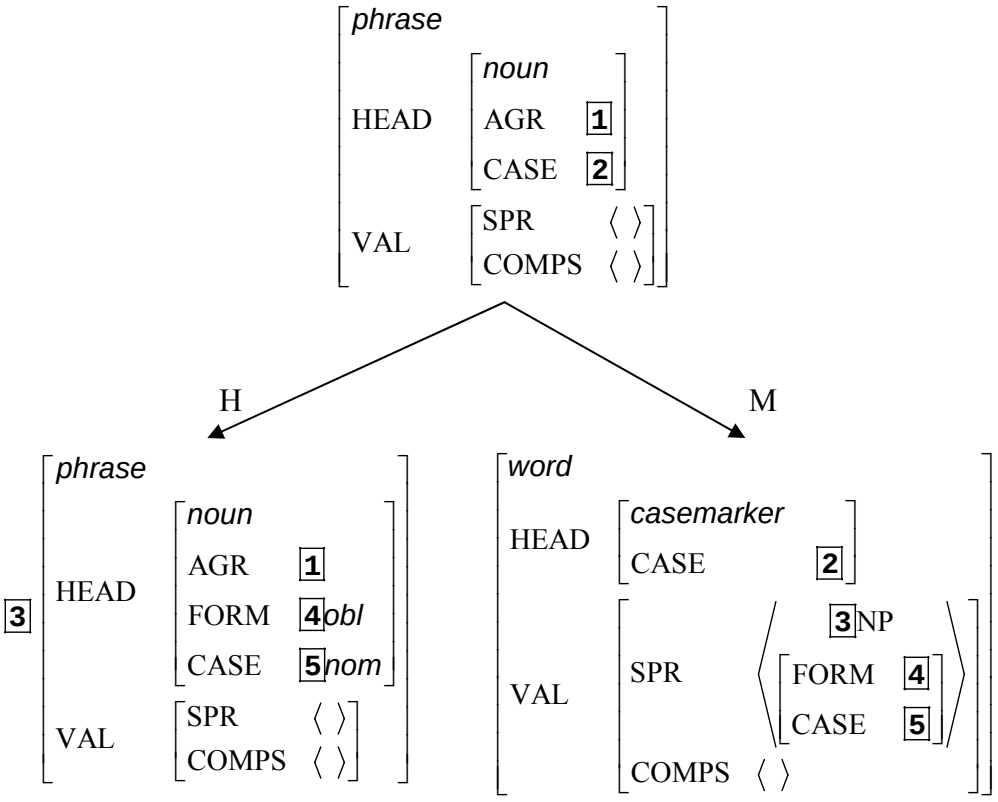


Figure 7.4: Case Marking in Urdu: Proposal 2

The HPSG based rule 'proposal 1' may be used to form noun phrases using the case markers in Urdu. However, in the absence of a case marker, the default case of the noun phrase needs to be 'nominative' which is not mentioned in the rule. Moreover, the above rule may result in recursion and could generate sentences with cascaded case markers as shown in (105), which means that above rule may register more than one case marker for a single noun phrase. To handle the above conditions, the following is proposed. The attribute CASE of each lexical 'noun' is assigned a

value 'nominative' by default along with extra constraint that the 'noun phrase' 3 in the specifier argument of case marker (M) requires that its CASE be 'nominative' using 5 in addition to 'oblique' FORM through 4. The extended rule as 'proposal 2' is shown in Figure 7.4, which takes care of default nominative case requirement for a noun phrase in the absence of case marker and at the same time avoids recursive inclusion of cascaded case markers.

It may be noted that the 'proposal 2' rule does not include 'genitive' or 'possessive' marker. It is assumed that the 'possessive markers' are distinct from the 'case markers' due to characteristics presented in section 7.1.4 and therefore require separate treatment. The phrase structure rules for 'possessive markers' are proposed in section 7.5.

The LFG based phrase structure rule for case marked noun phrase is shown in (106), which describes that a mother noun phrase (NP) can be constructed with a noun phrase (NP) followed by a case marker (CM).

(106) NP ⟶

NP	CM
(↑ NUM) = (↓ NUM)	(↑ CASE) = (↓ CASE)
(↑ GEND) = (↓ GEND)	(↑ FORM) =c *oblique*
(↑ FORM) = (↓ FORM)	
(↓ CASE) =c *nom*	

The functional schemata attached with daughter NP describes that mother NP's f-structure will take NUM, GEND and FORM attributes from the f-structure of daughter NP. The daughter NP has a constraint that its CASE value be 'nominative', which is needed to avoid cascaded inclusion of case markers. The functional schemata attached with the CM node expresses that mother NP's CASE value is to be taken from the f-structure of CM. A constraint equation at CM node checks that the FORM attribute of the mother NP has a value 'oblique'. Indirectly this constraint is applied to the FORM attribute of daughter NP, as the mother NP has taken this 'oblique' value from the daughter NP.

7.3 Analysis for Urdu Case Markers

The following sections present analysis of Urdu case markers along with example sentences and analysis. The nominative, ergative, dative and accusative case has been analyzed extensively in the literature (Mohanan 1994; Butt and King 2002). A brief review of these case markers has been included with somewhat different analysis by including semantic features of nouns and by using verb valency instead of verb transitivity. However, a detailed analysis of case marked with '*sey*' marker and its role in causative verbs is discussed using semantic features of nouns and verb valency.

7.3.1 Nominative Case

If there is no case marker with the noun (or the noun phrase), the noun is said to be in nominative case, which is the default case for noun phrases, as shown in (107) below. Here both 'boy' and 'book' are in nominative form, which assume subject and object functions respectively.

(107) لڑکا کتاب خریدے گا

laRk-aa	*ketaab*	*xareed-ey*	*g-aa*
boy-*sg.masc=nom*	book=*nom*	buy-*subj.obl*	AUX-*future-sg.masc*

A boy will buy a book

(108) S ⟶ NP NP V

(↑ SUBJ) = ↓ (↑ OBJ) = ↓ ↑ = ↓
(↓ CASE) = *nom* (↓ CASE) = *nom*
(↓ N-CONCEPT) =c *animate*

The example contains two nominative NP's in a sentence and both NP's can fill subject and object slot of verb's argument structure. The subject slot should be filled with an agent. For nominative subjects, LFG rule shown in (108) includes a constraint that a NP can fill the subject slot only if it has a value 'animate' for the noun concept attribute. The agreement between a verb and a noun is with the highest nominative argument in the argument structure of the verb. In this example, therefore according to thematic hierarchy shown in (94), agent (subject) assumes higher role and the verb agreement is with '*laRkaa*' (the boy), instead of agreement with object '*ketaab*' (the book), which assumes lower role. The f-structure for sentence in (107) is shown in Figure 7.5, where both subject and object have nominative case but the 'animate' attribute helps to find that a 'boy' is more suitable as a subject.

$$\begin{bmatrix} \text{PRED} & \text{' } \textit{buy} \langle \text{SUBJ, OBJ} \rangle \text{ '} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{boy} \text{ '} \\ \text{N-SEM} & [\text{N-CONCEPT} \quad \textit{animate}] \\ \text{CASE} & \textit{nom} \\ \text{SPEC} & \text{' } \textit{a} \text{ '} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{book} \text{ '} \\ \text{N-SEM} & [\text{N-CONCEPT} \quad \textit{thing}] \\ \text{CASE} & \textit{nom} \\ \text{SPEC} & \text{' } \textit{a} \text{ '} \end{bmatrix} \\ \text{TENSE} & \textit{future} \end{bmatrix}$$

Figure 7.5: F-Structure of Sentence '*laRkaa ketaab xareedey gaa*'

7.3.2 Ergative Case

Noun phrase marked with case marker 'نے', *ney*, expresses the role of an actor or agent that fills the 'subject' argument in the list of grammatical functions. The ergative case appears for verbs in a perfective form having valency greater than one. An example is shown in sentence (109) for transitive verb '*xareed-naa*' (to buy).

(109) لڑکے نے کتاب خریدی
laRkey=ney *ketaab* *xareed-ee*
boy-*sg.masc=erg* book.*nom* buy-*perf.sg.fem*
A boy bought a book.

The example contains one ergative and one nominative argument in the sentence. The verb-noun agreement is with highest nominative argument of in the argument structure of the verb according to thematic hierarchy shown in (94). In this example, subject NP is ergative and object NP is the nominative. Therefore, the verb agreement is with object '*ketaab*' (the book).

As a general rule, the ergative case marker '*ney*' is not used with intransitive verbs but there are few exceptions to this rule for intransitive (monovalent) verbs like '*thook-naa*' (to spit) and '*moot-naa*', (to piss) for which case marker '*ney*' is required and nominative form is not acceptable. The acceptable and unacceptable usage of ergative case for intransitive verbs is shown in (110).

(110)

(a) وہ گرا
woh geraa
He=*nom* fall.*perf*
He fell

(b) * اس نے گرا
**aes ney geraa*
He=*erg* fall.*perf*
He fell

(c) وہ سویا
woh saoyaa
He=*nom* sleep.*perf*
He slept

(d) * اس نے سویا
**aes ney saoyaa*
He=*erg* sleep.*perf*
He slept

(e) مظفر ڈرا
mozafar daraa
Mozafar=*nom* scare.*perf*
Mozafar scared

(f) * مظفر نے ڈرا
**mozafar ney daraa*
Mozafar=*erg* scare.*perf*
Mozafar scared

(g) ظفر نے تھوکا
Zafar ney thookaa
Zafar=*erg* spit.*perf*
Zafar spitted.

(h) * ظفر تھوکا
**Zafar thookaa*
Zafar=*nom* spit.*perf*
Zafar spitted.

(i) بکری نے موتا
bakree ney mootaa
Goat=*erg.fem* piss.*perf*
Goat pissed

(j) *بکری موتی / *بکرا موتا
**bakree mootee / *bakraa mootaa*
Goat=*nom* piss.*perf*
Goat pissed

Some intransitive verbs listed in (111) are usually used without ergative case but they are also known to be acceptable in ergative case for deliberate and purposeful actions (Abdul-Haq 1991; Mohanan 1994; Butt and King 2002). A brief survey is carried out to check contemporary Urdu usage in Lahore and Islamabad and the sentences shown in (111) are presented to few people. It is found that the ergative form is scarcely acceptable in a volitional sense for transitive verbs and to show volitional effect it is better to use a participle adverbial conjunctive, '*jaan boojh kar*' (deliberately).

It is not a general rule that using ergative subjects with intransitive verbs expresses a volitional effect, and only few intransitive verbs may require ergative subject in perfective tenses to show volitional effect. It is therefore suggested that we can use a general rule that intransitive verbs of Urdu require nominative subjects. If there are intransitive verbs that could be used with ergative subjects, they may be specifically marked for the ergative requirement in the lexicon.

(111)

(a) وہ نہایا
woh nahaayaa
He=*nom* bathe.*perf*
He bathed

(b) * اس نے نہایا
** aes ney nahaayaa*
He=*erg* bathe.*perf*
He bathed (deliberately).

(c) وہ کھانسا
woh khaansaa
He=*nom* cough.*perf*
He coughed

(d) ? اس نے کھانسا
? aes ney khaansaa
He=*erg* cough.*perf*
He coughed (deliberately).

(e) ظفر چھینکا
Zafar chheenkaa
Zafar=*nom* sneeze.*perf*
Zafar sneezed

(f) ? ظفر نے چھینکا
? Zafar ney chheenkaa
Zafar=*erg* sneeze.*perf*
Zafar sneezed (deliberately).

(g) مظفر چیخا
mozafar cheeKhaa
Mozafar=*nom* scream.*perf*
Mozafar screamed

(h) ? مظفر نے چیخا
? mozafar ney cheeKhaa
Mozafar=*erg* scream.*perf*
Mozafar screamed (deliberately).

(i) وہ چلایا
woh chelaayaa
He=*nom* shout.*perf*
He shouted

(j) ? اس نے چلایا
? aes ney chelaayaa
He=*erg* shout.*perf*
He shouted (deliberately).

(112)

(a) ظفر نے شیشی توڑی
Zafar ney sheeshee taoRee
Zafar=*erg* bottle=*nom* break.*perf*
Zafar broke the (glass) bottle.

(b) * ظفر شیشی توڑی
** Zafar sheeshee taoRee*
Zafar=*nom* bottle=*nom* break.*perf*
Zafar broke the (glass) bottle.

(c) مظفر نے آم کھایا
mozafar ney aam khaayaa
Mozafar=*erg* mango=*nom* eat.*perf*
Mozafar ate the mango.

(d) * مظفر آم کھایا
** mozafar aam khaayaa*
Mozafar=*nom* mango=*nom* eat.*perf*
Mozafar ate the mango.

(e) میں نے بات سمجھی
mayN ney baat samjhee
I=*erg* communication=*nom*
comprehend.*perf*
I comprehended the communication

(f) * میں بات سمجھا
** mayN baat samjhaa*
I=*nom* communication=*nom*
comprehend.*perf*
I comprehended the communication

(g) میں نے پڑھنا سیکھا
mayN ney paRhnaa seekhaa
I=*erg* read=*nom* learn.*perf*
I learned reading

(h) * میں پڑھنا سیکھا
**mayN paRhnaa seekhaa*
I=*nom* read=*nom* learn.*perf*
I learned reading

Transitive and ditransitive verbs (or for the verbs having valency greater than one, this includes tetravalent verbs) when appear in perfective form require subjects marked with case marker '*ney*', i.e., ergative subjects. Sentences shown in (112) employ transitive verbs in perfective form, the sentences with nominative subject are not acceptable, while sentences with an ergative subject are acceptable. However, few exceptions exist for divalent verbs, which require nominative subjects even in perfective forms, the examples are shown in (113).

(113)

(a) وہ کتاب لایا
woh ketaab laayaa
He=*nom* book=*nom* bring.*perf*
He bring the book

(b) * اس نے کتاب لائی
**aes ney ketaab laayee*
He=*nom* book=*nom* bring.*perf*
He bring the book

(c) وہ شادی سے شرمایا
woh shaadee sey sharmaayaa
He=*nom* marriage=*inst* embarrass.*perf*
He embarrassed from the marriage

(d) ? اس نے شادی سے شرمایا
? aes ney shaadee sey sharmaayaa
He=*erg* marriage=*inst* embarrass.*perf*
He embarrassed from the marriage

(114) *ney*

```
(↑ CASE) = ergative
(↑ N-SEM N-CONCEPT) =c animate
((SUBJ ↑) V-FORM) =c perfect
((SUBJ ↑) V-VAL) ~= 1
((SUBJ ↑) SUBJ) = ↓
```

$$\begin{bmatrix} \text{PRED} & \text{' } \textit{xareednaa} \ \langle \text{SUBJ, OBJ} \rangle \text{ '} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{laRkaa} \text{ '} \\ \text{N-SEM} & [\text{N-CONCEPT} \quad \textit{animate}] \\ \text{CASE} & \textit{erg} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{ketaab} \text{ '} \\ \text{N-SEM} & [\text{N-CONCEPT} \quad \textit{thing}] \\ \text{CASE} & \textit{nom} \end{bmatrix} \\ \text{TENSE} & \textit{past} \\ \text{V-FORM} & \textit{perfect} \\ \text{V-VAL} & 2 \end{bmatrix}$$

Figure 7.6: F-Structure of '*laRkey=ney ketaab xareedee*'

Functional schema for the LFG based lexical entry of '*ney*' has been shown in (114), which marks an 'ergative case'. The entry expresses in the first equation that the CASE attribute of mother NP has a value 'ergative'. In second equation, which is a constraint equation, it is described that mother NP's semantic attribute must have a value 'animate', this is to verify that ergative case can be assigned only to animate nouns and inanimate nouns are not marked with ergative case. In third equation, a constraint is applied to verb form to be 'perfect'. The notation (**SUBJ ↑**) is for inside-out functional uncertainty, which is used to refer to a f-structure by traversing inside-out through the hierarchy of f-structures until the required f-structure having attribute SUBJ is found. The next constraint equation checks verb valency for ergative case should not be one, therefore the verb valency attribute 'V-VAL' can take values 2, 3 or 4 for Urdu verbs. The last equation expresses that noun marked with ergative case fills the subject argument of the verb.

Sometimes, apparently 'inanimate' nouns are assigned ergative case to mark them as agents, which could be assigned only to 'animate' nouns. These nouns are not intrinsically animated but there is some external force or power, which imparts them 'animate' attribute. The use of ergative case for such externally 'animated' nouns is shown in (115) and (116), and it is assumed that these nouns have semantic feature value as 'animate', which allows them to be used in ergative case.

(115) ریل گاڑی نے مجھے لاہور پہنچا دیا

rayl gaaRee=ney mojhey laahaor pohanch-aa dee-aa
train-*sg.masc=erg* me.*pron* Lahore.*nom* help reach-*caus1.perf.sg.mas* completely
The train caused me reach Lahore.

(116) زلزلے نے مکان گرا دیا

zzalzzaley=ney makaan ger-aa dee-aa
earthquake-*sg.masc=erg* house.*nom* cause fall-*caus1.perf.sg.mas* completely
The earthquake caused the house fall.

7.3.3 Dative Case

In a dative case, a noun phrase marked with case marker 'کو', *kao*, expresses the role of an indirect object, recipient, beneficiary or receiver as the third argument in the argument structure of ditransitive verbs, where the other two arguments are the subject and the object. An Urdu sentence expressing dative case is shown in (117), where 'book' is a direct object and receiver 'boy' is an indirect object marked with the dative case.

(117) میں نے لڑکے کو کتاب دی

mayN=ney *laRk-ey=kao* *ketaab* *d-ee*
I=*erg* boy-*sg.obl=dat* book.*nom* buy-*perf.sg.fem*
I gave the book to the boy.

(118) لڑکے کو سردی لگ رہی ہے

laRkey=kao *sardee* *lag rahee hay*
boy-*sg.obl=dat* cold.*nom* feel-*pres.continuous.sg.fem*
The boy is feeling cold.

(119) لڑکے کو بخار ہو گیا ہے

laRkey=kao *boxaar* *hao+ga-yaa* *hay*
boy-*sg.obl=dat* fever.*nom* happened-*perf.sg.masc* AUX-*pres*
The boy has got fever.

$$
\begin{bmatrix}
\text{PRED} & \text{' } dee \langle \text{SUBJ, OBJ}_{\text{GOAL}}, \text{OBJ} \rangle \text{ '} \\
\text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{' } pro \text{ '} \\ \text{PERS} & 1st \\ \text{NUM} & sg \\ \text{CASE} & erg \\ \text{N-SEM} & [\text{N-CONCEPT} \quad animate] \end{bmatrix} \\
\text{OBJ}_{\text{GOAL}} & \begin{bmatrix} \text{PRED} & \text{' } laRkaa \text{ '} \\ \text{CASE} & dat \\ \text{N-FORM} & oblique \\ \text{N-SEM} & [\text{N-CONCEPT} \quad animate] \end{bmatrix} \\
\text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } ketaab \text{ '} \\ \text{CASE} & nom \\ \text{N-SEM} & [\text{N-CONCEPT} \quad thing] \end{bmatrix} \\
\text{TENSE} & past \\
\text{V-FORM} & perfect \\
\text{V-VAL} & 3
\end{bmatrix}
$$

Figure 7.7: F-Structure of '*mayN=ney laRk-ey=kao ketaab dee*'

The Urdu verbs, which express some feeling or state change of someone, do not take ergative or nominative subjects in their argument structure, and employ dative

case for subjects as shown in (118) and (119). Some Urdu verbs that show 'physical feelings' like cold '*sardee*', hot '*garmee*', hunger '*bhook*', thirst '*peyaas*', etc. are used in dative case pattern shown in (118). Similarly, state change of subjects is expressed in dative case as in (119), for verbs like fever '*boxaar*', headache '*sar daard*', love '*peyaar*', hate '*nafrat*', etc.

The example (120) shows a usage of the dative case to represent an 'unwilling agent'. This dative case appears to represent a subject when infinitive verb form is used with auxiliary (or light-verb) '*paR-aa*', which represents a 'forced mood'. Another sentence mood represents a 'willing agent' having 'obligation' to do something. This 'obligation mood' is represented with dative case subject as shown in (121), where infinitive form is used with present auxiliary '*hay*'. This 'obligation mood' with the same semantics is sometimes used with ergative subject, but dative subject should be preferred over ergative subject.

(120) حامد کو سکول جانا پڑا

Haamed=kao	*sakool*	*jaanaa*	*paRaa*
Hamid-*sg=dat*	school.*nom*	go-*inf.sg.masc*	AUX-*forced mood*

Hamid went to the school (unwillingly, forcefully).

(121) حامد کو سکول جانا ہے

Haamed=kao	*sakool*	*jaanaa*	*hay*
Hamid-*sg=dat*	school.*nom*	go-*inf.sg.masc*	AUX-*pres*

Hamid has to go to the school (as a duty, obligation or responsibility).

The example (122) shows usage of a dative agent assuming the subject role in a sentence in the 'suggestion mood'. This mood uses infinitive form followed by a mood auxiliary '*chaah-ee-ey*', which signals recommendation, advisability or suggestion for the agent. This auxiliary is translated to 'should' in English.

(122) حامد کو سکول جانا چاہیئے

Haamed=kao	*sakool*	*jaanaa*	*chaaheeey*
Hamid-*sg=dat*	school.*nom*	go-*inf.sg.masc*	AUX-*suggestion mood*

Hamid should go to the school.

The features and constraints applied by '*kao*' for dative case using LFG based lexical entry are shown in (123). The first line of the lexical entry (123) for the dative marker '*kao*' assigns mother node's CASE value to be dative. The second line puts constraint that the semantic concept of noun be animate, which means that dative case can be assigned only for animate nouns. The curly brackets '**{**' and '**}**' are used to group choices. The choices are separated using 'or' symbol '**|**'. The first choice uses inside-out functional uncertainty to refer to some outer f-structure having attribute OBJ_{goal}, in that f-structure the verb valency is constrained to have value 3, which

means that dative case will assign 'object goal' function if the corresponding f-structure's verbal predicate is ditransitive. The second choice uses inside-out functional uncertainty to refer to the outer f-structure with SUBJ attribute, the verb valency attribute V-VAL is constrained to take value 2, which means dative-subject occurs for transitive verbs.

(123) *kao*

```
(↑ CASE) = dative
(↑ N-SEM N-CONCEPT) =c animate
{
   ((OBJgoal ↑) V-VAL) =c 3
   (OBJgoal ($) ↑)
|
   ((SUBJ ↑) V-VAL) =c 2
   (SUBJ ($) ↑)
}
```

7.3.4 Accusative Case

The accusative case of a noun or noun phrase is represented using case marker 'کو', *kao*, which expresses direct object, undergoer or patient usually for transitive verbs. The accusative marker '*kao*' is phonetically the same case marker used to mark the dative case, however, it marks a different grammatical function and therefore represents a separate case. The object represented by accusative case typically becomes subject under passivization. One example of it is given in sentence (125), in which 'dog' is in accusative case and occupies the patient, 'مفعول', *mafAool*, or 'object' grammatical function position in the argument structure of the verb. The accusative case is mostly used with the transitive verbs while dative case is used with ditransitive verbs to mark 'object' and 'indirect object' respectively. The accusative case is normally used to mark animate nouns as object, such as ergative case is used to mark animate nouns as subjects. The accusative marker is necessary especially for proper-animate nouns.

The use of accusative '*kao*' with animate nouns is dictated by the verb argument structure. The sentences in (125) to (128) are interesting examples, which illustrate that both nominative and accusative case can appear in the same structure, and which case will be allowed is dictated by the respective verb argument structure. The examples show phonetically the same verbs having different meaning and argument structure. In (125) 'dog' is in accusative case, while in (126) it is in nominative case. The verb 'مارا', *maar-aa*, is not the same in both sentences. In sentence (125) it means 'to beat', while in sentence (127) it means 'to kill'. The verbs 'beat' and 'kill' have different cases to fill 'object' role in the argument structures, as shown by lexical entries in (124), where 'beat' requires accusative case and 'kill' requires nominative

case for the object. Similarly, in sentence (127) the causative verb 'to help someone take bath' requires accusative case, while the causative verb 'to make someone fly' in (128) requires nominative object.

(124) مارنا (beat)
```
(↑ PRED)='maar-naa<SUBJ, OBJ>'
(↑ SUBJ CASE) =c { ergative | nominative }
(↑ OBJ CASE) =c accusative
```

مارنا (kill)
```
(↑ PRED)='maar-naa<SUBJ, OBJ>'
(↑ SUBJ CASE) =c { ergative | nominative }
(↑ OBJ CASE) =c nominative
```

$$\begin{bmatrix} \text{PRED} & \text{' } maaraa \langle \text{SUBJ, OBJ} \rangle \text{ '} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{' } aakmal \text{ '} \\ \text{CASE} & erg \\ \text{N-SEM} & \begin{bmatrix} \text{N-CONCEPT} & animate \\ \text{N-CLASS} & proper \end{bmatrix} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } kottaa \text{ '} \\ \text{CASE} & acc \\ \text{N-SEM} & [\text{N-CONCEPT} \quad animate] \end{bmatrix} \\ \text{TENSE} & past \\ \text{V-FORM} & perfect \\ \text{V-VAL} & 2 \end{bmatrix}$$

Figure 7.8: F-Structure of '*aakmal=ney kott-ey=kao maar-aa*'

(125) اکمل نے کتّے کو مارا

aakmal=ney	*kott-ey=kao*	*maar-aa*
Akmal=*erg*	dog-*sg.obl=acc*	beat-*perf.sg.masc*

Akmal beat a dog.

(126) اکمل نے کتّا مارا

aakmal=ney	*kott-aa*	*maar-aa*
Akmal=*erg*	dog-*sg.masc=nom*	kill-*perf.sg.masc*

Akmal killed a dog.

(127) بچّہ بکری کو نہلاتا ہے

bach.ch-aa	*bakr-ee=kao*	*nehl-aa-taa*	*hay*
mother-*sg.masc =nom*	goat-*sg.fem=acc*	bath-make.*caus1-repeat.sg.masc*	AUX.*pres*

A child is used to give bath to a goat.

(128) بچّے نے کبوتر اُڑایا

bach.ch-ey=ney	*kabootar*	*aoR-aa-yaa*
child-*pl.masc =erg*	pigeon-*sg.masc=nom*	fly-make.*caus1-perf.sg.masc*

A child made the pigeon fly.

The argument structure of the verbs used in the above example sentences is shown in (129) for the particular verb form and tense shown in examples. It is assumed that argument structure dictates the case selection.

(129) *maar-aa*< 'agent – ergative case', 'patient – ***accusative case***'>
maar-aa< 'agent – ergative case', 'patient – ***nominative case***'>
nehl-aa-taa< 'agent – nominative case', 'patient – ***accusative case***'>
aoR-aa-yaa< 'agent – ergative case', 'patient – ***nominative case***'>

The accusative 'کو', *kao*, is also known for signaling 'specificity' (Butt and King 2002) for inanimate objects (and sometimes for animate objects) as shown in (130). Moreover, if there is no nominative verb argument as in (125) and (130), then the default verb agreement is singular and masculine. By presenting the sentence (130) to native speakers of Urdu in Lahore and Islamabad, it is found that the specifier is either missing or implied by default in the sentence (or perhaps the pro-drop phenomenon). The more acceptable form of sentence (130) is shown in (131). For unspecified objects, the sentence (132) is more acceptable. Therefore, it is suggested that 'کو', *kao*, itself is not a marker for 'specificity' but there is missing or implied pronoun, which generates attribute for 'specificity' and requires '*kao*' to accompany.

(130) لڑکے نے کتاب کو خریدا ؟
? *laRk-ey=ney* *ketaab=kao* *xareed-aa*
boy-*sg.masc=erg* book.*sg.fem=acc* buy-*perf.sg.masc*
The boy bought the/this/that (particular) book.

(131) لڑکے نے اِس کتاب کو خریدا
laRk-ey=ney *aes* *ketaab=kao* *xareed-aa*
boy-*sg.masc=erg* **this**.*spec* book.*sg.fem=acc* buy-*perf.sg.masc*
The boy bought this (particular) book.

(132) لڑکے نے کتاب خریدی
laRk-ey=ney *ketaab* *xareed-ee*
boy-*sg.masc=erg* book.*sg.fem=nom* buy-*perf.sg.fem*
The boy bought a book.

LFG based lexical entry for '*kao*' expressing accusative case is shown in (133), which applies complex constraints. The first line of lexical entry (133) tells that the accusative marker '*kao*' assigns a value of 'accusative' to the mother node's CASE attribute. The second line describes that this f-structure is the object 'OBJ' in some outer f-structure found by traversing inside-out. Lines 3 to 7 put constraints on verb valency attribute 'V-VAL' that it can be assigned a value 2 or 4. Lines 9 and 10 put constraints that if the noun's semantic concept 'animate' and its class is 'proper', then accusative case can be assigned. The lines 13 and 14 describe another possibility that

for 'animate' or 'thing' object, the accusative case can be used but along with another constraint in line 12, which is applied to check the presence of a specifier.

(133) *kao*

```
(↑ CASE) = accusative
(OBJ ($) ↑)
{
  ((OBJ ↑) V-VAL) =c 2
|
  ((OBJ ↑) V-VAL) =c 4
}
{
   (↑ N-SEM N-CONCEPT) =c animate
   (↑ N-SEM N-CLASS) =c proper
|
  {
     (↑ N-SEM N-CONCEPT) =c thing
  |  (↑ N-SEM N-CONCEPT) =c animate
  }
   (↑ SPEC) =c definite
}
```

The 'accusative case' of Urdu needs more detailed analysis to describe the usage of marker '*kao*'. The example in (134) shows the **postpositional use** of '*kao*', which typically follows an infinitive. This Urdu postpositional '*kao*' can be replaced with another equivalent and more popular Urdu postposition, کے لیے, '*key leeey*' as shown in (135). Both '*kao*' and '*key leeey*' are translated to preposition 'for' in English. Although '*kao*' is sometimes acceptable after an infinitive, yet normally '*key leeey*' is preferred as it is unambiguous and more frequently used.

(134) حامد نے انجم کو کتاب پڑھنے کو دی ؟

? *Haamed=ney anjom=kao ketaab paRh-ney=kao d-ee*
Hamid-sg.m=erg Anjom=*dat* book.*sg.fem=nom* read-*inf.pl=pp* give-*perf.sg.fem*
Hamid gave Anjom the book for reading.

(135) حامد نے انجم کو کتاب پڑھنے کے لیے دی

Haamed=ney anjom=kao ketaab paRh-ney=key leeey d-ee
Hamid-sg.m=erg Anjom=*dat* book.*sg.fem=nom* read-*inf.pl=pp* give-*perf.sg.fem*
Hamid gave Anjom the book for reading.

(136) حامد نے انجم کو کتاب پڑھنے دی

Haamed=ney anjom=kao ketaab paRh-ney d-ee
Hamid-sg.m=erg Anjom=*dat* book.*sg.fem=nom* read-*inf.pl* AUX-*permissive*
Hamid let Anjom read the book.

The sentence (136) is similar to (134), but the verb '*d-ee*' in (134) and (136) are different in meaning and argument structure. In (134), '*d-ee*' means 'give' and requires three arguments 'a giver', 'a recipient' and 'a gift', while in (136), '*d-ee*'

means 'let' and requires three arguments 'one who allows an action', 'one who is allowed' and 'an action which is allowed'.

7.4 Classification of Cases Marked with '*sey*'

The noun (or noun phrases) marked with case marker 'سے', *sey* are mostly characterized as an 'instrumental case' in the Urdu and Hindi literature (Mohanan 1994; Butt and King 2002). The case marker '*sey*' is too versatile and noun cases marked with '*sey*' occupy different grammatical relations. The '*sey*' as case marker fills subject, object, indirect subject and oblique argument roles that are controlled by verb argument structure and '*sey*' as postposition appear in a post-positional phrase or in an adverbial phrase which act as adjunct to the verb phrase. Sometimes '*sey*' is used for comparison between two things and sometimes it is used with adjectives. Therefore, the use of post-position '*sey*' is quite versatile and it may be classified according to the function in various roles, instead of using it as a bare 'instrumental case' marker in all cases. In the following sections, this case-marker and/or post-position is being modeled for different situations.

7.4.1 Agentive Case

An animate noun (or noun phrase) marked with case marker 'سے', *sey*, is categorized as an 'agentive case' and it occupies 'subject' or 'indirect subject' role in the verb's argument structure. Sentence (137) shows agent in passive voice form, where focus is on the object 'letter', which appears in the nominative case and therefore the gender-number agreement of verb is with object. In Urdu, the agent in active voice is assigned 'nominative' or 'ergative' case, while in passive voice it is changed to 'agent case'. For the English sentence in passive voice, the subject and the object positions are interchanged and therefore it is assumed that the object (in active voice) has become the subject (in passive voice). While in Urdu, the position of the subject and the object are relatively less important due to its free phrase order.

(137) خط لڑکے سے لکھا گیا

xatt	*laRk-ey=sey*	*lekh-aa*	*ga-yaa*
letter.*sg.masc=nom*	boy-*sg.masc=agent*	write-*perf.sg.masc*	go-*perf.sg.masc*

A letter was written by a boy.

(138) خط لکھا گیا

xatt	*(X=sey)*	*lekh-aa*	*ga-yaa*
letter.*sg.masc=nom*	(X=*agent*)	write-*perf.sg.masc*	go-*perf.sg.masc*

A letter was written (by someone).

For example of a passive sentence in (137), in both English and Urdu, the 'doer of the action' is 'a boy' and the 'undergoer of the action' is 'a letter', therefore

according to thematic hierarchy they should fill subject and object arguments respectively. However, the analysis become troublesome, when a well-formed passive voice sentence could be produced without an agent as shown in (138). The analysis of passive, *majhool* – مجہول, presented in this work assumes that in a passive voice, the primary focus is on the undergoer and the agent becomes secondary, and therefore sometimes omitted. It is assumed that 'semantic subject' is still the agent and if the agent is omitted from a passive sentence, then it is 'semantically implied' as there is a slot for agent in the argument structure of the verb. We cannot assume that for an action there is no actor. Therefore, for sentence (138), an unknown agent 'X' is assumed to fill the 'writer' slot of the verb 'write'.

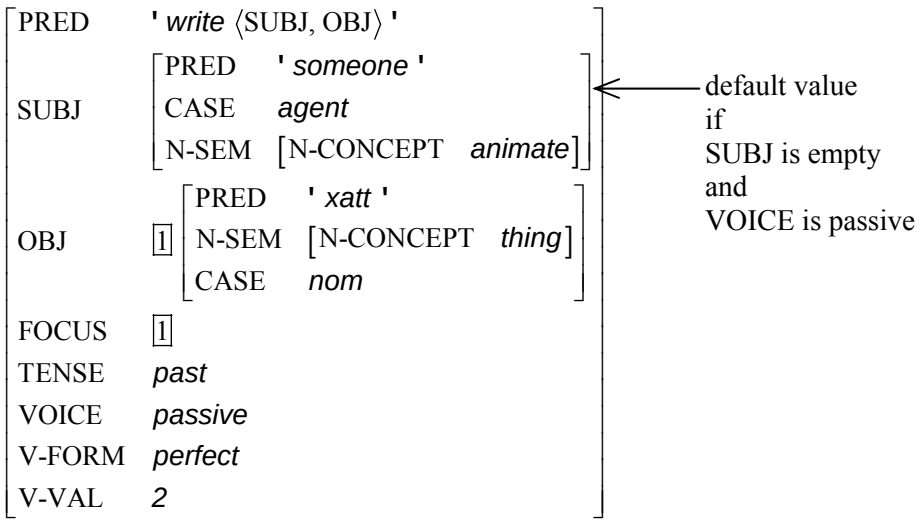


Figure 7.9: F-Structure of '*xatt (X=sey) lekh-aa ga-yaa*'

This work analyzes passive by assuming that there is no change in the verb argument structure, as shown in Figure 7.9, the FOCUS attribute points to OBJ and a default SUBJ is assumed if it is omitted in a passive sentence. More evidence is found if we make negative of the passive sentence (137) shown in (139) and another negative of a passive is shown in (140). These examples show the inability of an agent marked with '*sey*' to perform an action.

(139) خط لڑکے سے لکھا نہیں گیا

xatt	*laRk-ey=sey*	*lekh-aa*	*ga-yaa*
letter.*sg.masc=nom*	boy-*sg.masc=agentive*	write-*perf.sg.masc*	go-*perf.sg.masc*

A boy was not able to write a letter.

(140) لڑکے سے کھانا کھایا نہیں جاتا

laRk-ey=sey	*khaanaa*	*khaa-yaa*	*naheeN*	*jaa-taa*
boy-*sg.masc=agent*	food.*sg.masc=nom*	eat-*perf.sg.masc*	not	go-*perf.sg.masc*

The boy is not able to eat food

(141) *sey*
```
(↑ CASE) = agent
(↑ N-SEM N-CONCEPT) =c animate
((SUBJ ↑) V-VAL) =c 2
{
   ((SUBJ ↑) NEG) =c +
   ((SUBJ ↑) TNS-ASP MOOD) =c inability
|
   ((SUBJ ↑) TNS-ASP VOICE) =c passive
}
(SUBJ ↑)
```

There is another agentive form of animate noun that appears in the argument structure of causative verb forms, where noun marked with '*sey*' appears as an agent, which will be discussed in more detail in section 7.6. The LFG based lexical entry for case marker '*sey*' is shown in (141). The first line of lexical entry for '*sey*' marks the CASE as 'agentive'. The second line puts a constraint that noun semantic concept is animate. The third line puts the constraint on verb valency to be 2, which means that this case is assigned to transitive verbs. The lines 4 to 9 constrain that sentences should be passive voice. The last line tells that this entry is to become the SUBJ of the outer predicate.

7.4.2 Participant Case

Some verbs represent a reciprocal activity, which is performed mutually between two (or more) animate and/or human subject and objects. In these activities, the presence of each participants is needed to perform the activity. The case marker '*sey*' is used to mark animate participating nouns for grammatical 'object' position in the verb's argument structure. Here the marked noun is undergoer or experiencer of the action involved and thus occupies object position. The example sentences are shown in (143), (144), (145) and (146). Again, it is the argument structure of the verbs, which requires object marked with case marker '*sey*', instead of nominative or accusative case. In these examples, the verb is neither causative nor it is in the passive mode. The verb's argument structure requires 'ergative case' for subject and 'participant case' for object. This case is usually translated in English as a prepositional phrase employing 'with' or 'from' as a preposition.

(142) talk
```
(↑ PRED)='baat kar-naa<SUBJ, OBJ>'
(↑ SUBJ CASE) =c ergative
(↑ OBJ CASE) =c participant
```

(143) حامد نے حمید سے بات کی

Haamed=ney	*Hameed=sey*	*baat*	*k-ee*
Hamid=*erg*	Hameed=*participant*	talk=*nom*	do.*perf.sg.fem*

Hamid talked with Hameed.

(144) حامد نے حمید سے مدد لی

Haamed=ney *Hameed=sey* *madad* *l-ee*
Hamid=*erg* Hameed=*participant* help=*nom* take-*perf.sg.fem*
Hamid took help from Hameed.

(145) حامد نے حمید سے وعدہ کیا

Haamed=ney *Hameed=sey* *waAdah* *kee-aa*
Hamid=*erg* Hameed=*participant* promise=*nom* do-*perf.sg.masc*
Hamid 'did a promise' with Hameed.

(146) حامد نے حمید سے سوال پوچھا

Haamed=ney *Hameed=sey* *sawaal* *poochh-aa*
Hamid=*erg* Hameed=*participant* question=*nom* ask-*perf.sg.masc*
Hamid asked a question from Hameed.

(147) *sey*
```
(↑ CASE) = participant
((OBJ ↑) SUBJ N-SEM N-CONCEPT) =c animate
((OBJ ↑) OBJ N-SEM N-CONCEPT) =c animate
(OBJ ↑)
```

The lexical entry for the 'participant case' is shown in (147). The first line of lexical entry assigns the case as 'participant'. The second line puts constraint on SUBJ that it should be an animate noun. The third line puts constraint on OBJ that it should be an animate noun. Therefore, in participant case, both the subject and the object nouns are animate. The last line tells that noun marked as 'participant case' using '*sey*' will become the object of the predicate. The f-structure of sentence (143) is shown in Figure 7.10.

$$\begin{bmatrix} \text{PRED} & \text{' } \textit{baat karnaa } \langle \text{SUBJ, OBJ} \rangle \text{ '} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{Haamed} \text{ '} \\ \text{CASE} & \textit{erg} \\ \text{N-SEM} & \begin{bmatrix} \text{N-CONCEPT} & \textit{animate} \\ \text{N-CLASS} & \textit{proper} \end{bmatrix} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{Hameed} \text{ '} \\ \text{CASE} & \textit{participant} \\ \text{N-SEM} & \begin{bmatrix} \text{N-CONCEPT} & \textit{animate} \\ \text{N-CLASS} & \textit{proper} \end{bmatrix} \end{bmatrix} \\ \text{TENSE} & \textit{past} \\ \text{V-FORM} & \textit{perfect} \\ \text{V-VAL} & \textit{2} \end{bmatrix}$$

Figure 7.10: F-Structure of '*Haamed=ney Hameed=sey baat k-ee*'

7.4.3 Instrumental Case

For the inanimate nouns (or noun phrases) known as the instrumental nouns in Urdu: 'اسم آلہ' *aesm-e-aalah*, marked with case marker 'سے', *sey*, are categorized as 'instrumental case'. For 'instrumental case' the nouns are inanimate and classified as instrumental nouns. These are typically used by some agent or actor as an aid to accomplish some task. Example sentences are given in (148) and (149). The noun phrases in 'instrumental case' are oblique grammatical functions and sometimes act as adjunct to a sentence. This case is usually translated in English as a prepositional phrase employing 'with' as a preposition.

(148) لڑکے نے پنسل سے خط لکھا

laRk-ey=ney	*pensel=sey*	*xatt*	*lekh-aa*
boy-*sg.masc=erg*	pencil.*sg.fem=inst*	letter	write-*perf.sg.masc*

A boy wrote a letter with the pencil

(149) ماں نے چھری سے سیب کاٹا

maaN=ney	*chhoor-ee=sey*	*seyb*	*kaat-aa*
mother-*sg.fem=erg*	knife-*sg.fem=inst*	apple=*nom*	cut-*perf.sg.masc*

The mother cut the apple with the knife

(150) *sey*

```
(↑ CASE) = instrumental
(↑ N-SEM N-CONCEPT) =c instrument
(OBL-inst ↑)
```

LFG based lexical entry for 'instrumental case' is shown in (150), which assigns the case of mother noun phrase as 'instrumental'. The constraint is applied such that only those nouns that have semantic concept as 'instrument' will be assigned this case. The last line describes that instrumental case fills the oblique argument of the verb's argument structure. The f-structure of sentence (149) for instrumental case is shown in Figure 7.11.

(151) *sey*

```
(↑ CASE) = instrumental
(↑ N-SEM N-CONCEPT) =c instrument
(↑ PRED) = 'sey<(↑ OBJ)>'
(↑ P-CASE) = sey
(ADJUNCT ($) ↑)
```

If the verb argument structure does not allow an instrument, then the instrumental phrase will be treated as an adjunct using the lexical entry shown in (151). The lexical entry makes the instrumental noun phrase the object of postposition '*sey*' in the line 3. In the line 4, the value of postpositional case attribute is oblique instrumental. The line 5 makes the postpositional phrase an adjunct to main predicate.

$$\begin{bmatrix}
\text{PRED} & \text{' } \textit{kaatnaa} \langle \text{SUBJ, OBJ, OBL}_{\text{INST}} \rangle \text{ '} \\
\text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{maaN} \text{ '} \\ \text{CASE} & \textit{erg} \\ \text{N-SEM} & [\text{N-CONCEPT} \quad \textit{animate}] \end{bmatrix} \\
\text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{seyb} \text{ '} \\ \text{CASE} & \textit{nom} \\ \text{N-SEM} & [\text{N-CONCEPT} \quad \textit{thing}] \end{bmatrix} \\
\text{OBL}_{\text{INST}} & \begin{bmatrix} \text{PRED} & \text{' } \textit{chhooree} \text{ '} \\ \text{CASE} & \textit{instrumental} \\ \text{N-SEM} & [\text{N-CONCEPT} \quad \textit{instrument}] \end{bmatrix} \\
\text{TENSE} & \textit{past} \\
\text{V-FORM} & \textit{perfect} \\
\text{V-VAL} & \textit{3}
\end{bmatrix}$$

Figure 7.11: F-Structure of '*maaN=ney chhoor-ee=sey seyb kaat-aa*'

7.4.4 Travel Cases

The verbs that depict activity related to movement or travel. These require various inanimate noun (or noun phrase), marked with case marker 'سے', *sey*, to convey information about 'transportation means'/ 'vehicle', 'path'/ 'passage' or 'source location'. The sentence (152) shows example, where someone traveled by boarding on some vehicle, the noun representing vehicle is marked with case marker '*sey*'. If someone travels 'on foot' without a vehicle, then no case marker or postposition is required with the noun '*paydal*' as shown in (153). The sentence in (154) describes a *path* and in (155) describes a *passage* followed in a journey.

(152) اس نے جہاز سے سفر کیا

aos=ney	*jahaaz=sey*	*safar*	*kee-aa*
He/She-*sg=erg*	plane.*sg.masc=vehicle*	travel.*sg.masc*	go-*perf.sg.masc*

He/She traveled by a plane

(153) اس نے پیدل سفر کیا

aos=ney	*paydal*	*safar*	*kee-aa*
He/She-*sg=erg*	on foot.*sg.masc*	travel.*sg.masc*	go-*perf.sg.masc*

He/She traveled on foot.

(154) اس نے سڑک سے سفر کیا

aos=ney	*saRak=sey*	*safar*	*kee-aa*
He/She-*sg=erg*	road.*sg.masc=path*	travel.*sg.masc*	do-*perf.sg.masc*

He traveled by a road

(155) وہ دروازے سے کمرے میں آئی

woh	*darwaaz-ey=sey*	*kamrey=meyN*	*aa-ee*
He/She-*sg=nom*	door-*obl.sg.m=passage*	room=*loc.in*	come-*perf.sg.fem*

She came to room through the door

The nouns representing 'space' in Urdu are known as spatial nouns, ' اسم ظرف مکاں' *aesm-e-Zarf-e-makaN*, and when these accompany marker '*sey*', they represents *source location* as shown in (156) and (157).

(156) وہ لاہور سے آیا ہے

woh	*laahaor=sey*	*aa-yaa*	*hay*
He/She-*sg=nom*	Lahore=*source*	come-*perf.sg.masc*	be.*pres*

He has come from Lahore.

(157) تیل زمین سے نکلتا ہے

teyl	*zameen=sey*	*nekal-taa*	*hay*
oil-*sg-masc=nom*	earth=*source*	come out-*repeat.sg.masc*	be.*pres*''

'The oil comes out from earth'. The oil is taken out from underground.

The Urdu cases, which describe travel or transport, and sometimes represent path, passage or source as a location, have been described in the above mentioned examples. These cases are usually translated in English with different prepositional phrases depending upon the usage of noun concept as summarized below in the form of a short table.

Noun Concept	Noun Case	English preposition
vehicle	conveyor	by
path	locative.path	by
passage	locative.passage	through
source	locative.source	from

7.4.5 Temporal Case

Temporal nouns, in Urdu known as *aesm-e-zzarf-e-zamaN* 'اسم ظرف زماں' refer to 'time' or 'duration', and when these accompany marker '*sey*', they represents temporal case as shown in (158), (159) and (160). These cases are usually translated in English as a prepositional phrase by using 'since' and 'for' as a preposition.

(158) وہ صبح سے مقالہ لکھ رہا ہے

woh	*SobaH=sey*	*maqaalah*	*lekh rahaa hay*
He/She-*sg=nom*	morning=*temporal*	paper=*nom*	write.*root.sg.masc.cont.pres*

He is writing a paper since morning

(159) وہ دو دن سے تمہارا انتظار کر رہی ہے

woh	*dao den=sey*	*tomhaaraa*	*aentezzaar kar rahee hay*
He/She-*sg=nom*	two days=*temporal*	your=*nom*	wait.*root.sg.fem.cont.pres*

She has been waiting for you for two days.

(160) وہ مدت سے بیمار ہے

woh	*modat=sey*	*beemaar*	*hay*
He/She-*sg=nom*	long=*temporal*	ill=*nom*	be.*pres*

He/She is ill since long.

(161) *sey*
```
(↑ CASE) = temporal
(↑ N-SEM N-CONCEPT) =c temporal
(↑ PRED) = 'sey<(↑ OBJ)>'
(↑ P-CASE) = sey
(ADJUNCT ($) ↑)
```

A LFG based lexical entry is shown in (161) which assigns temporal case only to those nouns that bear temporal characteristics. The f-structure of sentence (158) is shown in Figure 7.12 for a temporal case, where temporal noun phrase is added as an adjunct to the f-structure.

$$\begin{bmatrix} \text{PRED} & \text{' } \textit{lekhnaa} \ \langle \text{SUBJ, OBJ} \rangle \text{ '} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{pro} \text{ '} \\ \text{CASE} & \textit{erg} \\ \text{PERS} & \textit{3rd} \\ \text{NUM} & \textit{sg} \\ \text{N-SEM} & [\text{N-CONCEPT} \quad \textit{animate}] \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{maqaalah} \text{ '} \\ \text{CASE} & \textit{nom} \\ \text{N-SEM} & [\text{N-CONCEPT} \quad \textit{thing}] \end{bmatrix} \\ \text{ADJUNCT} & \left\{ \begin{bmatrix} \text{PRED} & \text{' } \textit{sey} \ \langle \text{OBJ} \rangle \text{ '} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{SobaH} \text{ '} \\ \text{CASE} & \textit{temporal} \\ \text{N-SEM} & [\text{N-CONCEPT} \quad \textit{temporal}] \end{bmatrix} \end{bmatrix} \right\} \\ \text{TENSE} & \textit{present} \\ \text{ASPECT} & \textit{progressive} \\ \text{V-VAL} & \textit{2} \end{bmatrix}$$

Figure 7.12: F-Structure of '*woh SobaH=sey maqaalah lekh rahaa hay*'

7.4.6 Adverbial Case

Adverbs add some information to a verb. In English adverbs could be formed morphologically from nouns such as hurriedly, carefully, and attentively. However, in Urdu to form an adverbial phrase from a noun, the marker '*sey*' is used with nouns, normally with those nouns that represent various 'concepts'. Some examples of adverbial phrases in Urdu are shown in sentences (162), (163) and (164). These are normally translated in English using an adverb and alternately these can be translated using prepositions such as 'in a hurry', 'with keenness' and 'with attention' instead of adverbs 'hurriedly', 'keenly' and 'attentively'. The lexical entry for 'adverbial' case is shown in (165). The entry has a constraint that adverbial case can be marked with marker '*sey*' only for nouns representing concepts. The adverbial phrase is added to the set of adjuncts.

(162) وہ جلدی سے سکول پہنچی

woh	*jaldee=sey*	*sakool*	*pohanch-ee*
He/She-*sg=nom*	hurriedly=*adverbial*	school	reach-*perf.sg.fem*

She reached school hurriedly.

(163) ظفر شوق سے سبق پڑھتا ہے

Zafar	*shaoq=sey*	*sabaq*	*paRh-taa*	*hay*
Zafar-*sg.m=nom*	keenly=*adverbial*	lesson	read-*repeat.sg.m*	be=*pres*

Zafar reads the lesson keenly.

(164) مظفر توجہ سے کارٹون دیکھتا ہے

mozzafar	*tawajah=sey*	*caartoon*	*dekh-taa*	*hay*
Mozafar-*sg.m=nom*	attentively=*adverbial*	cartoon	watch-*repeat.sg.m*	be=*pres*

Mozafar watches cartoons attentively.

(165) *sey*

```
(↑ CASE) = adverbial
(↑ N-SEM N-CONCEPT) =c concept
(↑ PRED) = 'sey<(↑ OBJ)>'
(↑ P-CASE) = sey
(ADJUNCT ($) ↑)
```

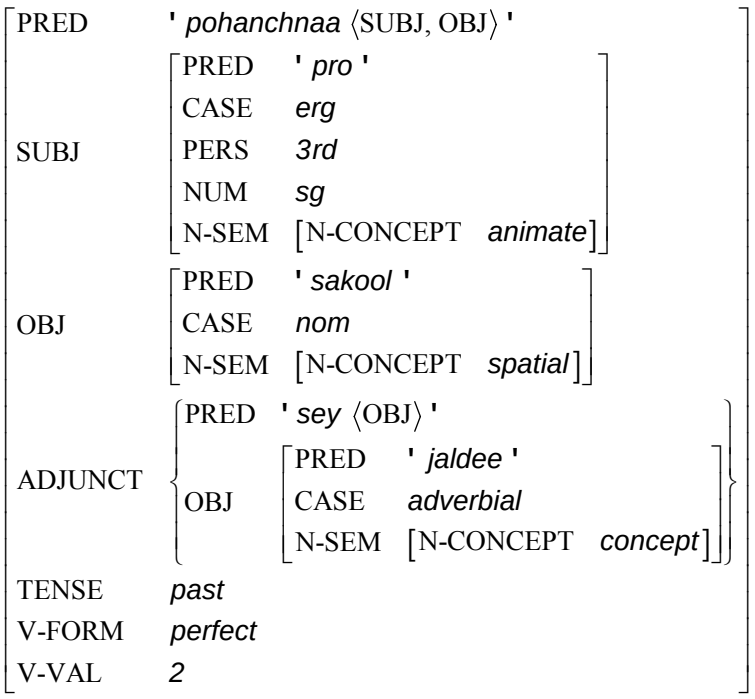


Figure 7.13: F-Structure of '*woh jaldee=sey sakool pohanchee*'

7.4.7 Infinitive Case

In an infinitive case, the Urdu infinitives (also called 'verbal nouns') are marked with '*sey*' and sometimes with other markers. Some example sentences of infinitives marked with '*sey*' are shown in (166) to (168). These phrases are normally translated

in English by using an infinitive (to + verb) or a prepositional phrase using English gerund form (–ing). The LFG based lexical entry for infinitive case is shown in (169).

(166) اسے پڑھنے سے نفرت ہے

aosey	*paRh-ney=sey*	*nafrat*	*hay*
He/She=*acc/dat*	read-*inf.obl.m=inf*	hatred=*nom*	be.*pres*

He/She has hatred for reading.

(167) مجھے گرنے سے چوٹ لگی

mojhey	*ger-ney=sey*	*chaoT*	*lag-ee*
I=*acc/dat*	fall-*inf.obl.m=inf*	injury.*sg.fem=nom*	touch-*perf.sg.fem*

I got injury from falling.

(168) میں نے کامران کو بولنے سے منع کیا

mayN=ney	*kaamraan=kao*	*baol-ney=sey*	*manA keeaa*
I=*erg*	Kamran=*acc*	injury.*sg.fem=inf*	forbid-*nom*

I prohibited Kamran from speaking.

$$\begin{bmatrix}
\text{PRED} & \text{' } \textit{manA karnaa} \langle \text{SUBJ, OBJ, OBL}_{\text{INF}} \rangle \text{ '} \\
\text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{pro} \text{ '} \\ \text{CASE} & \textit{erg} \\ \text{PERS} & \textit{first} \\ \text{NUM} & \textit{sg} \\ \text{N-SEM} & [\text{N-CONCEPT} \quad \textit{animate}] \end{bmatrix} \\
\text{OBJ} & \boxed{1} \begin{bmatrix} \text{PRED} & \text{' } \textit{kaamran} \text{ '} \\ \text{CASE} & \textit{acc} \\ \text{N-SEM} & [\text{N-CONCEPT} \quad \textit{animate}] \end{bmatrix} \\
\text{OBL}_{\text{INF}} & \begin{bmatrix} \text{PRED} & \text{' } \textit{bolnaa} \langle \text{SUBJ} \rangle \text{ '} \\ \text{CASE} & \textit{infinitive} \\ \text{P-CASE} & \textit{sey} \\ \text{N-SEM} & [\text{N-CONCEPT} \quad \textit{infinitive}] \\ \text{SUBJ} & \boxed{1} \end{bmatrix} \\
\text{TENSE} & \textit{past} \\
\text{V-VAL} & \textit{3}
\end{bmatrix}$$

Figure 7.14: F-Structure of '*mayN=ney kaamraan=kao baolney=sey manA keeaa*'

(169) *sey*

```
(↑ CASE) = infinitive
(↑ PRED) = 'sey<(↑ OBJ)>'
(↑ P-CASE) = sey
(ADJUNCT ($) ↑)
```

7.4.8 Comparison Case

The marker '*sey*' is also used in Urdu for the comparison between two noun phrases in a declarative or indicative mood. Two examples of such cases are shown in

(170) and (171). The LFG based Lexical Entry is shown in (172), which uses a constraint to check that the semantic concept of two nouns being compared is the same. The dissimilar nouns may not be compared.

(170) یہ جوتا اس سے بہتر ہے

yeh	*jootaa*	*aos=sey*	*behtar*	*hay*
this=*pro*	shoe=*nom*	that.*pro=comp*	better	AUX.*pres*

This shoe is better than that (shoe).

(171) ظفر مظفر سے لمبا ہے

Zafar	*mozzafar=sey*	*lambaa*	*hay*
Zafar=*nom*	Mozafar=*comp*	taller	AUX.*pres*

Zafar is taller than Mozafar.

(172) *sey*

```
(↑ CASE) = comparison
((OBJ ↑) SUBJ N-SEM N-CONCEPT) =
  ((OBJ ↑) OBJ N-SEM N-CONCEPT
(OBJ ($) ↑)
```

7.5 Possession Markers

The possession markers define a possessor and a possessee relationship between two noun phrases. The possessive markers require that the first noun (or noun phrase) is in 'oblique' form and require number and gender agreement with the second noun (or noun phrase). The possessive noun phrases, therefore, require two nouns (or noun phrases) one each on left and right side of the marker, as shown in noun phrases (173) , (174) and (175).

(173) NP لڑکے کی کتاب

laRk-ey	*k-ee*	*ketaab*	
boy-*sg.obl.masc*		PM-*sg.fem*	book.*sg.fem*

(174) NP گاڑی کا تالا

gaaR-ee	*k-aa*	*taal-aa*
car-*sg.fem*	PM-*sg.masc*	lock.*sg.masc*

(175) NP گاڑی کے تالے

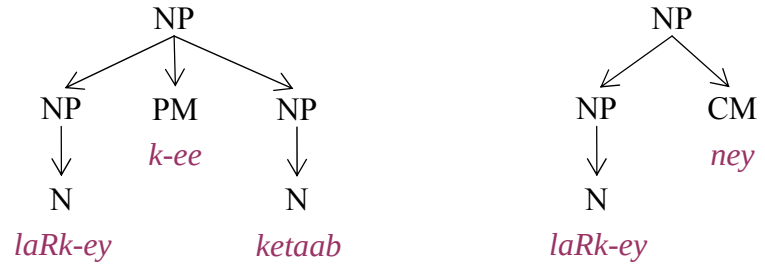
gaaR-ee	*k-ey*	*taal-ey*
car-*sg.fem*	PM-*pl.masc*	lock.*pl.masc*

(176) NP * لڑکے کی

laRk-ey	*k-ee*
boy-*sg.obl.masc*	PM-*sg.fem*

(177) NP * گاڑی کا

gaaR-ee	*k-aa*
car-*sg.fem*	PM-*sg.masc*



(a) Possession Marker (PM) (b) Case Marker (CM)

Figure 7.15: Possession Marker versus Case Marker

Figure 7.15 shows phrase structures of 'possession marker' (PM) and 'case marker' (CM). To make a well-formed noun phrase, a possession-marker requires two noun phrases both on the left and on the right side of a possession-marker, while a case-marker just requires a noun phrase ahead of itself. Using a possessive marker as a case-marker results in phrases like the one shown in (176) and (177), which cannot be used at a place where a noun phrase is required. Such phrases are incomplete 'noun phrases' and need another noun phrase for the completion. In other words, 'possessive marker' has valency for combining with two noun phrases, while 'case marker' has valency for combining with one noun phrase. Figure 7.16 shows HPSG based lexical entries of possessive markers '*kaa*', '*kee*' and '*key*'.

(178)

kaa
```
(↑ PRED) = 'kaa<POSSESSOR, POSSESSEE>'
(↑ NUM) =c sg
(↑ GEND) =c masc
```

kee
```
(↑ PRED) = 'kee<POSSESSOR, POSSESSEE>'
(↑ GEND) =c fem
```

key
```
(↑ PRED) = 'key<POSSESSOR, POSSESSEE>'
{(↑ NUM) =c pl
    (↑ GEND) =c masc
 |  (↑ N-FORM) =c oblique
}
```

The LFG based lexical entries for Urdu possession markers '*kaa*', '*kee*', and '*key*' are shown in (178), each of which require a 'possessor' noun phrase and a 'possessee' noun phrase in the argument structure with associated constraints. The LFG based phrase structure rule is shown in (179), which can be used recursively. The LFG based rule checks that first noun phrase form is oblique. The first NP is followed by a PM. The second NP assigns all of its characteristics, such as, the number, gender, case, form and other semantic properties, to the mother NP. Figure 7.17 shows f-structure of a possessive noun phrase.

(a)
$$\begin{bmatrix} word \\ \text{PHON} \quad \langle kaa \rangle \\ \text{HEAD} \quad \begin{bmatrix} possessionmarker \\ \text{AGR} \quad \boxed{1}\begin{bmatrix} \text{NUM} & sg \\ \text{GEND} & masc \end{bmatrix} \end{bmatrix} \\ \text{VAL} \quad \begin{bmatrix} \text{SPR} \quad \left\langle \begin{bmatrix} \text{NP} \\ \text{FORM} \quad obl \end{bmatrix} \right\rangle \\ \text{COMPS} \quad \left\langle \begin{bmatrix} \text{NP} \\ \text{AGR} \quad \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

(b)
$$\begin{bmatrix} word \\ \text{PHON} \quad \langle kee \rangle \\ \text{HEAD} \quad \begin{bmatrix} possessionmarker \\ \text{AGR} \quad \boxed{1}\begin{bmatrix} \text{NUM} & sg \mid pl \\ \text{GEND} & fem \end{bmatrix} \end{bmatrix} \\ \text{VAL} \quad \begin{bmatrix} \text{SPR} \quad \left\langle \begin{bmatrix} \text{NP} \\ \text{FORM} \quad obl \end{bmatrix} \right\rangle \\ \text{COMPS} \quad \left\langle \begin{bmatrix} \text{NP} \\ \text{AGR} \quad \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

(c)
$$\begin{bmatrix} word \\ \text{PHON} \quad \langle key \rangle \\ \text{HEAD} \quad \begin{bmatrix} possessionmarker \\ \text{AGR} \quad \boxed{1}\begin{bmatrix} \text{NUM} & pl \\ \text{GEND} & masc \end{bmatrix} \end{bmatrix} \\ \text{VAL} \quad \begin{bmatrix} \text{SPR} \quad \left\langle \begin{bmatrix} \text{NP} \\ \text{FORM} \quad obl \end{bmatrix} \right\rangle \\ \text{COMPS} \quad \left\langle \begin{bmatrix} \text{NP} \\ \text{AGR} \quad \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

Figure 7.16: HPSG based Lexical Entries of Urdu Possession Markers (a) '*kaa*', (b) '*kee*' and (c) '*key*'

(179) NP → NP PM NP

(↓ N-FORM) =c *oblique*

(↑ NUM) = (↓ NUM)
(↑ GEND) = (↓ GEND)
(↑ CASE) = (↓ CASE)
(↑ N-FORM) = (↓ N-FORM)
(↑ N-SEM) = (↓ N-SEM)

$$\begin{bmatrix}
\text{PRED} & \text{' } \textit{kee} \ \langle \text{POSSESSOR, POSSESSEE} \rangle \text{ '} \\
\text{POSSESSOR} & \begin{bmatrix} \text{PRED} & \text{' } \textit{laRkaa} \text{ '} \\ \text{CASE} & \textit{nom} \\ \text{N-FORM} & \textit{oblique} \\ \text{N-SEM} & [\text{N-CONCEPT} \quad \textit{animate}] \end{bmatrix} \\
\text{POSSESSEE} & \begin{bmatrix} \text{PRED} & \text{' } \textit{ketaab} \text{ '} \\ \text{CASE} & \textit{nom} \\ \text{GEND} & \textit{fem} \\ \text{NUMB} & \textit{sg} \\ \text{N-FORM} & \textit{nom} \\ \text{N-SEM} & [\text{N-CONCEPT} \quad \textit{thing}] \end{bmatrix} \\
\text{CASE} & \textit{nom} \\
\text{GEND} & \textit{fem} \\
\text{NUMB} & \textit{sg} \\
\text{N-FORM} & \textit{nom} \\
\text{N-SEM} & [\text{N-CONCEPT} \quad \textit{thing}]
\end{bmatrix}$$

Figure 7.17: F-Structure of the NP '*laRkey kee ketaab*'

7.6 Argument Structure of Causatives Verbs

The Urdu and Hindi languages are known to have a morphological causative formation in contrast to English language, which engages idiomatic use of verbs like 'make', 'get', 'have', 'help' or 'let' for representing causative structures. The causative verb forms (or transitivitized verb forms) in Urdu are normally derived from intransitive and transitive verb-root-forms by adding suffixes: *–aa*, *–waa*. Adding these suffixes to root-form of a verb forms the stems of new verbs. These stems are morphologically productive like verb roots, which have been described in Chapter 4 on the verb morphology. It is assumed in the analysis presented here that the causativization is normally a valency increasing morphological process in Urdu, which changes not only the argument structure of the verb but also the meanings conveyed. The formation of higher valency causative argument structure from the univalent and bivalent verbs can be seen in the examples presented in this section.

The example (180) shows a univalent verb '*ger-naa*' (to fall), which requires an unergative subject. The causative form 1 of the verb is '*ger-aa-naa*' (to make someone fall), which is a bivalent verb as shown in (181). It requires an ergative agent for perfect verb form and nominative agent otherwise. The verb '*ger-aa-naa*' requires accusative object if the object is 'animate' and nominative object otherwise. The causative form 2 of the verb is '*ger-waa-naa*' (to make someone fall through someone), which is a trivalent verb as shown in (182).

(180) حامد گرا

Haamed *ger-aa*
Hamid.*sg.m=nom* fall.*perf.sg.m*
Hamid fell (down).

(181) حمید نے حامد کو گرایا

Hameed=ney *Haamed=kao* *ger-aa-yaa*
Hameed.*sg.m=erg* Hamid.*sg.m=acc* fall-make.*caus1.perf.sg.m*
Hameed caused Hamid fall (down).

(182) حمید نے حامد کواحمد سے گروایا

Hameed=ney *Haamed=kao* *aeHmad=sey* *ger-waa-yaa*
Hameed=*erg* Hamid=*acc* Ahmad=*agent* fall-make.*caus2.perf.sg.m*
Hameed engaged Ahmad to cause Hamid fall (down).

(183) حمید نے حامد کو گروایا

Hameed=ney *Haamed=kao* *(X=sey)* *ger-waa-yaa*
Hameed=*erg* Hamid=*acc* (X=*agent*) fall-make.*caus2.perf.sg.m*
Hameed engaged someone to cause Hamid fall (down).

It is normally argued that the 'intermediate agent' marked with '*sey*' is optional and even after semantically recognizing the presence of an 'intermediate' or 'logical' agent, it is assumed that the presence of an 'intermediate agent' not dictated by the verb argument structure because it is syntactically optional (Mohanan 1990; Bhatt and Embick 2003; Butt 2003). However, this work assumes the following:

1. The 'intermediate agent' marked with '*sey*' is governed by the argument structure of the causative verb form 2.
2. The 'intermediate agent' marked with '*sey*' is *not optional*, however, it is sometimes *omitted* due to the reason that either the 'intermediate agent' is already known in a discourse, requires least focus or cannot be precisely stated.

This work presents the following arguments to support the above stated assumptions:

1. The 'intermediate agent' marked with '*sey*' cannot be used with causative verb form 1. The use of an 'intermediate agent' is syntactically wrong, because it does not act as a normal adjunct.
2. If the 'intermediate agent' marked with '*sey*' is omitted, then it is semantically implied. Because, if two sentences have the same words with the same syntactic structures, such that one employs causative verb form 1 and the other uses causative verb form 2, then the interpretation of the two

sentences should be different. For example, if the sentence in (181) is compared with the sentence in (183), the different interpretations are seen, because the announcement of the 'intermediate agent' is embedded in causative form 2, and these semantics could be observed in similar sentence pairs.

3. The 'intermediate agent' marked with '*sey*' when used with causative verb form 2, does not add extra meaning to interpretation but only gives the information about the 'intermediate agent'. In (183), the 'intermediate agent' is omitted and the interpretation is 'Hameed caused Hamid fall down, *through someone*', but in (182) the interpretation is more specific about 'intermediate agent' that 'Hameed caused Hamid fall down, *through Ahmad*'.
4. Omitting a syntactic unit is not a new concept. It is well known that Urdu and Hindi are 'pro-drop' languages, i.e., sometimes these languages can make a sentence without a noun (or a pronoun), if the noun (or noun phrase) could be semantically implied in a discourse.

The negative sentences employing causative form 1 and 2 in (184) and (185), similar to those given in (181) and (183) have complementary interpretation. The interpretation for example in (184) is that it is not Hameed who made Hamid fall down, but he might have engaged someone to do this task. In example (185), which uses causative form 2 and omits the phrase marked with '*sey*', the interpretation is 'Hameed did not engage *any* 'intermediate agent' to cause Hamid fall down', however he himself might have done so. While the interpretation in (186) is 'Hameed *did not engage Ahmad* to make Hamid fall down, although he *might have engaged someone else* to cause Hamid fall down.'

(184) حمید نے حامد کو نہیں گرایا
Hameed=ney *Haamed=kao* *ger-aa-yaa*
Hameed.*sg.m=erg* Hamid.*sg.m=acc* fall-make.*caus1.perf.sg.m*
Hameed didn't cause Hamid fall (down).

(185) حمید نے حامد کو نہیں گروایا
Hameed=ney *Haamed=kao* *(X=sey)* *ger-waa-yaa*
Hameed=*erg* Hamid=*acc* (X=*agent*) fall-make.*caus2.perf.sg.m*
Hameed didn't engage ***anyone*** to cause Hamid fall (down).

(186) حمید نے حامد کواحمد سے نہیں گروایا
Hameed=ney *Haamed=kao* *aeHmad=sey* *ger-waa-yaa*
Hameed=*erg* Hamid=*acc* Ahmad=*agent* fall-make.*caus2.perf.sg.m*
Hameed didn't engage Ahmad to cause Hamid fall (down).

The example of a transitive verb '*son-ee*' (to listen something) is shown in the sentence (187). The examples in (188) and (189) show causative forms of the transitive verb '*son-ee*'. The causative form 1 of this verb is '*son-naa-ee*', which is trivalent and means 'to involve someone listen something, recited by the agent himself', is shown in the sentence (188). The causative form 2 of the verb is '*son-naa-ee*', which is tetravalent and means 'to involve someone listen something, recited by some intermediate agent (including electronic devices)', is shown in (189).

(187) حامد نے نظم سنی

Haamed=ney *naZam* *son-ee*
Hamid=*erg.sg.m* poem=*nom.sg.f* listen.*perf.sg.f*
Hamid listened a poem.

(188) حمید نے حامد کو نظم سنائی

Hameed=ney *Haamed=kao* *naZam* *son-aa-ee*
Hameed.*sg.m=erg* Hamid.*sg.m=acc* poem=*nom.sg.f* listen-make.*caus1.perf.sg.f*
Hameed made Hamid listen a poem (recited by Hameed).

(189) حمید نے حامد کو احمد سے نظم سنوائی

Hameed=ney *Haamed=kao* *aeHmad=sey* *naZam* *son-waa-ee*
Hameed=*erg* Hamid=*acc* Ahmad=*agent* poem=*nom* listen-make.*caus2.perf*
Hameed made Ahmad recite and made Hamid listen a poem (recited by Ahmad).

The following is a pair of intransitive and transitive verbs, which after causative formation becomes ambiguous, as the ditransitive form is phonetically very close, but have different meaning and argument structure.

بولنا	*baol-naa*, to speak	intransitive	بلوانا	*bol-waa-naa*, to make someone speak something	ditransitive
بلانا	*bolaa-naa*, to call/invite	transitive	بلوانا	*bol-waa-naa*, to make someone call someone	ditransitive

The difference of meaning and argument structure of these verbs is shown in examples (190) to (193). This shows that the verb valency is not always increased by one, it may increase by a value of 1 or 2.

(190) بچہ بولا

bach.ch-ah *baol-aa*
child.*sg.m=nom* speak.*perf.sg.m*
The child spoke.

(191) ماں نے بچے سے شیر بلوایا

maaN=ney *bach.ch-ey=sey* *sheyr* *bol-waa-yaa*
mother.*erg* child.*sg.m=agent* lion speak-*caus2-perf*
A mother caused/helped a child to speak 'lion'.

(192) بچے نے باپ کو بلایا

bach.ch-ey=ney *baap=kao* *bolaa-yaa*
child=*erg* father=*acc* call-*perf*
A child called a father.

(193) ماں نے بچے سے باپ کو بلوایا

maaN=ney *bach.ch-ey=sey* *baap=kao* *bol-waa-yaa*
mother.*erg* child-*sg.m=agent* father=*acc* summon.*caus2.perf*
A mother asked a child to call a father.

For the sentence in (191), we can say agent of action 'speak' is a child, while mother is the causer of the action. Similarly, for the sentence in (193) the agent of action 'call' is the child. Therefore, for the causative form 1 (formed by using suffix *-aa*) the causee is in 'accusative case' marked with case marker '*kao*', while for causative form 2 (formed by using suffix *-waa*) the causee is in 'agent case' marked with case marker '*sey*'. The examples (194) to (198) have been taken from (Butt and King 2002), which show that accusative case is compatible with causative form 1, while agent case is compatible with causative form 2. While using agent case with causative form 1 and using accusative case with causative form 2 is incorrect. There the case selection for the verb argument is dictated by causative form. The causative form 1, '*kat-aa-yaa*', is also sometimes used in place '*kat-waa-yaa*' to mean the same semantics, but actually it does not exist in Urdu usage, because '*kat-aa-naa*' is not compatible with agent case.

(194) انجم نے صدف کو/*سے کھانا کھلایا

anjom=ney *Saddaf=kao/*sey* *khaanaa* *khel-aa-yaa*
Anjom=*erg* Saddaf =*dat/*agent* food.*nom* eat.*caus1.perf*
Anjom made Saddaf eat food (gave Saddaf food to eat).

(195) انجم نے صدف *کو/سے پودا کٹوایا

anjom=ney *Saddaf=*kao/sey* *paodaa* *kat-waa-yaa*
Anjom=*erg* Saddaf=**acc/agent* plant.*nom* cut-*caus2-perf*
Anjom had Saddaf cut a/*the plant.

(196) انجم نے صدف کو مصالحہ چکھایا

anjom=ney *Saddaf=kao* *meSaalHah* *chakh-aa-yaa*
Anjom=*erg* Saddaf=*acc* spice=*nom* taste-*caus1-perf*
Anjom had Saddaf taste the seasoning.

(197) انجم نے صدف سے مصالحہ چکھوایا

anjom=ney *Saddaf=sey* *meSaalHah* *chakh-waa-yaa*
Anjom=*erg* Saddaf=*agent* spice=*nom* taste-*caus2-perf*
Anjom made Saddaf *had someone* taste the seasoning.
Anjom made Saddaf *had herself* taste the seasoning.

(198) انجم نے صدف کو مصالحہ چکھوایا

anjom=ney	*Saddaf=kao*	*meSaalHah*	*chakh-waa-yaa*
Anjom=*erg*	Saddaf=*acc*	spice.*nom*	taste-*caus2-perf*

Anjom *made someone* had Saddaf taste the seasoning.

There is a semantic difference in the meanings of the sentences in (196), (197) and (198). In (196), the meaning conveyed is 'Anjom presented 'gravy' to Saddaf and Saddaf tasted the seasoning'. In (197), the meaning conveyed is 'Anjom ordered (or requested) Saddaf to make seasoning tasted by someone (or by herself)'. In this case Anjom has somehow initiated the action but she is not involved directly and even she could be away from the place. In (198), the meaning conveyed is 'Anjom engaged some intermediate agent and made Saddaf taste the seasoning. It was some intermediate agent engaged by Anjom, who presented seasoning to Saddaf and Saddaf tasted it.

The argument structures of some verbs, under the assumptions made in this work, is shown in (199).

(199)

a. fall
*ger-naa***<SUBJ>**
*ger-aa-naa***<SUBJ, OBJ>**
*ger-waa-naa***<SUBJ, SUBJ2, OBJ>**

b. laugh
*hans-naa***<SUBJ>**
*hans-aa-naa***<SUBJ, OBJ>**
*hans-waa-naa***<SUBJ, SUBJ2, OBJ>**

c. taste
*chakh-naa***<SUBJ, OBJ>**
*chakh-aa-naa***<SUBJ, OBJ2, OBJ>**
*chakh-waa-naa***<SUBJ, SUBJ2, OBJ2, OBJ>**

d. eat
*khaa-naa***<SUBJ, OBJ>**
*khel-aa-naa***<SUBJ, OBJ2, OBJ>**
*khel-waa-naa***<SUBJ, SUBJ2, OBJ2, OBJ>**

The causatives of ditransitive verbs shown in (199), under the analysis presented in this work, appear as tetravalent verbs. The semantics of well-formed sentences employing these verbs, suggest the evidence for their analysis as tetravalent verbs, due to the following considerations.

1. A noun with instrument case is not optional; if it is omitted, then it is generally implied.
2. A noun with instrument case is the actual actor or agent of action performed, and therefore it is assigned the notion of an 'intermediate' agent or a 'logical' subject.

3. A noun with instrument case is not like a bare instrument, which is typically used by the agent to perform the action, and the agent is animate having capability to perform action itself.
4. A noun with ergative case engages someone (forcefully or by request) to perform an action but is not the actual actor of the action performed

Therefore the four arguments of a tetravalent verb in (200) are: (i) an ergative (or nominative) subject, (ii) an indirect subject (intermediate agent), (iii) a direct object and (iv) an indirect object in dative case. These arguments are summarized in Table 7.2.

Table 7.2: Arguments of a Tetravalent Verb (Perfective Form)

Argument	NP Case	Thematic Role
subject	ergative	causer/ initiator of the action
indirect subject	agentive	causee/ agent of the action
indirect object	dative	beneficiary of the action
object	nominative	object of the action

(200) ماں نے باپ سے بچے کو کھانا کھلوایا

maaN=ney baap=sey bach.ch-ey=kao khaanaa khel-waa-yaa
mother=*erg* father=*ag* child.*obl=dat* food.*nom* make eat.*caus.perf*
The mother caused (asked, requested) the father to give food to the child.

(201) ماں نے چمچے سے بچے کو کھانا کھلایا

maaN=ney chamchey=sey bach.ch-ey=kao khaanaa khel-aa-yaa
mother.*erg* spoon.*inst* child.*obl.dat* food.*nom* make eat.*caus.perf*
The mother gave the food to the child by using spoon, *or*
The mother made the child eat food by means of a spoon.

The sentences in (200) and (201) have four noun phrases with the same case markers, and each sentence has one verbal predicate. The tetravalent predicate, *khel-waa-yaa*, in (200), accepts all the four noun phrases as functional arguments, while the trivalent predicate, *khel-aa-yaa*, in (201), accepts only three noun phrases as functional arguments: The spoon in (201) is used as an instrument. The spoon is not animate to perform the action on its will, and therefore cannot take the position of an agent for performing the action. The mother in (201), is the actual performer of the action, making child to eat food. The spoon is used by the mother to perform the action. The instrumental argument 'spoon' is optional, and therefore it is not controlled by the predicate and acts as an adjunct. It may again be noted that the phrase '*baap=sey*', cannot be used in place of '*chamchey=sey*' in (201), however '*chamchey=sey*' can be used in (200). Figure 7.18 shows f-structure with tetravalent predicate for the sentence in (200) and Figure 7.19 shows f-structure with trivalent

predicate for the sentence in (201). The difference of 'indirect subject' SUB2 and an optional ADJUNCT can be seen in the f-structures.

$$\begin{bmatrix} \text{PRED} & \text{' } \textit{khelwaanaa} \ \langle \text{SUBJ, OBJ, OBJ2, SUBJ2} \rangle \text{ '} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{maaN} \text{ '} \\ \text{N-SEM} & [\text{N-CONCEPT} \quad \textit{animate}] \\ \text{CASE} & \textit{erg} \end{bmatrix} \\ \text{SUBJ2} & \begin{bmatrix} \text{PRED} & \text{' } \textit{baap} \text{ '} \\ \text{N-SEM} & [\text{N-CONCEPT} \quad \textit{animate}] \\ \text{CASE} & \textit{agent} \end{bmatrix} \\ \text{OBJ2} & \begin{bmatrix} \text{PRED} & \text{' } \textit{bachchah} \text{ '} \\ \text{N-SEM} & [\text{N-CONCEPT} \quad \textit{animate}] \\ \text{CASE} & \textit{dat} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{khaanaa} \text{ '} \\ \text{N-SEM} & [\text{N-CONCEPT} \quad \textit{thing}] \\ \text{CASE} & \textit{nom} \end{bmatrix} \end{bmatrix}$$

Figure 7.18: F-Structure of
'*maaN=ney baap=sey bach.ch-ey=kao khaanaa khel-waa-yaa*'

$$\begin{bmatrix} \text{PRED} & \text{' } \textit{khelaanaa} \ \langle \text{SUBJ, OBJ, OBJ2} \rangle \text{ '} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{maaN} \text{ '} \\ \text{N-SEM} & [\text{N-CONCEPT} \quad \textit{animate}] \\ \text{CASE} & \textit{erg} \end{bmatrix} \\ \text{OBJ2} & \begin{bmatrix} \text{PRED} & \text{' } \textit{bachchah} \text{ '} \\ \text{N-SEM} & [\text{N-CONCEPT} \quad \textit{animate}] \\ \text{CASE} & \textit{dat} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{khaanaa} \text{ '} \\ \text{N-SEM} & [\text{N-CONCEPT} \quad \textit{thing}] \\ \text{CASE} & \textit{nom} \end{bmatrix} \\ \text{ADJUNCT} & \left\{ \begin{bmatrix} \text{PRED} & \text{' } \textit{sey} \ \langle \text{OBJ} \rangle \text{ '} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{chamchah} \text{ '} \\ \text{N-SEM} & [\text{N-CONCEPT} \quad \textit{instrument}] \end{bmatrix} \end{bmatrix} \right\} \end{bmatrix}$$

Figure 7.19: F-Structure of
'*maaN=ney chamchey=sey bach.ch-ey=kao khaanaa khel-aa-yaa*'

Figure 7.20 shows f-structure with trivalent predicate for the sentence in (196), which has all the three required grammatical functions. However, Figure 7.21 shows f-structure with tetravalent predicate for the sentence in (197), which has three grammatical functions and the 'intermediate agent' is omitted.

It is proposed that 'intermediate agent', in the absence of an actual argument, can take a ***default*** value of '**someone**' in non-negative sentences and '**anyone**' in negative sentences. This is needed to fulfill the notion of completeness and to meet the assumption that "if intermediate agent is omitted, it is semantically implied".

It is an interesting problem to investigate that in a discourse, the 'intermediate agent' in the absence of an actual argument, may be bind to other nouns, already present in the discoure, using anaphora resolution strategies.

$$\begin{bmatrix} \text{PRED} & \text{' } \textit{chakhaanaa} \ \langle \text{SUBJ, OBJ, OBJ2} \rangle \text{ '} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{aanjom} \text{ '} \\ \text{N-SEM} & \begin{bmatrix} \text{N-CONCEPT} & \textit{animate} \\ \text{N-CLASS} & \textit{proper} \end{bmatrix} \\ \text{CASE} & \textit{erg} \end{bmatrix} \\ \text{OBJ2} & \begin{bmatrix} \text{PRED} & \text{' } \textit{Sadaf} \text{ '} \\ \text{N-SEM} & \begin{bmatrix} \text{N-CONCEPT} & \textit{animate} \\ \text{N-CLASS} & \textit{proper} \end{bmatrix} \\ \text{CASE} & \textit{dat} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{maSaalHah} \text{ '} \\ \text{N-SEM} & [\text{N-CONCEPT} \quad \textit{thing}] \\ \text{CASE} & \textit{nom} \end{bmatrix} \end{bmatrix}$$

Figure 7.20: F-Structure of '*aanjom=ney Saddaf=kao meSaalHah chakh-aa-yaa*'

$$\begin{bmatrix} \text{PRED} & \text{' } \textit{chakhwaanaa} \ \langle \text{SUBJ, SUBJ2, OBJ, OBJ2} \rangle \text{ '} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{aanjum} \text{ '} \\ \text{N-SEM} & \begin{bmatrix} \text{N-CONCEPT} & \textit{animate} \\ \text{N-CLASS} & \textit{proper} \end{bmatrix} \\ \text{CASE} & \textit{erg} \end{bmatrix} \\ \text{OBJ2} & \begin{bmatrix} \text{PRED} & \text{'} \textit{Sadaf} \text{'} \\ \text{N-SEM} & \begin{bmatrix} \text{N-CONCEPT} & \textit{animate} \\ \text{N-CLASS} & \textit{proper} \end{bmatrix} \\ \text{CASE} & \textit{dat} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{maSaalHah} \text{ '} \\ \text{N-SEM} & [\text{N-CONCEPT} \quad \textit{thing}] \\ \text{CASE} & \textit{nom} \end{bmatrix} \\ \text{SUBJ2} & [\text{PRED} \quad \textit{'someone'}] \end{bmatrix}$$

Figure 7.21: F-Structure of '*aanjom=ney Saddaf=kao meSaalHah chakh-waa-yaa*'

The LFG based lexical entry for animate nouns is shown in (202), which assigns 'agent' case to animate nouns. If the verb valency is 2, then the noun phrase is used in a passive voice sentence or in an inability mood as a subject. In case of verbs

having causative form 2 and having valency 3 or 4, the agent case marked with '*sey*' can be used as an 'indirect subject'.

(202) *sey*

```
(↑ CASE) = agent
(^ N-SEM N-CONCEPT) =c animate
{
   ((SUBJ ↑) V-VAL) = 2
   { ((SUBJ ↑) NEG) = +
      ((SUBJ ↑) TNS-ASP MOOD) = inability
   |  ((SUBJ ↑) TNS-ASP VOICE) = passive    }
   (SUBJ ↑) }
|
{  {((SUBJ2 ↑) V-VAL) = 3
   |((SUBJ2 ↑) V-VAL) = 4}
   ((SUBJ2 ↑) V-FORM) = Caus2
   (SUBJ2 ↑) }
```

7.7 Conclusions

In this Chapter, the proposals to handle syntax of the noun phrase in Urdu have been presented. Use of semantic and verb valency features to better resolve nominative, ergative, dative and accusative cases has been suggested. Rule for possession markers is suggested. Noun semantic features also found useful for differentiating cases marked with '*sey*'. The agentive case marked with '*sey*' for animate nouns is also used to propose the concept of 'indirect subject' for the causative 2 verb forms in Urdu. A method for causative verbs in Urdu based on morphological valency alternation has been proposed (Butt and King 2006), which enables generation of new argument structure for a verb based on causative morphemes.

Chapter 8

MODELING URDU VERBAL SYNTAX BY IDENTIFYING TENSE, ASPECT AND MOOD FEATURES

A verb is a word, which is used to describe an action (doing), state (being), or occurrence (happening). A verb not only carries information about the argument-structure, but also contains information about tense, aspect, mood and voice. The argument-structure of a verb describes the number and type of phrases that may be required to make a well-formed sentence. The tense indicates the time of action, state, or occurrence in relation to the time of utterance. The aspect expresses a feature of the action without reference to time, such as completion, repetition or duration. The mood of a verb expresses a feature representing the type of an action, such as command, request, question, wish, or conditionality. The voice expresses the focus (or topic) of a sentence, e.g., in active-voice the focus is on the subject, while in the passive-voice the focus is on the object.

A verb, in some languages, uses the inflectional affixes to represent tense, aspect, and mood. In some other languages, it uses tense, aspectual and modal auxiliaries. Urdu uses both verb auxiliaries and affixes to represent tense, aspect and mood. As described in Chapter 3, a verb in Urdu can have 60 forms having different agreement features, while in English a verb has only five forms. Therefore, in Urdu, the verb-form dependency is relatively complex as compared to the dependency in English. In addition to this, in Urdu, sometimes a verb form depends on the 'gender' and 'number' of the object, and sometimes depends on the 'gender' and 'number' of the subject. Similarly, the auxiliaries also change their form to comply with various attributes.

In this Chapter, the modeling of the verbal structure in Urdu is presented by assembling tense, aspect and mood features from the verbal morphemes and auxiliaries used in a sentence. The agreement tables traditionally appear in Urdu grammars, which are presented here to gather agreement information and based on those tables the information associated with various verb morphemes and auxiliaries is collected. In this Chapter, the phrase structure rules, c-structures and f-structures are proposed to describe the tense, aspect and mood variations in Urdu language.

8.1 Urdu Verb Agreement

The verbs in Urdu require agreement with noun phrase for various attributes, such as, the 'gender', 'number', 'person', 'case' and 'honor form'. All nouns in Urdu carry 'gender' attribute, which also require agreement with the verb-forms. To show the agreement dependency involved in the tense system, traditionally, Urdu grammars show different sentence formations for a particular tense in a tabular form – normally known as a *gardaan* (گردان) or a paradigm of a tense. The present-repetitive-tense paradigm, which requires a subject-agreement, is shown in Table 8.1 and the present perfect tense paradigm, which requires an object-agreement, is shown in Table 8.2. The gender (GEND), number (NUM), person (PERS) and honor-form (H-FORM) attributes of the subject are shown in columns of a table, while gender and number variation of the object is shown in sub-tables. Urdu has honor attributes associated with second person pronouns. The pronoun '*too* – you' is used either in a *frank* manner with a friendly tone or in a *rude* speech with an impolite tone. The pronoun '*tom* – you' is a *formal* (or normal) way to talk with colleagues or with *familiar* person. The pronoun '*aap* – you' expresses *polite* mood even with younger persons or it is used as *respect*. The second person pronoun is usually a singular, however, for plural reference, phrases such as '*tom laog* – you people', '*tom saarey* – you all', and '*tom sab* – you all', '*aap laog* – you people', '*app saarey* – you all', and '*aap sab* – you all', are used. This means that '*too*' appears only as a singular pronoun, but '*tom*' and '*aap*' can be used as plural pronouns.

Table 8.1: A Present-Repetitive-Tense Paradigm for a Transitive Verb Having Subject-Agreement

(a) Singular Feminine Object, book (*ketaab* – کتاب)

Transliteration	Urdu Script	GEND	PERS	NUM	H-FORM
mayN ketaab xareedtaa haoN	میں کتاب خریدتا ہوں	masc	1st	sg	–
mayN ketaab xareedtee haoN	میں کتاب خریدتی ہوں	fem			
ham ketaab xareedtey hayN	ہم کتاب خریدتے ہیں	masc	1st	pl	–
ham ketaab xareedtee hayN	ہم کتاب خریدتی ہیں	fem			
too ketaab xareedtaa hay	تو کتاب خریدتا ہے	masc	2nd	sg	frank, rude
too ketaab xareedtee hay	تو کتاب خریدتی ہے	fem			
tom ketaab xareedtey hao	تم کتاب خریدتے ہو	masc	2nd	sg	formal, familiar
tom ketaab xareedtee hao	تم کتاب خریدتی ہو	fem			
aap ketaab xareedtey hayN	آپ کتاب خریدتے ہیں	masc	2nd	sg	polite, respect
aap ketaab xareedtee hayN	آپ کتاب خریدتی ہیں	fem			
woh ketaab xareedtaa hay	وہ کتاب خریدتا ہے	masc	3rd	sg	–
woh ketaab xareedtee hay	وہ کتاب خریدتی ہے	fem			
woh ketaab xareedtey hayN	وہ کتاب خریدتے ہیں	masc	3rd	pl	–
woh ketaab xareedtee hayN	وہ کتاب خریدتی ہیں	fem			

(b) Plural Feminine Object, books (*ketaabeyN* – کتابیں)

Transliteration	Urdu Script	GEND	PERS	NUM	H-FORM
mayN ketaabeyN xareedtaa haoN	میں کتابیں خریدتا ہوں	masc	1st	sg	–
mayN ketaabeyN xareedtee haoN	میں کتابیں خریدتی ہوں	fem			
ham ketaabeyN xareedtey hayN	ہم کتابیں خریدتے ہیں	masc	1st	pl	–
ham ketaabeyN xareedtee hayN	ہم کتابیں خریدتی ہیں	fem			
too ketaabeyN xareedtaa hay	تو کتابیں خریدتا ہے	masc	2nd	sg	frank, rude
too ketaabeyN xareedtee hay	تو کتابیں خریدتی ہے	fem			
tom ketaabeyN xareedtey hao	تم کتابیں خریدتے ہو	masc	2nd	sg	formal, familiar
tom ketaabeyN xareedtee hao	تم کتابیں خریدتی ہو	fem			
aap ketaabeyN xareedtey hayN	آپ کتابیں خریدتے ہیں	masc	2nd	sg	polite, respect
aap ketaabeyN xareedtee hayN	آپ کتابیں خریدتی ہیں	fem			
woh ketaabeyN xareedtaa hay	وہ کتابیں خریدتا ہے	masc	3rd	sg	–
woh ketaabeyN xareedtee hay	وہ کتابیں خریدتی ہے	fem			
woh ketaabeyN xareedtey hayN	وہ کتابیں خریدتے ہیں	masc	3rd	pl	–
woh ketaabeyN xareedtee hayN	وہ کتابیں خریدتی ہیں	fem			

(c) Singular Masculine Object, lock (*taalaa* – تالا)

Transliteration	Urdu Script	GEND	PERS	NUM	H-FORM
mayN taalaa xareedtaa haoN	میں تالا خریدتا ہوں	masc	1st	sg	–
mayN taalaa xareedtee haoN	میں تالا خریدتی ہوں	fem			
ham taalaa xareedtey hayN	ہم تالا خریدتے ہیں	masc	1st	pl	–
ham taalaa xareedtee hayN	ہم تالا خریدتی ہیں	fem			
too taalaa xareedtaa hay	تو تالا خریدتا ہے	masc	2nd	sg	frank, rude
too taalaa xareedtee hay	تو تالا خریدتی ہے	fem			
tom taalaa xareedtey hao	تم تالا خریدتے ہو	masc	2nd	sg	formal, familiar
tom taalaa xareedtee hao	تم تالا خریدتی ہو	fem			
aap taalaa xareedtey hayN	آپ تالا خریدتے ہیں	masc	2nd	sg	polite, respect
aap taalaa xareedtee hayN	آپ تالا خریدتی ہیں	fem			
woh taalaa xareedtaa hay	وہ تالا خریدتا ہے	masc	3rd	sg	–
woh taalaa xareedtee hay	وہ تالا خریدتی ہے	fem			
woh taalaa xareedtey hayN	وہ تالا خریدتے ہیں	masc	3rd	pl	–
woh taalaa xareedtee hayN	وہ تالا خریدتی ہیں	fem			

(d) Plural Masculine Object, locks (*taaley* – تالے)

Transliteration	Urdu Script	GEND	PERS	NUM	H-FORM
mayN taaley xareedtaa haoN	میں تالے خریدتا ہوں	masc	1st	sg	–
mayN taaley xareedtee haoN	میں تالے خریدتی ہوں	fem			
ham taaley xareedtey hayN	ہم تالے خریدتے ہیں	masc	1st	pl	–
ham taaley xareedtee hayN	ہم تالے خریدتی ہیں	fem			
too taaley xareedtaa hay	تو تالے خریدتا ہے	masc	2nd	sg	frank, rude
too taaley xareedtee hay	تو تالے خریدتی ہے	fem			
tom taaley xareedtey hao	تم تالے خریدتے ہو	masc	2nd	sg	formal, familiar
tom taaley xareedtee hao	تم تالے خریدتی ہو	fem			
aap taaley xareedtey hayN	آپ تالے خریدتے ہیں	masc	2nd	sg	polite, respect
aap taaley xareedtee hayN	آپ تالے خریدتی ہیں	fem			
woh taaley xareedtaa hay	وہ تالے خریدتا ہے	masc	3rd	sg	–
woh taaley xareedtee hay	وہ تالے خریدتی ہے	fem			
woh taaley xareedtey hayN	وہ تالے خریدتے ہیں	masc	3rd	pl	–
woh taaley xareedtee hayN	وہ تالے خریدتی ہیں	fem			

By observing the present repetitive tense paradigm shown in Table 8.1, it may be seen that the verb-form and the auxiliary-form remain the same for objects having different 'number' and/or 'gender' attributes. Therefore, the verb-form and the auxiliary-form do not require agreement with the 'number' and 'gender' of an object for the present-repetitive-tense. The verb-form and the auxiliary-form agree with the highest nominative argument of the verb, the subjects of the sentences in Table 8.1 are nominative, and therefore, require verb-subject agreement.

Table 8.2: A Present-Perfect-Tense Paradigm for a Transitive Verb Having Object-Agreement

(a) Singular Feminine Object, book (*ketaab* – کتاب)

Transliteration	Urdu Script	GEND	PERS	NUM	H-FORM
mayN ney ketaab xareedee hay	میں نے کتاب خریدی ہے	masc/ fem	1st	sg	–
ham ney ketaab xareedee hay	ہم نے کتاب خریدی ہے	masc/ fem	1st	pl	–
too ney ketaab xareedee hay	تو نے کتاب خریدی ہے	masc/ fem	2nd	sg	frank
tom ney ketaab xareedee hay	تم نے کتاب خریدی ہے	masc/ fem	2nd	sg	formal
aap ney ketaab xareedee hay	آپ نے کتاب خریدی ہے	masc/ fem	2nd	sg	polite
aes ney ketaab xareedee hay	اس نے کتاب خریدی ہے	masc/ fem	3rd	sg	–
aenhaoN ney ketaab xareedee hay	انہوں نے کتاب خریدی ہے	masc/ fem	3rd	pl	–

(b) Plural Feminine Object, books (*ketaabeyN* – کتابیں)

Transliteration	Urdu Script	GEND	PERS	NUM	H-FORM
mayN ney ketaabeyN xareedee hayN	میں نے کتابیں خریدی ہیں	masc/ fem	1st	sg	–
ham ney ketaabeyN xareedee hayN	ہم نے کتابیں خریدی ہیں	masc/ fem	1st	pl	–
too ney ketaabeyN xareedee hayN	تو نے کتابیں خریدی ہیں	masc/ fem	2nd	sg	frank
tom ney ketaabeyN xareedee hayN	تم نے کتابیں خریدی ہیں	masc/ fem	2nd	sg	formal
aap ney ketaabeyN xareedee hayN	آپ نے کتابیں خریدی ہیں	masc/ fem	2nd	sg	polite
aes ney ketaabeyN xareedee hayN	اس نے کتابیں خریدی ہیں	masc/ fem	3rd	sg	–
aenhaoN ney ketaabeyN xareedee hayN	انہوں نے کتابیں خریدی ہیں	masc/ fem	3rd	pl	–

(c) Singular Masculine Object, lock (*taalaa* – تالا)

Transliteration	Urdu Script	GEND	PERS	NUM	H-FORM
mayN ney taalaa xareedaa hay	میں نے تالا خریدا ہے	masc/ fem	1st	sg	–
ham ney taalaa xareedaa hay	ہم نے تالا خریدا ہے	masc/ fem	1st	pl	–
too ney taalaa xareedaa hay	تو نے تالا خریدا ہے	masc/ fem	2nd	sg	frank
tom ney taalaa xareedaa hay	تم نے تالا خریدا ہے	masc/ fem	2nd	sg	formal
aap ney taalaa xareedaa hay	آپ نے تالا خریدا ہے	masc/ fem	2nd	sg	polite
aes ney taalaa xareedaa hay	اس نے تالا خریدا ہے	masc/ fem	3rd	sg	–
aenhaoN ney taalaa xareedaa hay	انہوں نے تالا خریدا ہے	masc/ fem	3rd	pl	–

(d) Plural Masculine Object, locks (*taaley* – تالے)

Transliteration	Urdu Script	GEND	PERS	NUM	H-FORM
mayN ney taaley xareedey hayN	میں نے تالے خریدے ہیں	masc/ fem	1st	sg	–
ham ney taaley xareedey hayN	ہم نے تالے خریدے ہیں	masc/ fem	1st	pl	–
too ney taaley xareedey hayN	تو نے تالے خریدے ہیں	masc/ fem	2nd	sg	frank
tom ney taaley xareedey hayN	تم نے تالے خریدے ہیں	masc/ fem	2nd	sg	formal
aap ney taaley xareedey hayN	آپ نے تالے خریدے ہیں	masc/ fem	2nd	sg	polite
aes ney taaley xareedey hayN	اس نے تالے خریدے ہیں	masc/ fem	3rd	sg	–
aenhaoN ney taaley xareedey hayN	انہوں نے تالے خریدے ہیں	masc/ fem	3rd	pl	–

From the present-perfect-tense paradigm shown in Table 8.2, it is observed that 'number' and 'gender' of the object has agreement dependency in determining the verb-form and auxiliary-form. In this case, the 'number', 'person' or 'gender' of a subject is not playing any role in the agreement dependency. The subject of the sentence in these sentences is in the ergative case (instead of a nominative case), and the object is in the nominative case (instead of an accusative case), therefore the agreement is with the object. If the object is in the accusative case, depending on the argument-structure of the verb used, then in the absence of a nominative argument, the default-agreement-form (i.e., singular masculine verb-form) is required.

The examples of the present-repetitive-tense and present-perfect-tense paradigms for Urdu shown above demonstrate that the verb and auxiliary forms sometimes depend on the 'gender', 'case' and 'number' of an object and sometimes on the 'person', 'gender', 'number', 'honor form' and 'case' of a subject. This agreement requirement can be observed for most of the transitive and ditransitive verbs (i.e., verb valencies two or more). However, if a verb is intransitive (i.e., monovalent verb, which only takes a subject) the subject is normally in a nominative case even for perfect tenses, and therefore, the intransitive verb agrees with the subject for all tenses.

Table 8.3: The Pattern of the Present Repetitive Tense for an Optional Object (obj) and a Verb Root/Stem (vs)

Transliteration			Urdu Script			GEND	PERS	NUM	H-FORM
mayN	(obj) vs–*taa*	*hooN*	ہوں	تا–vs (obj)	میں	masc	1st	sg	–
mayN	(obj) vs–*tee*	*hooN*	ہوں	تی–vs (obj)	میں	fem			
ham	(obj) vs–*tey*	*hayN*	ہیں	تے–vs (obj)	ہم	masc	1st	pl	–
ham	(obj) vs–*tee*	*hayN*	ہیں	تی–vs (obj)	ہم	fem			
too	(obj) vs–*taa*	*hay*	ہے	تا–vs (obj)	تو	masc	2nd	sg	frank
too	(obj) vs–*tee*	*hay*	ہے	تی–vs (obj)	تو	fem			
tom	(obj) vs–*tey*	*hao*	ہو	تے–vs (obj)	تم	masc	2nd	sg	formal
tom	(obj) vs–*tee*	*hao*	ہو	تی–vs (obj)	تم	fem			
aap	(obj) vs–*tey*	*hayN*	ہیں	تے–vs (obj)	آپ	masc	2nd	sg	polite
aap	(obj) vs–*tee*	*hayN*	ہیں	تی–vs (obj)	آپ	fem			
woh	(obj) vs–*taa*	*hay*	ہے	تا–vs (obj)	وہ	masc	3rd	sg	–
woh	(obj) vs–*tee*	*hay*	ہے	تی–vs (obj)	وہ	fem			
woh	(obj) vs–*tey*	*hayN*	ہیں	تے–vs (obj)	وہ	masc	3rd	pl	–
woh	(obj) vs–*tee*	*hayN*	ہیں	تی–vs (obj)	وہ	fem			

Table 8.3 and Table 8.4 show pattern for the formation of the present-repetitive-tense and past-repetitive-tense respectively. In these tenses, it has been observed that the verb-form dependence is not on the object. The verb form depends on the person, number and gender of the subject. Both verb morpheme and auxiliary verb change their form to agree in 'number', 'gender', 'person' and 'honor form' with the subject.

The same sentence formation pattern can be used for the intransitive and transitive verbs. For an intransitive verb, the object is omitted from the pattern, and for a transitive verb, an object having any 'gender' and 'number' attributes can be placed. Table 8.5 shows the pattern for the formation of future tense. The agreement of the verb-form and auxiliary-form, in the future tense, is also with the gender, number and person of a nominative subject.

Table 8.4: The Pattern of the Past Repetitive Tense for an Optional Object (obj) and a Verb Root/Stem (vs)

Transliteration			Urdu Script			GEND	PERS	NUM	H-FORM
mayN	(obj) vs–*taa*	*thaa*	میں	(obj) vs–تا	تھا	masc	1st	sg	–
mayN	(obj) vs–*tee*	*thee*	میں	(obj) vs–تی	تھی	fem			
ham	(obj) vs–*tey*	*they*	ہم	(obj) vs–تے	تھے	masc	1st	pl	–
ham	(obj) vs–*tee*	*theeN*	ہم	(obj) vs–تی	تھیں	fem			
too	(obj) vs–*taa*	*thaa*	تو	(obj) vs–تا	تھا	masc	2nd	sg	frank
too	(obj) vs–*tee*	*thee*	تو	(obj) vs–تی	تھی	fem			
tom	(obj) vs–*tey*	*they*	تم	(obj) vs–تے	تھے	masc	2nd	sg	formal
tom	(obj) vs–*tee*	*theeN*	تم	(obj) vs–تی	تھیں	fem			
aap	(obj) vs–*tey*	*they*	آپ	(obj) vs–تے	تھے	masc	2nd	sg	polite
aap	(obj) vs–*tee*	*theeN*	آپ	(obj) vs–تی	تھیں	fem			
woh	(obj) vs–*taa*	*thaa*	وہ	(obj) vs–تا	تھا	masc	3rd	sg	–
woh	(obj) vs–*tee*	*thee*	وہ	(obj) vs–تی	تھی	fem			
woh	(obj) vs–*tey*	*they*	وہ	(obj) vs–تے	تھے	masc	3rd	pl	–
woh	(obj) vs–*tee*	*theeN*	وہ	(obj) vs–تی	تھیں	fem			

Table 8.5: The Pattern of the Future Tense for an Optional Object (obj) and a Verb Root/Stem (vs)

Transliteration			Urdu Script			GEND	PERS	NUM	H-FORM
mayN	(obj) vs–*ooN*	*gaa*	میں	(obj) vs–وں	گا	masc	1st	sg	–
mayN	(obj) vs–*ooN*	*gee*	میں	(obj) vs–وں	گی	fem			
ham	(obj) vs–*eyN*	*gey*	ہم	(obj) vs–یں	گے	masc	1st	pl	–
ham	(obj) vs–*eyN*	*gee*	ہم	(obj) vs–یں	گی	fem			
too	(obj) vs–*ey*	*gaa*	تو	(obj) vs–ے	گا	masc	2nd	sg	frank
too	(obj) vs–*ey*	*gee*	تو	(obj) vs–ے	گی	fem			
tom	(obj) vs–*ao*	*gey*	تم	(obj) vs–و	گے	masc	2nd	sg	formal
tom	(obj) vs–*ao*	*gee*	تم	(obj) vs–و	گی	fem			
aap	(obj) vs–*eyN*	*gey*	آپ	(obj) vs–یں	گے	masc	2nd	sg	polite
aap	(obj) vs–*eyN*	*gee*	آپ	(obj) vs–یں	گی	fem			
woh	(obj) vs–*ey*	*gaa*	وہ	(obj) vs–ے	گا	masc	3rd	sg	–
woh	(obj) vs–*ey*	*gee*	وہ	(obj) vs–ے	گی	fem			
woh	(obj) vs–*eyN*	*gey*	وہ	(obj) vs–یں	گے	masc	3rd	pl	–
woh	(obj) vs–*eyN*	*gee*	وہ	(obj) vs–یں	گی	fem			

It may also be observed from the above tables that the future and past auxiliaries are not dependant on the 'person' attribute of a subject, while present auxiliaries are

dependant on the 'person' attribute. In future tense, the 'person' attribute is marked on the verb-morpheme. In past tense, the 'person' attribute is marked neither on the verb-morpheme nor on the auxiliary. Such irregular variations in the agreement dependency for verb-morphemes and auxiliaries are shown in Table 8.6 and Table 8.7 respectively.

Table 8.6: The Dependence of Verb Morphemes for the Subject Agreement

Morpheme		Person	Number	Gender	Tense	H-Form
taa	تا	1st, 3rd	sg	masc	present, past	–
tee	تی	1st, 3rd	sg, pl	fem	present, past	–
tey	تے	1st, 3rd	pl	masc	present, past	–
taa	تا	2nd	sg	masc	present, past	frank
tee	تی	2nd	sg, pl	fem	present, past	–
tey	تے	2nd	sg, pl	masc	present, past	formal, polite
ooN	وں	1st	sg	masc, fem	future	–
ey	ے	3rd	sg	masc, fem	future	–
eyN	یں	1st, 3rd	pl	masc, fem	future	–
ey	ے	2nd	sg	masc, fem	future	frank
ao	و	2nd	sg	masc, fem	future	formal
eyN	یں	2nd	sg, pl	masc, fem	future	polite

Table 8.7: The Dependence of Auxiliary Verb for the Subject Agreement

Auxiliary		Tense	Gender	Number	Person	H-Form
hooN	ہوں	present	masc, fem	sg	1st	–
hay	ہے	present	masc, fem	sg	3rd	–
hayN	ہیں	present	masc, fem	pl	1st, 3rd	–
hay	ہے	present	masc, fem	sg	2nd	frank
hao	ہو	present	masc, fem	sg, pl	2nd	formal
hayN	ہیں	present	masc, fem	sg, pl	2nd	polite
thaa	تھا	past	masc	sg	1st, 3rd	–
they	تھے	past	masc	pl	1st, 3rd	–
thee	تھی	past	fem	sg	1st, 3rd	–
theeN	تھیں	past	fem	pl	1st, 3rd	–
thaa	تھا	past	masc	sg	2nd	frank
they	تھے	past	masc	pl	2nd	formal, polite
thee	تھی	past	fem	sg	2nd	–
theeN	تھیں	past	fem	pl	2nd	–
gaa	گا	future	masc	sg	1st, 3rd	–
gey	گے	future	masc	pl	1st, 3rd	–
gee	گی	future	fem	sg, pl	1st, 3rd	–
gaa	گا	future	masc	sg	2nd	frank
gey	گے	future	masc	sg, pl	2nd	formal, polite
gee	گی	future	fem	sg, pl	2nd	–

Table 8.6 shows the agreement of verb morphemes with reference to the person, number and gender for the present-repetitive, past-repetitive and future tenses, while

Table 8.7 shows the agreement of the auxiliary (helping) verbs with reference to the person, number and gender for the same tenses. In both of these tables, the agreement is with the nominative subject. Table 8.8 show pattern for present-perfect and past-perfect tenses. The object's gender (GEND) and number (NUM) attributes shown in columns require agreement with verb-form and auxiliary-form. The subject case should be ergative. This dependence of verb-morpheme and auxiliary-form is summarized in Table 8.9.

Table 8.8: The Pattern of the (a) Present Perfect Tense (b) Past Perfect Tense for a Subject (sub), an Object (obj) and a Verb Root/Stem (vs)

(a)

Transliteration				Urdu Script				GEND	NUM
sub	obj	vs–*aa*	*hay*	ہے	ا–vs	obj	sub	masc	sg
sub	obj	vs–*ee*	*hay*	ہے	ی–vs	obj	sub	fem	
sub	obj	vs–*ey*	*hayN*	ہیں	ے–vs	obj	sub	masc	pl
sub	obj	vs–*ee*	*hayN*	ہیں	ی–vs	obj	sub	fem	

(b)

Transliteration				Urdu Script				GEND	NUM
sub	obj	vs–*aa*	*thaa*	تھا	ا–vs	obj	sub	masc	sg
sub	obj	vs–*ee*	*thee*	تھی	ی–vs	obj	sub	fem	
sub	obj	vs–*ey*	*they*	تھے	ے–vs	obj	sub	masc	pl
sub	obj	vs–*ee*	*theeN*	تھیں	ی–vs	obj	sub	fem	

Table 8.9: The Dependence of (a) Verb Morphemes (b) Auxiliary for the Object Agreement

(a)

Verb Morpheme		Object		Subject Case	Aux
		Number	Gender		
aa	ا	sg	masc	erg	–
ee	ی	sg, pl	fem	erg	–
ey	ے	pl	masc	erg	–
eeN	یں	pl	fem	erg	no

(b)

Auxiliary Form		Tense	Gender	Number
hay	ہے	present	masc, fem	sg
hayN	ہیں	present	masc, fem	pl
thaa	تھا	past	masc	sg
they	تھے	past	masc	pl
thee	تھی	past	fem	sg
theeN	تھیں	past	fem	pl

The agreement dependency between the verb-form and noun-phrases, that has been presented in the above tables, is summarized as general rules shown in (203), which describes that subject agreement is observed, when the subject bears a nominative case and the verb-form is a repetitive or subjunctive. The present and past tenses appear with the repetitive verb-form, while future tense appears with the

subjunctive verb-form. The verb for subject agreement can be intransitive, transitive or ditransitive, therefore the two object noun phrases are optional for subject agreement. However, for object agreement, the object noun phrase is not optional, and therefore object agreement is observed only for transitive and ditransitive verbs. Moreover, for object agreement, the object must be in a nominative case, if it is in an accusative case then the agreement is not with any of the noun phrase and the default singular-masculine verb-form is used.

(203) $S_{\text{subject-agreement}} \rightarrow NP_{\text{SUBJ-nominative}}\ (NP_{\text{OBJ2}})\ (NP_{\text{OBJ}})\ V_{\text{narrative-form}}\ AUX^*$

$S_{\text{subject-agreement}} \rightarrow NP_{\text{SUBJ-nominative}}\ (NP_{\text{OBJ2}})\ (NP_{\text{OBJ}})\ V_{\text{subjunctive-form}}\ (AUX_{\text{future}})$

$S_{\text{object-agreement}} \rightarrow NP_{\text{SUBJ-ergative}}\ (NP_{\text{OBJ2}})\ NP_{\text{OBJ-nominative}}\ V_{\text{perfective-form}}\ AUX^*$

The dependence for the subject and object agreement, shown in above tables and rules, can be directly encoded into LFG based lexical entries, using functional equations. For example, the lexical entry for a verb '*xareed-taa*', buy, is shown in (204), and for an auxiliary '*hooN*' is shown in (205). Using the rule shown in (204), the verb, V, and auxiliary, AUX, can combine to form V_1, the f-structure of which is shown in Figure 8.1, which contains constraint on the 'number', 'gender', 'person' and 'case' for the subject.

(204) *xareed-taa* V
```
(↑ PRED) = 'xareednaa<SUBJ, OBJ>'
(↑ V-FORM) = repetitive
(↑ SUBJ NUM) =c sg
(↑ SUBJ GEND) =c masc
(↑ SUBJ CASE) =c nom
```

(205) *hooN* AUX
```
(↑ TENSE) = present
(↑ V-FORM) =c repetitive
(↑ SUBJ NUM) =c sg
{  (↑ SUBJ GEND) =c masc
|  (↑ SUBJ GEND) =c fem   }
(↑ SUBJ CASE) =c nom
(↑ SUBJ PERS) =c 1st
```

(206) $V_1 \rightarrow V\ (AUX)^*$

$$\begin{bmatrix} \text{PRED} & \text{' } \textit{xareednaa} \langle \text{SUBJ, OBJ} \rangle \text{ '} \\ \text{TENSE} & \textit{present} \\ \text{V-FORM} & \textit{narrative} \end{bmatrix} \text{ with constraint } \begin{bmatrix} \text{SUBJ} & \begin{bmatrix} \text{NUM} & \textit{sg} \\ \text{GEND} & \textit{masc} \\ \text{PERS} & \textit{1st} \\ \text{CASE} & \textit{nom} \end{bmatrix} \end{bmatrix}$$

Figure 8.1: F-Structure of a Phrase V_1 '*xareed-taa hooN*'

The formation of the f-structure for auxiliary '*hooN*' is relatively simple from all other auxiliaries because '*hooN*' appears only for first-person subject-agreement, and, therefore, has lesser restrictions. The lexical entry for the auxiliary '*hay*', shown in (207), is more complex because it requires agreement sometimes with the subject and sometimes with the object, along with other constraints.

(207) *hay* AUX
```
(↑ TENSE) = present
{ { (↑ SUBJ NUM) =c sg
    {  (↑ SUBJ GEND) =c masc
    |  (↑ SUBJ GEND) =c fem }
    (↑ SUBJ PERS) =c 3rd      }
  |{(↑ SUBJ H-FORM) = frank
    {  (↑ SUBJ GEND) =c masc
    |  (↑ SUBJ GEND) =c fem }
    (↑ SUBJ PERS) =c 2nd      }
  (↑ SUBJ CASE) =c nom
  (↑ V-FORM) =c repetitive      }
| {(↑ OBJ NUM) =c sg
    {  (↑ OBJ GEND) =c masc
    |  (↑ OBJ GEND) =c fem }
    (↑ SUBJ CASE) =c erg
    (↑ V-FORM) =c perfect    }  }
```

(208) *-taa hay* VM
```
(↑ TENSE) = present
(↑ SUBJ GEND) =c masc
(↑ SUBJ CASE) =c nom
(↑ V-FORM) = repetitive
{{(↑ SUBJ NUM) =c sg
 (↑ SUBJ PERS) =c 3rd }
|{(↑ SUBJ H-FORM) = frank
  (↑ SUBJ PERS) =c 2nd }
```

-aa hay VM
```
(↑ TENSE) = present
(↑ V-FORM) = perfect
{(↑ OBJ NUM) =c sg
 (↑ OBJ GEND) =c masc
 (↑ SUBJ CASE) =c erg
|(↑ SUBJ CASE) =c nom }
```

(209) *xareed-* VB
```
(↑ PRED) = 'xareednaa<SUBJ, OBJ>'
```

(210) $V_2 \rightarrow$ VB VM

To simplify the lexical entry, a proposal presented in this work is to lump the verb-form suffix and the verb auxiliary into one unit and to term the combination as verb morpheme (VM) as shown in (208) (Rizvi and Hussain 2002). This simplifies the lexical entries and reduces search space during parsing and unification by avoiding multiple options, for example, options for the auxiliary in (207). The verb

base (VB) is stored separately as shown in the lexical entry (209). The VB describes information about the argument-structure and the VM describes information about agreement requirements. Although, this proposal results in extra lexical entries for the VM, but for each verb only the VB needs to be stored instead of storing all 60 verb-forms, therefore, the total number of lexical entries are significantly reduced. Moreover, this proposal is simpler to carry out because it can be implemented without using a morphological analyzer. As shown in the rule (210), the lumped VM can combine with the verb base (VB) to form a verb V_2. The f-structure formed using this rule is shown in Figure 8.2.

$$\begin{bmatrix} \text{PRED} & \text{' } \textit{xareednaa} \langle \text{SUBJ, OBJ} \rangle \text{ '} \\ \text{TENSE} & \textit{present} \\ \text{V-FORM} & \textit{perfective} \end{bmatrix} \text{ with constraint } \begin{bmatrix} \text{SUBJ} & [\text{CASE} \quad \textit{erg}] \\ \text{OBJ} & \begin{bmatrix} \text{NUM} & \textit{sg} \\ \text{GEND} & \textit{masc} \end{bmatrix} \end{bmatrix}$$

Figure 8.2: F-Structure of a Phrase V_2 '*xareed-aa hay*'

In Urdu, generally, to make a sentence, we need zero or more case marked noun phrases (NP) followed by a verb as shown in (211), where the verb can have V_1 form as in rule (206) or can have V_2 form using rule (210).

(211) $S \rightarrow \quad NP^* \quad \{V_1 \mid V_2\}$
$\quad\quad\quad\quad (\uparrow \text{GF}) = \downarrow \quad \uparrow = \downarrow$

The Urdu sentences shown in (212), (213), and (214) give evidence that the representation of the verb V_2 using a combination of VB and VM, is relatively simple than the representation of V_1 using a combination of V and AUX. The sentence in (212) has 'perfective' verb-form '*xareed-ee*' followed by auxiliary '*hay*' representing 'present' tense. The sentence in (213) has the same verb-form followed by past auxiliary '*thee*'. Therefore, for modeling using V and AUX combination, the ASPECT feature gets value 'perfect' from V, and the TENSE feature gets value 'present' or 'past' from AUX. However, this scheme requires special handling in finding the TENSE feature for the sentence in (214), which uses the 'perfective' verb-form without an auxiliary verb, and the TENSE attribute for the sentence is simple 'past'.

(212) حامد نے کتاب خریدی ہے
Haamed=ney *ketaab* *xareed-ee* *hay*
Hamid=*erg* book=*nom* buy-*pref.sg.m* AUX.*pres*
Hamid has bought a book. (TENSE = *present*, ASPECT = *perfect*).

(213) حامد نے کتاب خریدی تھی

Haamed=ney	*ketaab*	*xareed-ee*	*thee*
Hamid=*erg*	book=*nom*	buy-*pref.sg.m*	AUX.*past*

Hamid had bought a book. (TENSE = *past*, ASPECT = *perfect*).

(214) حامد نے کتاب خریدی

Haamed=ney	*ketaab*	*xareed-ee*
Hamid=*erg*	book=*nom*	buy-*pref.sg.m*

Hamid bought a book. (TENSE = *past*).

(215)

-ee	VM	**(↑ TENSE) = past** **(↑ OBJ NUM) =c sg** **(↑ OBJ GEND) =c fem**
-ee hay	VM	**(↑ TENSE) = present** **(↑ ASPECT) = perfect** **(↑ OBJ NUM) =c sg** **(↑ OBJ GEND) =c fem**
-ee thee	VM	**(↑ TENSE) = past** **(↑ ASPECT) = perfect** **(↑ OBJ NUM) =c sg** **(↑ OBJ GEND) =c fem**

However, by separating verb base (VB) (the part of a verb, responsible for the argument structure of the verb) from the morphological affix (the part of a verb that contains agreement features) and then defining verb morpheme (VM) as the suffix of the verb including all auxiliary verbs, the above-mentioned case may be handled. The VM contains information about TENSE, ASPECT, MOOD and agreement-features as shown in (215). The c-structure formed by using the combination of VB and VM for the sentence in (214) is shown in Figure 8.3.

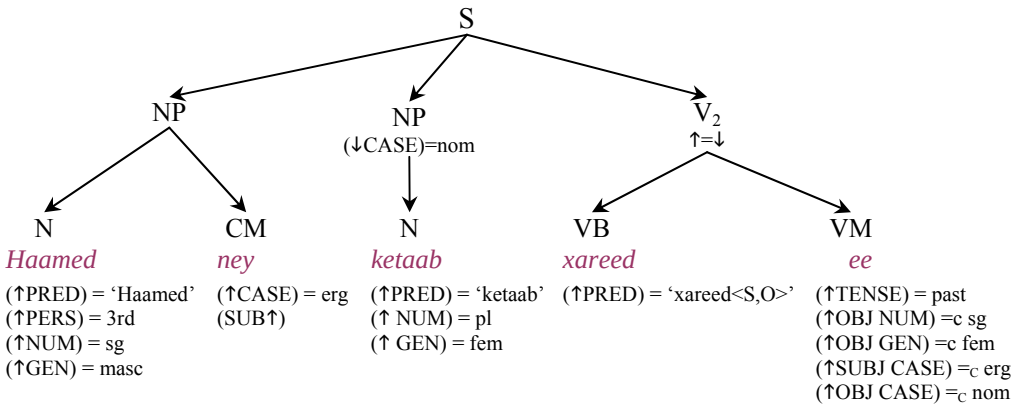


Figure 8.3: C-Structure of '*Haamed ney ketaab xareedee*' using VB and VM

However, combining VB and VM at the syntactic level is a violation of the 'lexical integrity principle', according to which only morphologically complete words

can be leaves in a c-structure tree (Bresnan 2001). For example, the use of VB and VM in coordinated structures is difficult to handle. Figure 8.4 shows the c-structure for the sentence in (212), which obeys the lexical integrity principle by combining morphologically complete words, i.e., V and AUX.

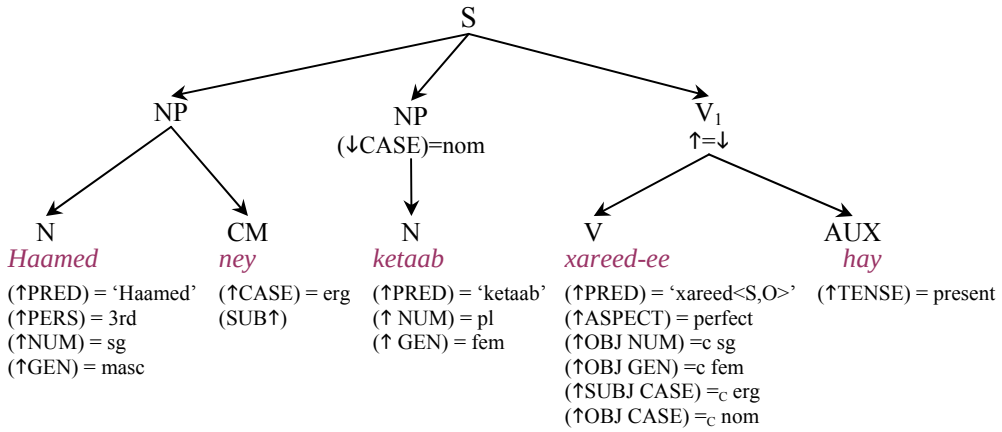


Figure 8.4: C-Structure of '*Haamed ney ketaab xareedee hay*' using V and AUX

In the following sections, the use of syntactic combination 'V and AUX' will be preferred over the combination 'VB and VM'. However, it has been worked out that the use of 'VB and VM' with aspectual and modal auxiliaries can also simplify LFG modeling equations. Similarly, this combination can also handle the 'complex predicate' by using the lexical entries for allowed N-V and V-V combination, instead of composing them at the syntactic level.

8.2 Verb Aspect in Urdu

The verb aspect gives a description about the duration, repetition and/or completion of an event without reference to its actual position in time. If the action of a verb has been completed, *tamaam* (تمام), it is termed as 'perfect', otherwise, if the action is incomplete, *naa tamaam* (نا تمام), the aspect is referred to as 'imperfect', 'progressive', or 'continuous' form, *jaaree* (جاری). In the previous section, we have seen that the perfective and imperfective (repetitive) morpheme is directly marked on the verb, which represents 'aspect' of the verb. In addition to 'aspectual morphemes', Urdu employs 'aspectual auxiliaries' to represent the 'aspect'. In Table 8.2, a verb's perfective form is used to represent present perfect tense, Table 8.10 shows the use of perfective aspectual-auxiliary '*chokaa*' to form present perfect tense.

Table 8.10: The Pattern of the Present-Perfect Tense using Perfective Auxiliary

Transliteration			Urdu Script			GEND	PERS	NUM	H-FORM
mayN	obj vs	*hooN*	میں	vs obj	چکا ہوں	masc	1st	sg	–
mayN	obj vs	*hooN*	میں	vs obj	چکی ہوں	fem			
ham	obj vs	*hayN*	ہم	vs obj	چکے ہیں	masc	1st	pl	–
ham	obj vs	*hayN*	ہم	vs obj	چکی ہیں	fem			
too	obj vs	*hay*	تو	vs obj	چکا ہے	masc	2nd	sg	frank
too	obj vs	*hay*	تو	vs obj	چکی ہے	fem			
tom	obj vs	*hao*	تم	vs obj	چکے ہو	masc	2nd	sg	formal
tom	obj vs	*hao*	تم	vs obj	چکی ہو	fem			
aap	obj vs	*hayN*	آپ	vs obj	چکے ہیں	masc	2nd	sg	polite
aap	obj vs	*hayN*	آپ	vs obj	چکی ہیں	fem			
woh	obj vs	*hay*	وہ	vs obj	چکا ہے	masc	3rd	sg	–
woh	obj vs	*hay*	وہ	vs obj	چکی ہے	fem			
woh	obj vs	*hayN*	وہ	vs obj	چکے ہیں	masc	3rd	pl	–
woh	obj vs	*hayN*	وہ	vs obj	چکی ہیں	fem			

Table 8.11: The Attributes Associated with the Aspectual Auxiliary Morphemes for the Agreement with a Nominative Subject

Morpheme		GEND	NUM	PERS	H-FORM	Tense Auxiliary
-aa	ا	masc	sg	1st, 3rd	–	–
-ee	ی	fem	sg, pl	1st, 3rd	–	–
-ey	ے	masc	pl	1st, 3rd	–	–
-eeN	یں	fem	pl	1st, 3rd	–	no
-aa	ا	masc	sg	2nd	frank	–
-ee	ی	fem	sg, pl	2nd	–	–
-ey	ے	masc	sg, pl	2nd	formal, polite	–

The aspectual auxiliaries in Urdu have 'gender' and 'number' morphemes: *-aa*, *-ee*, and *-ey* as shown in Table 8.11. The plural feminine morpheme *–eeN* appears only if the auxiliary is not used in a sentence. The aspectual auxiliaries with such morphemes usually follow verb's root-form (or stem-form). These auxiliaries require agreement in 'gender', 'number', 'person' and 'honor form' with a subject in the nominative case. In the following sub-sections, some commonly used Urdu aspectual auxiliaries are described.

8.2.1 Perfective Aspect

The 'perfective aspect' describes that the action or event has ended and appears in the present and past tenses. Urdu has two auxiliaries to show perfect aspect. More frequently used auxiliary to show perfective aspect is '*chok-aa*', the example sentence of which is shown in (216). It requires agreement with the nominative subject. Other auxiliary in Urdu, which describes completion, is '*l-ee-aa*' as shown in (217). This auxiliary has irregular morphology, appears with transitive verbs, and requires

agreement with an object in the 'gender' and 'number'. The LFG based lexical entries for both perfective auxiliaries are shown in (218).

There is a semantic difference between these two perfective auxiliaries. The auxiliary '*chok-aa*' tells about the end of an action. For example, the meaning of sentence (216) is: 'the event *reading* has ended, i.e., the whole book or a part of the book, whatever was intended to be read, has been read'. However, for sentence (217) the meaning is that 'whole book has been completely read'.

(216) وہ کتاب پڑھ چکا ہے

woh	*ketaab*	*paRh*	*chok-aa*	*hay*
He=*nom*	book=*nom*	read-*root*	AUX.*perf-sg.m*	AUX.*pres*

He has read the book.

(217) اس نے کتاب پڑھ لی ہے

aos=ney	*ketaab*	*paRh*	*l-ee*	*hay*
He=*erg*	book=*nom*	read-*root*	AUX.*completely-sg.f*	AUX.*pres*

He has (completely) read the book.

(218)

chok-aa	AUX	**(↑ TNS-ASP ASPECT) = perfect** **(↑ SUBJ GEND) =c masc** **(↑ SUBJ CASE) =c nom** **{ (↑ SUBJ NUM) =c sg** **{(↑ SUBJ PERS) =c 1st** **|(↑ SUBJ PERS) =c 3rd }** **| {(↑ SUBJ PERS) =c 2nd** **(↑ SUBJ H-FORM) =c frank} }.**
l-ee	AUX	**(↑ TNS-ASP ASPECT) = perfect** **(↑ TNS-ASP ACTION) = complete** **(↑ OBJ GEND) =c fem** **(↑ OBJ NUM) =c sg** **(↑ OBJ CASE) =c nom** **(↑ SUBJ CASE) =c erg.**

The general rule for the formation of sentences employing perfective auxiliaries is shown in (219), which shows that for auxiliary '*chok-aa*' the object NP is optional and subject case is nominative, while for '*l-ee-aa*', both NP's are required and the subject case is ergative. Figure 8.5 shows, side by side, f-structures of sentences (216) and (217). The value of the attribute ASPECT in both f-structures is 'perfect'. However, in the f-structure for the auxiliary '*l-ee-aa*', another attribute ACTION has a value 'complete' to show completion of the action with reference to the object.

(219) $S_{perfective} \rightarrow NP_{NOM\text{-}SUBJ}\ (NP_{OBJ})\ V_{ROOT\text{-}FORM}\ AUX_{chok\text{-}aa}\ AUX_{TENSE}$

$S_{perfective} \rightarrow NP_{ERG\text{-}SUBJ}\ NP_{OBJ}\ V_{ROOT\text{-}FORM}\ AUX_{l\text{-}ee\text{-}aa}\ AUX_{TENSE}$

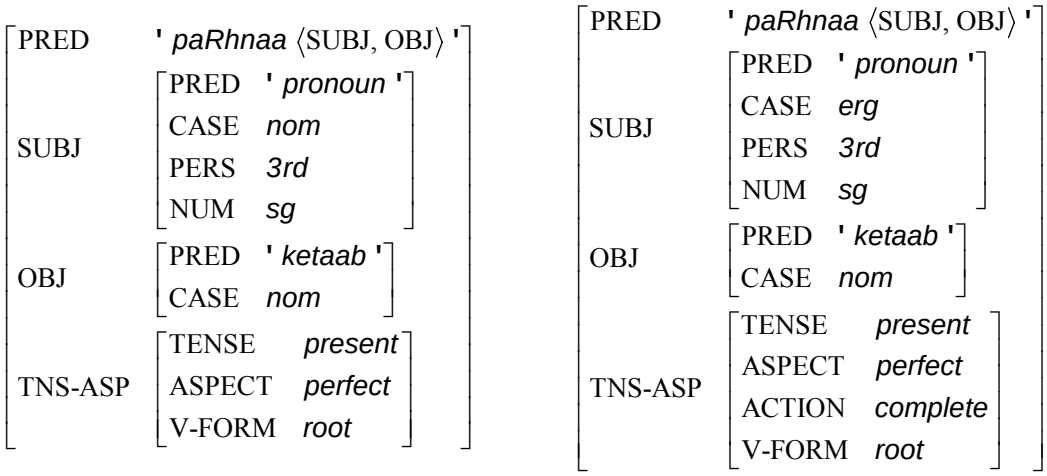


Figure 8.5: A Comparison of F-Structures of '*woh ketaab paRh chokaa hay*' versus '*aos ney ketaab paRh lee hay*'

8.2.2 Progressive Aspect

The progressive aspect describes the continuation of an event such that the event continues for the whole duration of the reference time. Urdu employs aspectual auxiliary '*rah-aa*' having morphemes: *-aa*, *-ee*, and *-ey*, which require subject agreement. The example sentence is shown in (220) and the rule for the progressive sentence formation is shown in (221), which can be extended for ditransitive and higher valency verbs. Figure 8.6 shows the f-structure of the progressive sentence in (220), which contains a value 'progressive' for the attribute ASPECT.

(220) وہ کتاب پڑھ رہا ہے

woh	*ketaab*	*paRh*	*rah-aa*	*hay*
He=*nom*	book=*nom*	read-*root*	AUX.*progressive-sg.m*	AUX.*pres*

He is reading a book.

(221) $S_{progressive} \rightarrow NP_{NOM\text{-}SUBJ}\ (NP_{OBJ})\ V_{ROOT\text{-}FORM}\ AUX_{rah\text{-}aa}\ AUX_{TENSE}$

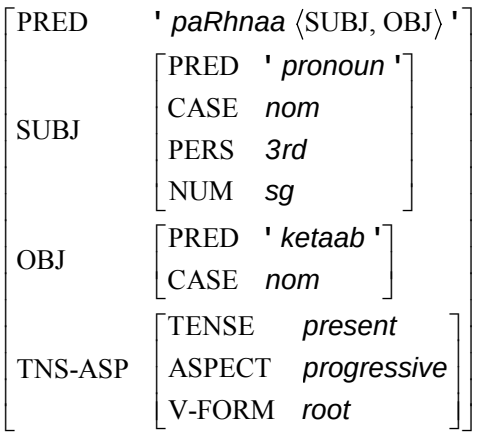


Figure 8.6: F-Structures of '*woh ketaab paRh rahaa hay*'

8.2.3 Repetitive Aspect

Urdu has aspectual auxiliaries, such as '*chal-aa*' and '*jaa-taa*', which show that an event or action is repeated for shorter and longer durations. In addition to show repetition, these auxiliaries, like English phrase 'keep on', also describe the persistency or resolve of the agent to perform an action. These auxiliaries are used with repetitive-form of a verb, and these require agreement with the subject in a nominative form. The repetitive-form of a verb, also called habitual-form 'استمراری', itself describes the repetition of an action. An example sentence without repetitive aspectual auxiliary is shown in (222). The auxiliary '*jaa-taa*' can be used without auxiliary '*chal-aa*', as shown in (223), but '*chal-aa*' always require '*jaa-taa*' to follow, as shown in (224). The auxiliary '*chal-aa*' adds the attributes of the continuation and/or longer-duration to the meanings of auxiliary '*jaa-taa*', and, therefore, increases the intensity of the persistency. The rule for the formation of repetitive sentence is shown in (225), which requires verb in the repetitive-form.

(222) وہ کتاب پڑھتا ہے

woh *ketaab* *paRh-taa* *hay*
He=*nom* book=*nom* read-*repeat* AUX.*pres*
He is used to read a book, or, He reads a book (daily or regularly).

(223) وہ کتاب پڑھتا جاتا ہے

woh *ketaab* *paRh-taa* *jaa-taa* *hay*
He=*nom* book=*nom* read-*repeat* AUX.*repeat* AUX.*pres*
He keeps on reading a book (repeatedly).

(224) وہ کتاب پڑھتا چلا جاتا ہے

woh *ketaab* *paRh-taa* *chal-aa* *jaa-taa* *hay*
He=*nom* book=*nom* read-*repeat* AUX.*cont* AUX.*repeat* AUX.*pres*
He keeps on reading a book (repeatedly and continuously).

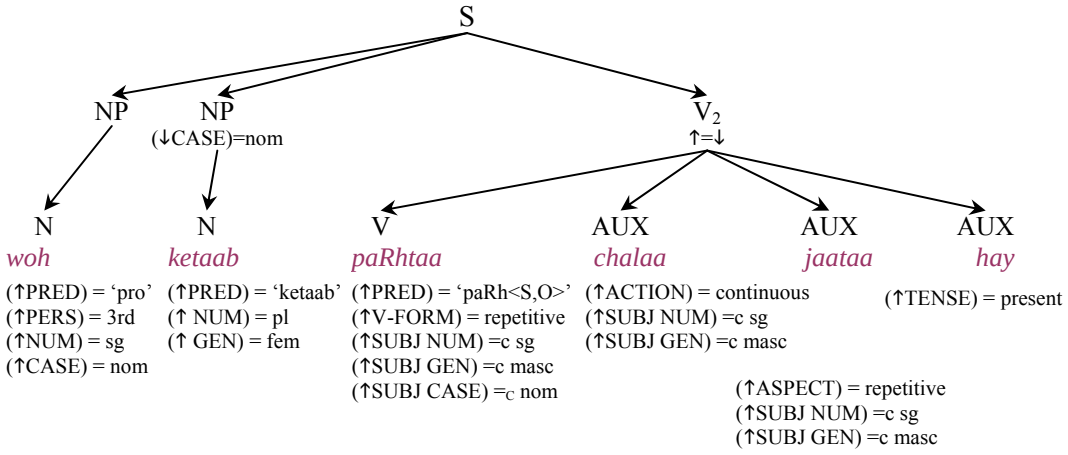


Figure 8.7: C-Structure of '*woh ketaab paRhtaa chalaa jaataa hay*'

$$\begin{bmatrix} \text{PRED} & \text{' } \textit{paRhnaa} \langle \text{SUBJ, OBJ} \rangle \text{ '} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{pronoun} \text{ '} \\ \text{CASE} & \textit{nom} \\ \text{PERS} & \textit{3rd} \\ \text{NUM} & \textit{sg} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{ketaab} \text{ '} \\ \text{CASE} & \textit{nom} \end{bmatrix} \\ \text{TNS-ASP} & \begin{bmatrix} \text{TENSE} & \textit{present} \\ \text{ASPECT} & \textit{repetitive} \\ \text{ACTION} & \textit{continuous} \\ \text{V-FORM} & \textit{repetitive} \end{bmatrix} \end{bmatrix}$$

Figure 8.8: F-Structures of '*woh ketaab paRhtaa chalaa jaataa hay*'

(225) $S_{\text{repetitive}} \rightarrow NP_{\text{NOM-SUBJ}}\ NP_{\text{OBJ}}\ V_{\text{NARRATIVE-FORM}}\ (AUX_{\text{chal-aa}})\ (AUX_{\text{jaa-taa}})\ AUX_{\text{TENSE}}$

There are two more repetitive aspectual auxiliaries in Urdu, which describe other features of repetition and persistency, such as a most occurring action and irregular but repeating action. The auxiliary '*rah-taa*' describes a predominant action over the reference time span, as shown in the sentence (226), in which the main action 'read' may be intercepted by other smaller actions, such as, eating or drinking, but the main action is 'read' and the interpretation of the sentence is that 'he usually keeps on reading a book'. The auxiliary '*kar-taa*' describes an irregular repetition of an action as shown in the sentence (227), the f-structure of which is shown in Figure 8.9.

$$\begin{bmatrix} \text{PRED} & \text{' } \textit{paRhnaa} \langle \text{SUBJ, OBJ} \rangle \text{ '} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{pronoun} \text{ '} \\ \text{CASE} & \textit{nom} \\ \text{PERS} & \textit{3rd} \\ \text{NUM} & \textit{sg} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{ketaab} \text{ '} \\ \text{CASE} & \textit{nom} \end{bmatrix} \\ \text{TNS-ASP} & \begin{bmatrix} \text{TENSE} & \textit{present} \\ \text{ASPECT} & \textit{repetitive} \\ \text{ACTION} & \textit{irregular} \\ \text{V-FORM} & \textit{perfective} \end{bmatrix} \end{bmatrix}$$

Figure 8.9: F-Structures of '*woh ketaab paRhaa kartaa hay*'

(226) وہ کتاب پڑھتا رہتا ہے

woh	*ketaab*	*paRh-taa*	*rah-taa*	*hay*
He=*nom*	book=*nom*	read-*repeat*	AUX.*mostly-sg.m*	be.*pres*

He mostly (for the maximum available time) reads a book.

(227) وہ کتاب پڑھا کرتا ہے

woh	*ketaab*	*paRh-aa*	*kar-taa*	*hay*
He=*nom*	book=*nom*	read-*perf*	AUX.*intermittently-sg.m*	AUX.*pres*

He intermittently (often but not regularly) reads a book.

(228) $S_{repetitive} \rightarrow NP_{NOM\text{-}SUBJ}\ NP_{OBJ}\ V_{NARRATIVE\text{-}FORM}\ AUX_{rah\text{-}taa}\ AUX_{TENSE}$

$S_{repetitive} \rightarrow NP_{NOM\text{-}SUBJ}\ NP_{OBJ}\ V_{PERFECTIVE\text{-}FORM}\ AUX_{kar\text{-}taa}\ AUX_{TENSE}$

8.2.4 Inceptive Aspect

The auxiliaries '*lag-aa*' and '*waal-aa*' describe the commencement of an action or event. For the same action, the position in time described by auxiliary '*lag-aa*' is closer in time than the auxiliary '*waal-aa*'. The auxiliary '*waal-aa*' describes that the action is going to start, the agent of which may be just finishing some other activity. While '*lag-aa*' describes that the action either is just going to start or even has just started.

(229) وہ کتاب پڑھنے لگا ہے

woh	*ketaab*	*paRh-ney*	*lagaa*	*hay*
He=*nom*	book=*nom*	read-*inf.m.obl*	AUX.*start*	be.*pres*

He has just started to read the book, (**`start = 1`**) *or*
He is just going to start reading a book. (**`start = 0`**).

(230) وہ کتاب پڑھنے والا ہے

woh	*ketaab*	*paRh-ney*	*waalaa*	*hay*
He=*nom*	book=*nom*	read-*inf.m.obl*	AUX.*start*	be.*pres*

He is going to start reading a book. (**`start = -1`**).

$$\begin{bmatrix} \text{PRED} & \text{' } \textit{paRhnaa} \langle \text{SUBJ, OBJ} \rangle \text{ '} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{pronoun} \text{ '} \\ \text{CASE} & \textit{nom} \\ \text{PERS} & \textit{3rd} \\ \text{NUM} & \textit{sg} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{ketaab} \text{ '} \\ \text{CASE} & \textit{nom} \end{bmatrix} \\ \text{TNS-ASP} & \begin{bmatrix} \text{TENSE} & \textit{present} \\ \text{ASPECT} & \textit{inceptive} \\ \text{ACTION} & \textit{going2start} \\ \text{V-FORM} & \textit{infinitive} \end{bmatrix} \end{bmatrix}$$

Figure 8.10: F-Structures of '*woh ketaab paRhney waalaa hay*'

8.3 Verb Mood in Urdu

The verb mood describes the purpose of an action, or the type of an action, such as a fact, news, command, request, wish, doubt, question, and potential. Languages

express distinctions of various moods either by inflecting the form of the verb or by using a modal auxiliary. In English, usually a modal auxiliary, such as should, would, could, etc., is used to show mood, while in Urdu, both modal auxiliaries and morphological affixation are used to show mood variations. In the following sub-sections, the commonly used moods in Urdu are described.

8.3.1 Declarative or News Mood

The declarative mood (also known as news-mood sentence – جمله خبریه) is used to describe state (being) of something, e.g., the state described in a factual statement, declaration, indication, information or news. This mood employs various verb-forms of the verb 'be' (*hao-naa* – ہونا), which normally in a non-declarative mood, is used as a tense-auxiliary without an argument structure. A verbal-predicate has an argument structure in contrast to a verb auxiliary, which indicates how many noun phrases are permitted in a sentence. The declarative mood of a sentence uses the verb 'be' as a verbal predicate having argument structure, instead of using it as a bare auxiliary. The argument structure has two arguments – one argument represents a subject noun phrase, which usually comes first in phrase order, and second argument represents a noun phrase, which instead of being a typical undergoer of an action, describes some 'information' or 'news' about the subject. In second noun phrase, sometimes a simple adjective is used as a noun to describe the state of the subject and sometimes a spatial location is used to describe the position of the subject. A general rule to form sentences with declarative mood is shown in (231).

(231) $S_{declarative} \rightarrow NP_{NOM\text{-}SUBJ} \{NP_{INFORMATION} \mid NP_{LOCATION}\} V_{BE}$

(232) حامد بیمار ہے

Haamed *beemaar* *hay*
Hamid=*nom* sick=*nom* be.*pres.sg*
Hamid is sick.

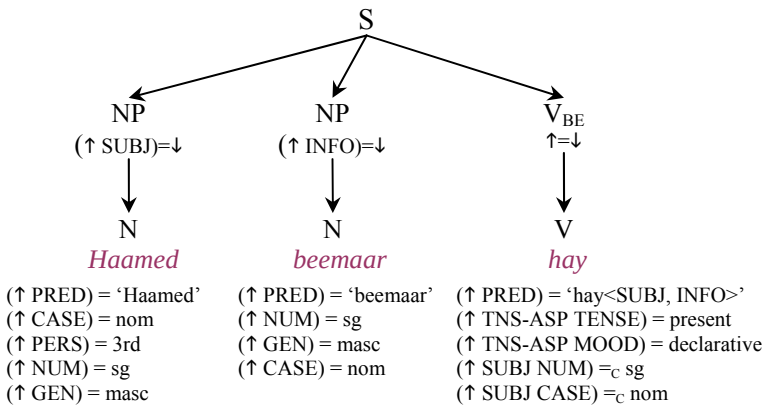


Figure 8.11: C-Structure of '*Haamed beemaar hay*'

An example of the declarative-mood sentence is shown in (232), the c-structure and f-structure of which are shown in Figure 8.11 and Figure 8.12 respectively. Figures show that the predicate '*hay*' requires two arguments – the subject and the information.

$$\begin{bmatrix} \text{PRED} & \text{' } \textit{hay} \ \langle \text{SUBJ, INFO} \rangle \text{ '} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{Haamed} \text{ '} \\ \text{CASE} & \textit{nom} \\ \text{PERS} & \textit{3rd} \\ \text{NUM} & \textit{sg} \end{bmatrix} \\ \text{INFO} & \begin{bmatrix} \text{PRED} & \text{' } \textit{beemaar} \text{ '} \\ \text{CASE} & \textit{nom} \end{bmatrix} \\ \text{TNS-ASP} & \begin{bmatrix} \text{TENSE} & \textit{present} \\ \text{MOOD} & \textit{declarative} \end{bmatrix} \end{bmatrix}$$

Figure 8.12: F-Structures of '*Haamed beemaar hay*'

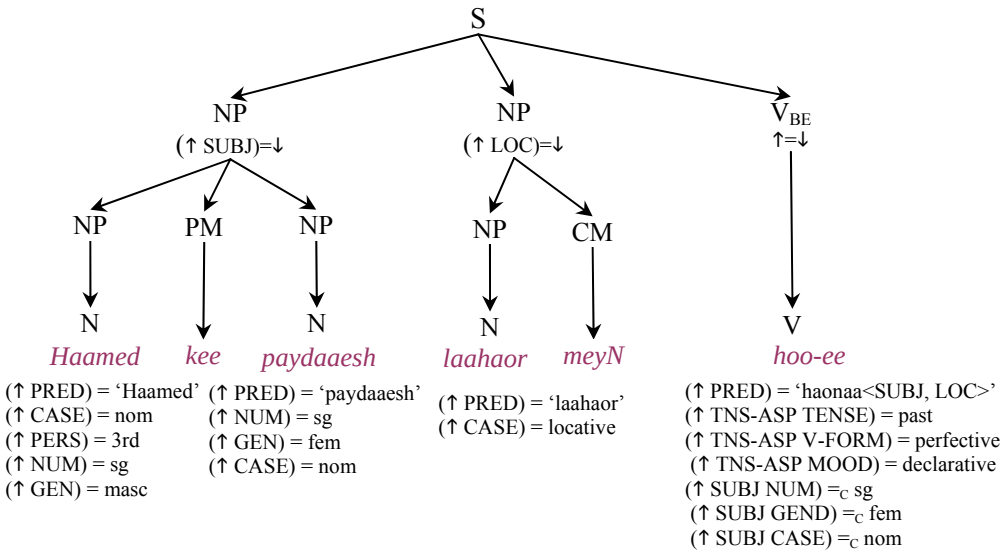


Figure 8.13: C-Structure of '*Haamed kee paydaaesh laahaor meyN hooee*'

Although the present form of predicate '*hay*' does not require gender agreement, the perfective forms '*hoo-aa*' and '*hoo-ee*' require gender agreement with the subject as shown in the examples (233) and (234). In the case of possessive NP, the agreement is with the last possessee NP in the chain of possessive NP's.

(233) حامد کی پیدائش لاہور میں ہوئی

Haamed=kee	*paydaaesh*	*laahaor=meyN*	*hoo-ee*
Hamid=*gen*	birth.*nom.sg.fem*	Lahore.*loc*	happen-*perf.sg.fem*

Hamid's birth took place in Lahore

(234) حامد کا جنم لاہور میں ہوا

Haamed=kaa *janam* *laahaor=meyN* *hoo-aa*

Hamid=*gen* birth.*nom.sg.masc* Lahore.*loc* happen-*perf.sg.masc*

Hamid's birth took place in Lahore.

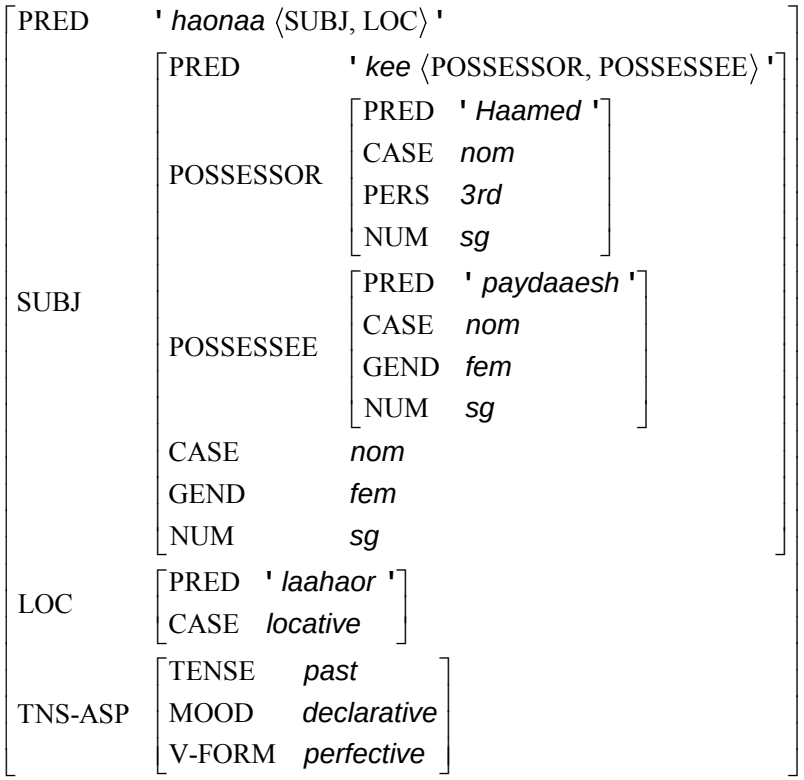


Figure 8.14: F-Structures of '*Haamed kee paydaaesh laahaor meyN hooee*'

The agreement of verb in sentence (233) is with the subject, which gets its gender, number and case features from the possessee as described in the c-structure and f-structure shown in Figure 8.13 and Figure 8.14. The sg-fem noun '*paydaaesh*' (birth) is used with the sg-fem verb-form '*hooee*' as shown in (235). Similarly, the sg-masc noun '*janam*' (birth) is used with sg-masc verb-form '*hooaa*', as shown in the sentence (234). The analysis presented in this work is different from (Mohanan 1990). Mohanan assumed that for a sentence like the one shown in (236), '*janam hooaa*' is a N-V complex predicate having genitive subject '*Haamed kaa*' and a locative object '*laahaor meyN*'. However, in this work it is assumed that, for the sentence in (236), '*Haamed kaa*' is an incomplete noun phrase, until it is joined with another ***nominative*** noun phrase. The phrase '*laahaor meyN*' is not a nominative noun phrase, therefore, '*paydaaesh*' is a possessee of the possessor '*Haamed*', which comes after a locative phrase in the phrase-order. Based on this observation, this work assumes a rule for the noun phrase order in Urdu and Hindi as shown in (237). This assumption is also found useful for the analysis of other moods like the permissive mood.

(235) حامد کی لاہور میں پیدائش ہوئی

Haamed=kee *laahaor=meyN* *paydaaesh* *hoo-ee*
Hamid=*gen* Lahore.*loc* birth.*nom.sg.fem* happen-*perf.sg.fem*
Hamid's birth took place in Lahore

(236) حامد کا لاہور میں جنم ہوا

Haamed=kaa *laahaor=meyN* *janam* *hoo-aa*
Hamid=*gen* Lahore.*loc* birth.*nom.sg.masc* happen-*perf.sg.masc*
Hamid's birth took place in Lahore.

(237) ***Assumption***: Surface Linear Order for Noun Phrases in Urdu/Hindi:

"If case-marking on noun phrases enables identifying arguments for all the predicates having the argument-structure in a sentence, then the noun phrases in Urdu and Hindi can take any surface linear order, and even the arguments of different predicates could be scrambled."

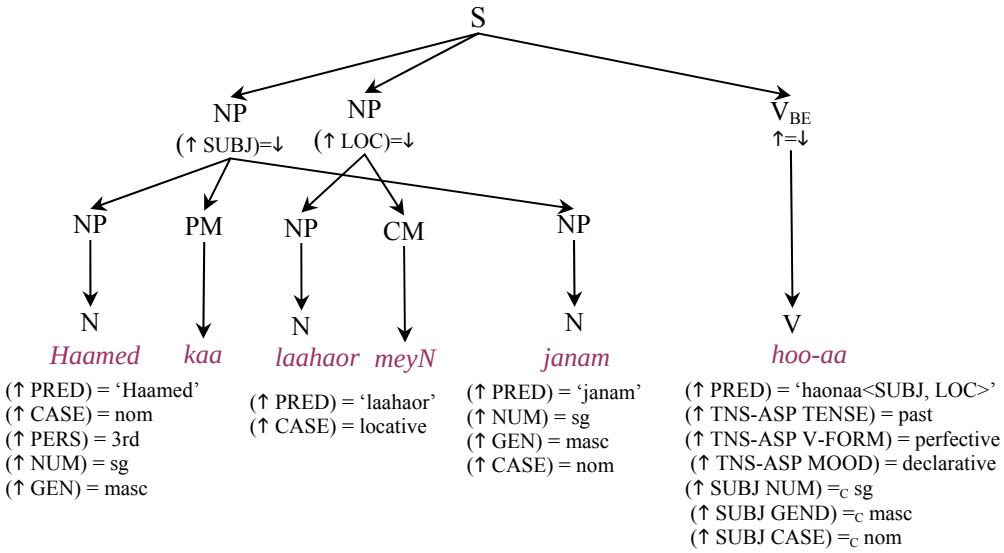


Figure 8.15: C-Structure of '*Haamed kaa laahaor meyN janam hooaa*'

According to the assumption (237), the c-structure of sentence (236) is shown in Figure 8.15, which looks inappropriate, because it cannot be generated with the context-free grammar's phrase structure rules. To generate such a c-structure, either parsing rules should be modified or a transformation to change the linear order may be needed in such a way that phrase '*Haamed kaa*' is followed by a nominative noun phrase '*janam*', by moving the phrase '*laahaor mayN*', prior to parsing.

The sentence in (238) gives more evidence for the assumption (237) presented in this work for the sentence in (236). The phrase '*janam haonaa*' (be born) may sometime be treated as an N-V complex predicate, but '*makan hay*' (house, be) cannot be treated as an N-V complex predicate.

(238) حامد کا لاہور میں مکان ہے

Haamed=kaa *laahaor=meyN* *makan* *hay*
Hamid=*gen* Lahore.*loc* house.*nom.sg.masc* be-*pres.sg*
Hamid's house is in Lahore.

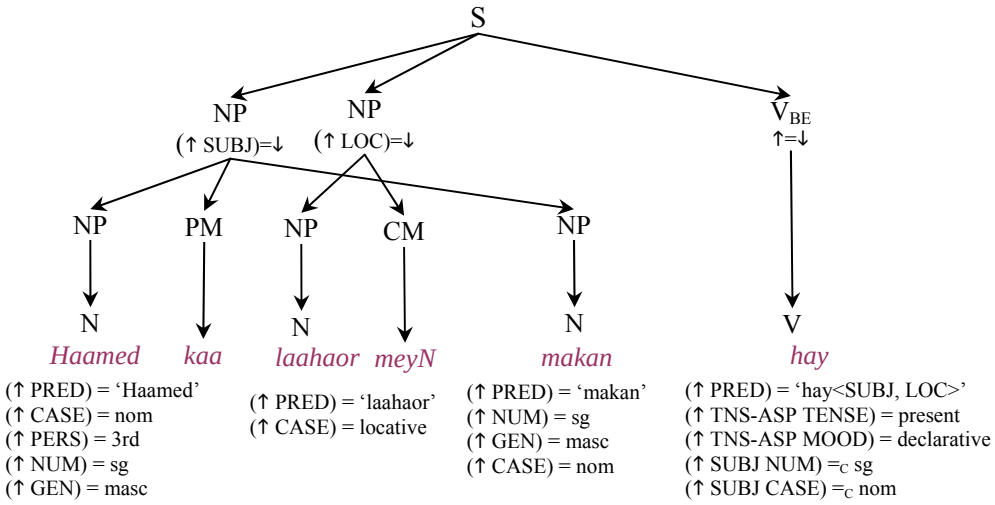


Figure 8.16: C-Structure of '*Haamed kaa laahaor meyN makan hay*'

(239) یہ حامد کی کتاب ہے

yeh *Haamed=kee* *ketaab* *hay*
This Haamed=*gen* book.*nom.sg.fem* be-*pres.sg*
This is Hamid's book.

(240) یہ کتاب حامد کی ہے

yeh *ketaab* *Haamed=kee* *hay*
This book.*nom.sg.fem* Haamed=*gen* be-*pres.sg*
This is Hamid's book (with a focus on the 'book').

(241) حامد کی کتاب یہ ہے

ketaab *Haamed=kee* *yeh* *hay*
book.*nom.sg.fem* Haamed=*gen* This be-*pres.sg*
This is Hamid's book (with a focus on 'Hamid').

The sentences (239)–(241) show scrambling of a sentence with a possessive phrase, in which '*Haamed*' is a possessor and the '*ketaab* – book' is a possessee. For a possessive phrase, the agreement is with the possessee, but the focus is on the possessor. As another example, in the sentence (242), verb agreement is with '*paydaaesh* – birth', but the focus is on '*Haamed*', therefore '*aapney* – his.*obl*' refers to '*Haamed*'.

(242) حامد کی پیدائش اپنے نانا کے گھر میں ہوئی

Haamed kee paydaaesh *aapney naanaa key ghar=meyN* *hoo-ee*
Hamid's.*focus* Birth.*sg.fem* his grandfather's home=*location* happen-*perf.sg.fem*
Hamid's birth took place at his grandfather's home.

8.3.2 Permissive Mood

The permissive mood describes that someone allows someone to perform an action. In this mood, an oblique-infinitive form is followed by a permissive verb '*deynaa*', which has an argument-structure, instead of a modal auxiliary. A rule for general surface order of phrases in this mood is shown in (243) and an example sentence is shown in (244).

(243) $S_{permissive} \rightarrow NP_{SUBJ\text{-}ergative}\ NP_{dative}\ NP_{nom}\ V_{infinitive\text{-}obl}\ V_{dey\text{-}naa}$

(244) حامد نے انجم کو کتاب پڑھنے دی

Haamed=ney *aanjom=kao* *ketaab* *paRh-ney* *d-ee*
Hamid=*erg* Anjom=*dat* book=*nom.sg.f* read-*inf.obl* let-*perf.sg.f*
Hamid let Anjom read a book.

(245) حامد نے انجم کو کتاب پڑھنے کے لیے دی

Haamed=ney *aanjom=kao* *ketaab* *paRh-ney* *key leeey* *d-ee*
Hamid=*erg* Anjom=*dat* book=*nom.sg.f* read-*inf.obl* for give-*perf.sg.f*
Hamid gave Anjom a book for reading.

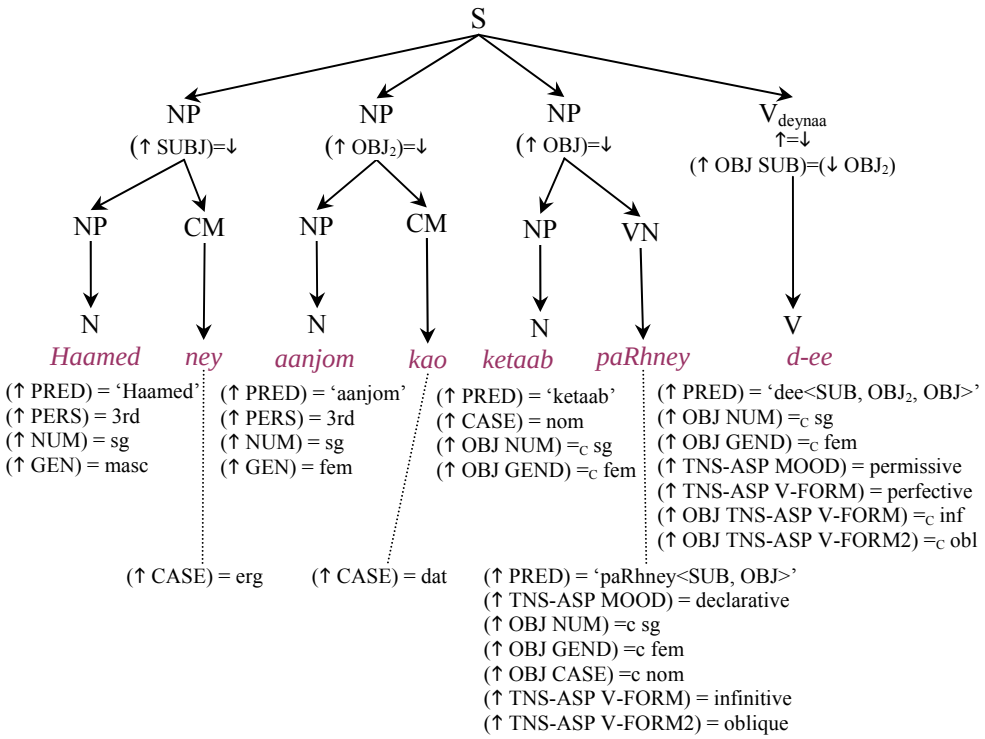


Figure 8.17: C-Structure of '*Haamed ney aanjom kao ketaab paRhney dee*'

In this work, one analysis of a permissive sentence in (244) is shown as annotated c-structure in Figure 8.17. This analysis considers the verb '*dee*' as permissive (let) if the subject is in ergative case, the goal object (OBJ_2) is in the dative case and the object is an oblique-infinitive phrase. In contrast to '*dee*' as 'give' in

sentence (245), which also has three arguments but the object in that case is not a verbal noun and, therefore, does not have attributes for V-FORM and V-FORM2 as infinitive and oblique, respectively. Moreover, the phrase '*paRhney key leeey*' (for reading) is a post-positional phrase in (245) and acts as adjunct in the final f-structure. The f-structure for the sentence (244) is shown in Figure 8.18

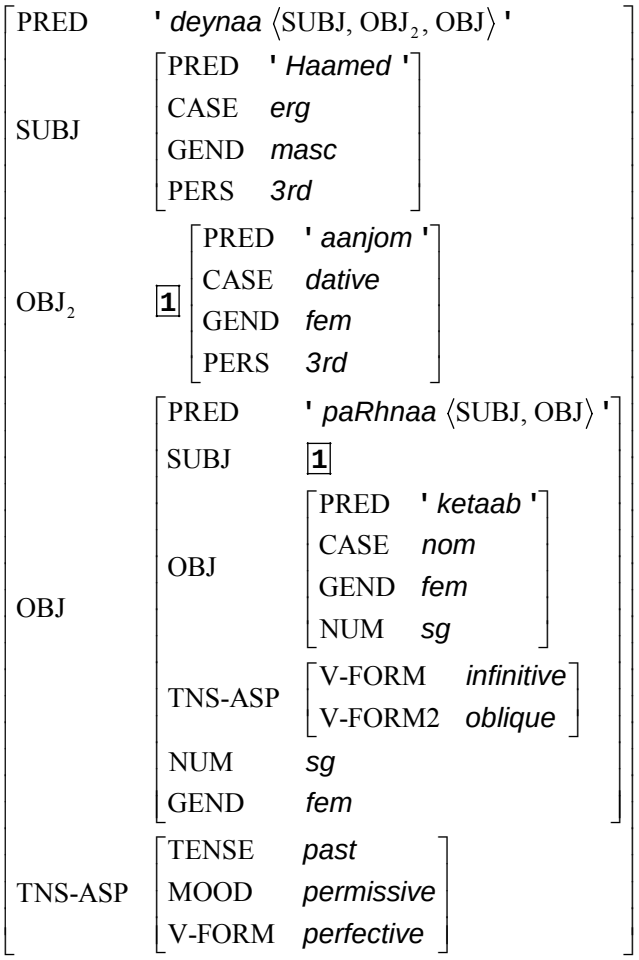


Figure 8.18: F-Structures of '*Haamed ney aanjom kao ketaab paRhney dee*'

The f-structure in Figure 8.18 assumes that infinitive '*paRhnaa*' also has an argument structure, the subject (SUBJ) of this infinitive verb is '*aanjom*', which is an indirect object (OBJ_2) of the permissive verb '*deynaa*' and the object of the infinitive is '*ketaab*'. The infinitive has its own tense-aspect attributes. The gender and number attributes of this verbal noun are the same as that of the object, which are singular and feminine.

(246) حامد نے کتاب انجم کو پڑھنے دی

Haamed=ney	*ketaab*	*aanjom=kao*	*paRh-ney*	*d-ee*
Hamid=*erg*	book=*nom.sg.f*	Anjom=*dat*	read-*inf.obl*	let-*perf.sg.f*

Hamid let Anjom read a book.

The sentence in (246) is the same as the sentence in (244), but shows a different order of noun phrases. The sentence in (244) is the more acceptable form of a permissive sentence, but the sentence in (246) is also acceptable. In these sentences, there are four NPs: 'Hamid' is ergative, 'book' is nominative, 'Anjom' is dative and 'to read' is infinitive. Moreover, there are two verbs with argument-structures: '*dee*' (let) requires three NPs: an ergative, a dative and an infinitive. The infinitive 'to read' requires two NPs: a dative subject and a nominative object. The argument of these verbs are satisfied based on the case-marking according to the assumption (237) made in this work. The *scrambled c-structure* of sentence (246) is shown in Figure 8.19. The f-structure of sentence (246) is the same as that of sentence (244) shown in Figure 8.18.

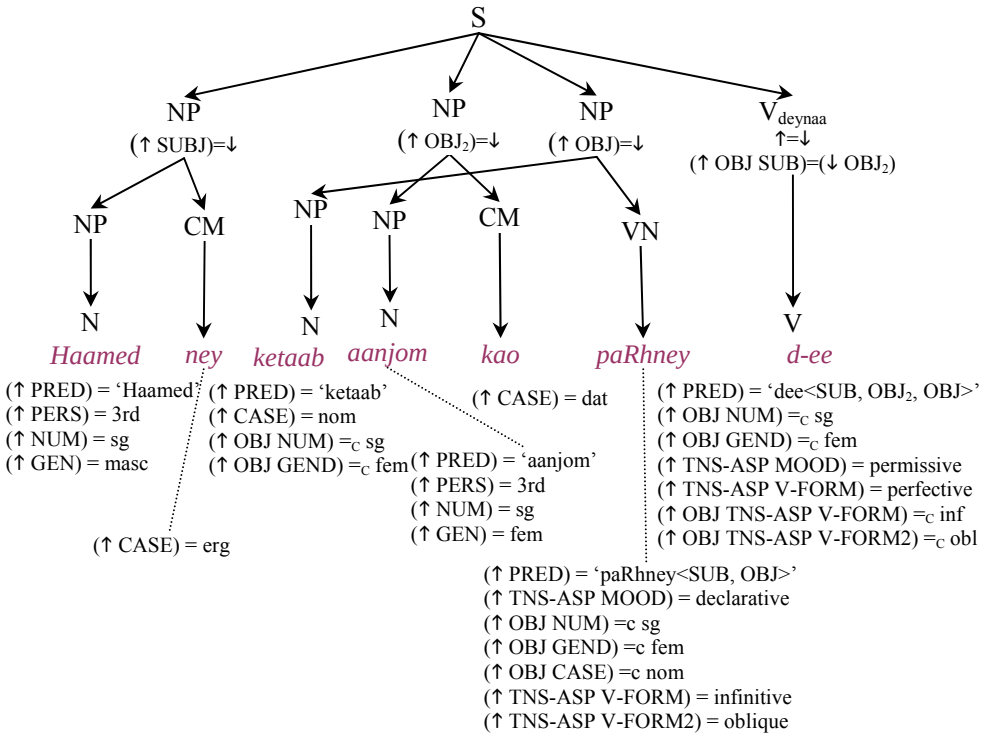


Figure 8.19: C-Structure of '*Haamed ney ketaab aanjom kao paRhney dee*'

8.3.3 Prohibitive Mood

The structure of prohibitive mood in Urdu is similar to permissive mood and this mood describes that someone disallows someone to perform an action. In this mood, an oblique-infinitive form with marker '*sey*' is followed by a prohibitive verb '*manA karnaa*', which has an argument-structure. A rule for general surface order of phrases in this mood is shown in (247) and an example sentence is shown in (248).

(247) $S_{prohibitive} \rightarrow NP_{SUBJ\text{-}ergative}\ NP_{dative}\ NP_{nom}\ V_{infinitive\text{-}obl}$ *sey* $V_{manA\ kar\text{-}naa}$

(248) حامد نے انجم کو کتاب پڑھنے سے منع کیا

Haamed=ney aanjom=kao ketaab paRh-ney=sey manA kee-aa

Hamid=*erg* Anjom=*dat* book=*nom.sg.f* read-*inf.obl=inf* prohibit-*perf.sg.m*

Hamid prohibited Anjom from reading a book.

$$\begin{bmatrix}
\text{PRED} & \text{' } manA\ keeaa\ \langle \text{SUBJ, OBJ}_2\text{, OBJ} \rangle \text{ '} \\
\text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{' } Haamed \text{ '} \\ \text{CASE} & erg \\ \text{GEND} & masc \\ \text{PERS} & 3rd \end{bmatrix} \\
\text{OBJ}_2 & \boxed{1} \begin{bmatrix} \text{PRED} & \text{' } aanjom \text{ '} \\ \text{CASE} & dative \\ \text{GEND} & fem \\ \text{PERS} & 3rd \end{bmatrix} \\
\text{OBJ} & \begin{bmatrix}
\text{PRED} & \text{' } paRhnaa\ \langle \text{SUBJ, OBJ} \rangle \text{ '} \\
\text{SUBJ} & \boxed{1} \\
\text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } ketaab \text{ '} \\ \text{CASE} & nom \\ \text{GEND} & fem \\ \text{NUM} & sg \end{bmatrix} \\
\text{TNS-ASP} & \begin{bmatrix} \text{V-FORM} & infinitive \\ \text{V-FORM2} & oblique \end{bmatrix} \\
\text{NUM} & sg \\
\text{GEND} & fem \\
\text{CASE} & infinitive
\end{bmatrix} \\
\text{TNS-ASP} & \begin{bmatrix} \text{TENSE} & past \\ \text{MOOD} & prohibitive \\ \text{V-FORM} & perfective \end{bmatrix}
\end{bmatrix}$$

Figure 8.20: F-Structures of '*Haamed ney aanjom kao ketaab paRhney sey manA keeaa*'

The f-structure of prohibitive sentence (248) is shown in Figure 8.20, which is similar to the permissive f-structure, except that object (OBJ) has infinitive case marked with the '*sey*'.

8.3.4 Imperative Mood

The imperative mood is used to expresses a command (*aamar* – امر), prohibition (*nahee* – نہی), suggestion or request. If an elder or powerful person uses this mood then it expresses command or prohibition and if younger or submissive person uses this mood then it expresses request. Urdu, English and many other languages, use the verb stem form to represent an imperative sentence. Moreover, the second-person is semantically *implied* as the subject of an imperative sentence to whom order is given

and may be omitted. Special verb morphemes in Urdu are used to represent variations within the imperative mood, such as, frank, formal, polite and request as shown in the following example sentences.

(249) تو کتاب پڑھ

too	*ketaab*	*paRh*
You=*nom.frank*	book.*sg.masc*	read=*imp.frank*

You read the book (in a frank or rude way).

(250) تم کتاب پڑھو

tom	*ketaab*	*paRh-ao*
You=*nom.formal*	book.*sg.masc*	read=*imp.formal*

You read the book (in a formal way or with someone familiar).

(251) آپ کتاب پڑھیں

aap	*ketaab*	*paRh-eyN*
You=*nom.polite*	book.*sg.masc*	read=*imp.polite*

You read the book (in a polite or respectful way).

(252) آپ کتاب پڑھیئے

aap	*ketaab*	*paRh-ee-ey*
You=*nom.polite*	book.*sg.masc*	read=*imp.request*

You read the book, please (in a more-polite way or as a request).

(253) آپ کتاب پڑھ لیجیئے

aap	*ketaab*	*paRh*	*l-ee-jee-ey*
You=*nom.polite*	book.*sg.masc*	read=*root*	AUX.*imp.appeal*

You, please, read the book (in an extra-polite way or as an appeal).

Table 8.12: The Urdu Imperative Verb Forms for the Imperative Mood

Imperative	Frank (or Rude)	Formal (or Familiar)	Polite (or Respect)	More Polite (or Request)	Extra Polite (or Appeal)
Read	پڑھ *paRh*	پڑھو *paRh-ao*	پڑھیں *paRh-eyN*	پڑھیئے *paRh-ee-ey*	پڑھ لیجیئے *paRh leejeeey*
Look	دیکھ *deykh*	دیکھو *deykh-ao*	دیکھیں *deykh-eyN*	دیکھیئے *deykh-ee-ey*	دیکھ لیجیئے *deykh leejeeey*
Speak	بول *baol*	بولو *baol-ao*	بولیں *baol-eyN*	بولیئے *baol-ee-ey*	بول دیجیئے *baol deejeeey*
Sit	بیٹھ *bayTh*	بیٹھو *bayTh-ao*	بیٹھیں *bayTh-eyN*	بیٹھیئے *bayTh-ee-ey*	بیٹھ جائیے *bayTh jaaeeey*
Eat	کھا *khaa*	کھاؤ *khaa-ao*	کھائیں *khaa-eyN*	کھایئے *khaa-ee-ey*	کھا لیجیئے *khaa leejeeey*

Table 8.12 summarizes the imperative verb forms, for some Urdu verbs, which are used in the imperative mood. Figure 8.21 shows the c-structure tree and Figure 8.22 shows the f-structure for the sentence in (252). The imperative verb-form in the

c-structure has constraints on the subject that it should be a second person, having nominative case and polite-form of the pronoun. If the second person pronoun is omitted, then these attributes are implied by default.

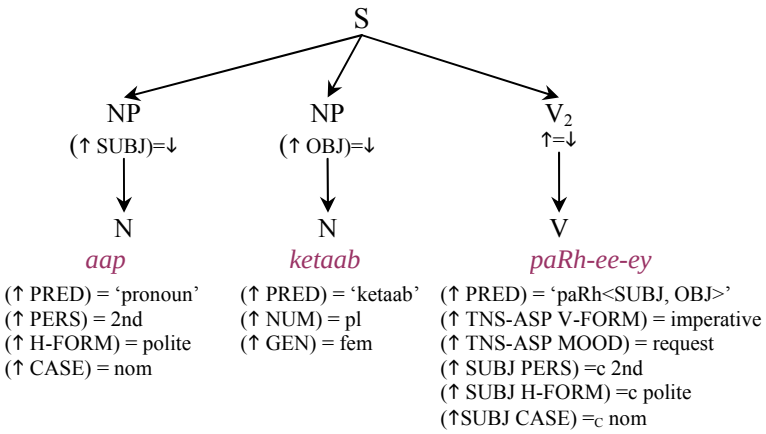


Figure 8.21: C-Structure of '*aap ketaab paRheeey*'

$$\begin{bmatrix} \text{PRED} & \text{' } \textit{paRhnaa} \ \langle \text{SUBJ, OBJ} \rangle \text{ '} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{pronoun} \text{ '} \\ \text{CASE} & \textit{nom} \\ \text{PERS} & \textit{2nd} \\ \text{H-FORM} & \textit{polite} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{ketaab} \text{ '} \\ \text{CASE} & \textit{nom} \end{bmatrix} \\ \text{TNS-ASP} & \begin{bmatrix} \text{MOOD} & \textit{request} \\ \text{V-FORM} & \textit{imperative} \end{bmatrix} \end{bmatrix}$$

Figure 8.22: F-Structures of '*aap ketaab paRheeey*'

8.3.5 Capacitive Mood

The capacitive mood shows the capability of the agent for performing an action. This mood employs the auxiliary '*sak-taa*' to tell that the attribute of this mood is capability. A general rule for capacitive mood is shown in (254) and the example sentence is shown in (255)

(254) $S_{capacitive} \rightarrow NP_{SUBJ\text{-}nom}\ NP_{OBJ\text{-}nom}\ V\ AUX_{sak\text{-}taa}\ AUX_{tense}$

(255) میں کتاب پڑھ سکتا ہوں

mayN	*ketaab*	*paRh*	*sak-taa*	*hooN*
I=*nom.1st.sg*	book=*nom.sg.f*	read-*root*	AUX-*capacitive.sg.m*	AUX-*pres.1st.sg*

I can read a book.

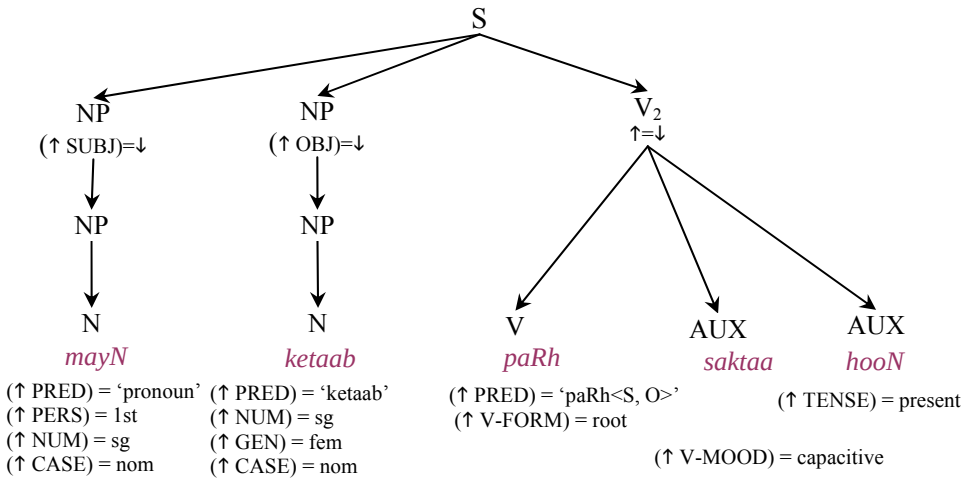


Figure 8.23: C-Structure of '*mayN ketaab paRh saktaa hooN*'

8.3.6 Suggestive Mood

In a suggestive mood, some suggestion or advice to performing an action is given to a subject in the accusative case. This mood employs the auxiliary '*chaaheeey*', which follows the infinitive verb-from and adds a value 'suggestive' to the attribute mood. A general rule for suggestive mood is shown in (256), the example sentence is shown in (257) and f-structure is shown in Figure 8.24.

(256) $S_{suggestive} \rightarrow NP_{SUBJ\text{-}accusative}\ NP_{OBJ\text{-}nom}\ V_{infinitive}\ AUX_{chaaheeey}\ (AUX_{past})$

(257) تمہیں کتابیں پڑھنی چاہیئے

tomheeN	*ketaabeyN*	*paRh-nee*	*chaaheeey*
You=*acc.2nd*	books=*nom.pl.fem*	read-*inf.fem*	AUX-*suggestive*

You should read books.

$$\begin{bmatrix} \text{PRED} & \text{' } \textit{paRhnaa} \langle \text{SUBJ, OBJ}_2\text{, OBJ} \rangle \text{ '} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{pronoun} \text{ '} \\ \text{CASE} & \textit{acc} \\ \text{PERS} & \textit{2nd} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{ketaabeyN} \text{ '} \\ \text{CASE} & \textit{nom} \\ \text{GEND} & \textit{fem} \\ \text{NUM} & \textit{pl} \end{bmatrix} \\ \text{TNS-ASP} & \begin{bmatrix} \text{TENSE} & \textit{present} \\ \text{MOOD} & \textit{suggestive} \\ \text{V-FORM} & \textit{infinitive} \end{bmatrix} \end{bmatrix}$$

Figure 8.24: F-Structures of '*tomheeN ketaabeyN paRhnee chaaheeey*'

(258) تمہیں کتاب پڑھنی چاہیئے

tomheeN *ketaab* *paRh-nee* *chaaheeey*
You=*acc.2nd* book=*nom.sg.fem* read-*inf.fem* AUX-*suggestive*
You should read a book.

8.3.7 Compulsive Mood

The sentence in (259) shows an example of the compulsive mood, in which an ergative or accusative subject has to perform an action as a desire or self-imposed obligation. The sentence (260) employs auxiliary '*paR-aa*' to show externally-imposed compulsion, in which an accusative subject has to perform an action. The pattern of compulsive mood is shown in (261).

(259) اس نے / اسے کتاب پڑھنی ہے

aos=ney/aosey *ketaab* *paRh-nee* *hay*
He=*erg/acc* book=*nom* read.*inf.sg.masc* AUX.*pres*
He wants to/has to read a book (self-imposed obligation).

(260) اس نے/ اسے کتاب پڑھنا پڑی ہے*

aosey *ketaab* *paRh-naa* *paR-ee* *hay*
He=**erg/acc* book=*nom* read.*inf.sg.masc* AUX.*compulsive-sg.fem* AUX.*pres*
He has to read a book (externally-imposed obligation, unwillingness).

(261) $S_{\text{self-compulsive}} \rightarrow NP_{\text{SUBJ-accusative}}\ NP_{\text{OBJ-nom}}\ V_{\text{infinitive}}\ AUX_{\text{tense}}$

$S_{\text{external-compulsive}} \rightarrow NP_{\text{SUBJ-accusative}}\ NP_{\text{OBJ-nom}}\ V_{\text{infinitive}}\ AUX_{\text{paR-aa}}\ (AUX_{\text{tense}})$

8.3.8 Dubitative/Presumptive Mood

The dubitative mood, (*shakeeah* – شکیہ) or presumptive mood, (*aeHteymaalee* – احتمالی) is used to express the speaker's doubt or uncertainty about an event. If this mood is used with '*chokaa*' auxiliary, then it represents perfective aspect and with '*rahaa*' auxiliary, it represents progressive aspect. Similarly, other aspect auxiliaries can be used to represent other aspects. In this mood, the tense is represented neither by the verb nor by the auxiliaries. Therefore, this mood can be used with the same verb form and auxiliaries for the present, past or future tense. Sometimes the accompanying adverbial or adjective phrases describe information about the tense and sometimes the context in which this sentence is used, could be used to find tense information. The example sentences are shown in (262) – (266) and general rule is shown in (267).

(262) وہ کتاب پڑھ چکا ہو گا

woh *ketaab* *paRh* *chokaa* *hao gaa*
He book read.*root* AUX.*perf* be.*sg.m.3.presumptive*
He would have read the book.

(263) اب تک پاکستان جیت چکا ہو گا

aab tak *paakestaan* *jeet* *chokaa* *hao gaa*
Until now, Pakistan won.*root* AUX.*perf* be.*sg.m.3.presumptive*
Until now, Pakistan would have won.

(264) وہ سکول جا رہی ہو گی

woh *sakool* *jaa* *rahee* *hao gee*
she school go.*root* AUX.*cont* be.*sg.f.2.presumptive*
She will be going to school.

(265) وہ مری جا رہے ہوں گے

woh *mare* *jaa* *rahey* *haoN gey*
They Murree go.*root* AUX.*cont* be.*pl.m.2.presumptive*
They will be going to Murree.

(266) کل میں لاہور جا رہا ہوں گا

kal *mayN* *laahaor* *jaa* *rahaa* *hooN gaa*
Tomorrow, I Lahore go.*root* AUX.*cont* be.*sg.m.1.presumptive*
Tomorrow, I will be going to Lahore.

(267) $S_{dubitative/\ presumptive} \rightarrow NP_{SUBJ\text{-}nom}\ (NP_{OBJ\text{-}nom})\ V_{root}\ AUX_{chok\text{-}aa}\ AUX_{hao}\ AUX_{g\text{-}aa}$

$S_{dubitative/\ presumptive} \rightarrow NP_{SUBJ\text{-}nom}\ (NP_{OBJ\text{-}nom})\ V_{root}\ AUX_{rah\text{-}aa}\ AUX_{hao}\ AUX_{g\text{-}aa}$

8.3.9 Subjunctive Mood

The subjunctive mood, *moJaarA* (مضارع), is used to express feelings, opinions, suggestions and imaginary events. This subjunctive mood expresses both present and future tense. Its desiderative form, *tamanaaee* (تمنائی), expresses desires, hopes, and wishes. Its conditional form, *sharTeeah* (شرطیہ), expresses conditions. A general rule to form a subjunctive is shown in (268), which shows that subjunctive mood is expressed using a morphological verb-from and no auxiliary is used to express this mood. The example sentences are given in (269) and (270), which show that subjective verb-form requires agreement with the subject in number and person.

(268) $S_{subjunctive} \rightarrow NP_{SUBJ\text{-}nom}\ (NP_{OBJ\text{-}nom})\ V_{subjunctive\text{-}form}$

(269) وہ آئے

woh *aa-ey*
He=*sg.3.nom* come-*sg.3.subjunctive*
He/They (might) come.

(270) میں کتاب پڑھوں

mayN *ketaab* *paRh-ooN*
I=*sg.1.nom* book=*nom* read.*sg.1.subjunctive*
(I wish that) I read a book – or – (Should) I read a book?

The subjunctive mood is also used for praying to Allah for something, as shown in (271) and also for seeking forgiveness of sins from Allah as shown in (272).

(271) اللہ تمہیں بیٹا دے

aal.lah	*tomheyN*	*beytaa*	*d-ey*
Allah=*3.nom*	you=*dat.2*	son.*sg.masc*	give.*sg.3.subjunctive*

May Allah give you a son.

(272) اللہ میرے گناہ معاف کرے

aal.lah	*meyrey*	*gonaah*	*moAaf*	*kar-ey*
Allah=*3.nom*	I=*gen.1*	sin=*nom.pl*	forgive=*nom*	do.*pl.3.subjunctive*

May Allah forgive my sins.

8.4 Verbal Coordination in Urdu

The verbal coordination refers to the use of two or more verbs with a common subject of the sentence as shown in the sentence (273). In this sentence, there is a single ergative subject '*Haamed*' and three actions 'eating breakfast', 'picking up a bag' and 'going to an office'. These three actions are associated with the subject using a conjunction 'and (*aor* – اور)'. This coordinated structure is well-formed because all three verbs in the sentence are transitive and require an ergative subject.

(273) حامد نے ناشتہ کیا، بیگ اٹھایا اور دفتر چلا گیا

Haamed=ney	*naashtah*	*kee-aa* ,	*bayg*	*aoThaa-yaa* ,
Hamid=*erg*	breakfast=*nom*	do.*perf* ,	bag=*nom*	pick up.*perf* ,
aor	*daftar*	*chal-aa ga-yaa,*		
and	office=*nom*	go.*perf*		

Hamid ate the breakfast, picked up (his) bag and went to (his) office.

However, sometimes a coordinated sentence with many transitive verbs and a single intransitive verb between transitive verbs is considered good as shown in (274), where the intransitive verb '*nahaa-naa* – bath' is used with other transitive verbs.

(274) حامد نے ناشتہ کیا، نہایا، بیگ اٹھایا اور دفتر چلا گیا

? Haamed=ney	*naashtah*	*kee-aa* ,	*nahaa-yaa* ,	*bayg*
Hamid=*erg*	breakfast=*nom*	do.*perf* ,	bath.*perf* ,	bag=*nom*
aoThaa-yaa ,	*aor*	*daftar* ,	*chal-aa ga-yaa*	
pick up.*perf* ,	and	office=*nom* ,	go.*perf*	

Hamid ate breakfast, took bath, picked up bag and went to office.

The sentence (275) and (276) use a transitive verb '*khaa-naa*' and an intransitive verb '*nahaa-naa*' in coordination. The transitive verb requires an ergative subject, while intransitive verb requires a nominative subject. The perfective verb-form agrees with the object for a transitive verb and with the subject for an intransitive verb. For sentence (275), the transitive verb '*khaa-naa*' agrees correctly

with the object '*aam*', but the intransitive verb '*nahaa-naa*' cannot agree with the subject, because the subject is ergative, and the intransitive verb cannot take an object, moreover, the default agreement singular-masculine is also not correct. Therefore, the only way to assume that sentence (275) is well formed is to assume that there is an implied pronoun '*woh*', like the one shown in sentence (277). Similarly, the verb '*khaa-naa*' in sentence (276) requires an ergative subject and considered well-formed if the pronoun '*aos ney*' is assumed, like the one shown in sentence (278). However, such a 'verbal coordination' between an intransitive and a transitive verb is well formed for other imperfective verb-forms (i.e., infinitive, repetitive, imperative and subjunctive), because these verb-forms require nominative subject for both transitive and intransitive verbs.

(275) * نادیہ نے آم کھایا اور نہائی

** naadyah=ney aam khaa-yaa aor nahaa-ee*
Nadya=*erg.fem* mango=*nom.masc* eat.*perf.masc* and bath.*perf.fem*
Nadya ate mango and bathed.

(276) * نادیہ نہائی اور آم کھایا

** naadyah nahaa-ee aor aam khaa-yaa*
Nadya=*nom.fem* bath.*perf.fem* and mango=*nom.masc* eat.*perf.masc*
Nadya bathed and ate mango.

(277) نادیہ نے آم کھایا اور وہ نہائی

naadyah=ney aam khaa-yaa aor woh nahaa-ee
Nadya=*erg.fem* mango=*nom.masc* eat.*perf.masc* and she=*nom* bath.*perf.fem*
Nadya ate mango and bathed.

(278) نادیہ نہائی اور اُس نے آم کھایا

naadyah nahaa-ee aor aos=ney aam khaayaa
Nadya=*nom.fem* bath.*perf.fem* and she=*erg* mango=*nom.masc* eat.*perf.masc*
Nadya bathed and ate mango.

The sentences (277) and (278) do not express 'verbal coordination' because each verb has its own subject and these sentences represent the coordination between two sentential phrases instead of between two verbs. The 'verbal coordination' is usually achieved in Urdu using two 'verbal conjunction – فعلی عطف' types, which are preferred to the general conjunction 'and (*aor* – اور)', and both of these act as 'participle adverbials'. The first type 'perfective verbal conjunction' is formed, as shown in (279), by adding the auxiliary '*kar*' to the verb stem-form. The auxiliary '*kar*' describes that the subject, after completing the first action, performs other action. The '*kar*' phrase in Urdu has similar meaning as the 'having' phrase in English. The example sentences are shown in (280) and (281), and this syntax is linguistically preferable to the coordination syntax used in (277) and (278).

(279) An action follows after completing other action
Perfective Verbal Conjunction = Verb Stem + *kar*

(280) حامد آم کھا کر نہایا

Haamed	*aam*	*khaa*	*kar*	*nahaa-yaa*
Hamid=*nom.masc*	mango=*nom.masc*	eat.*root*	AUX.*perf.conj*	bath.*perf.sg.masc*

Hamid, having eaten the mango, bathed.
Hamid, after eating the mango, bathed.

(281) حامد نے نہا کر آم کھایا

Haamed=ney	*nahaa*	*kar*	*aam*	*khaa-yaa*
Hamid=*erg.masc*	bath.*root*	AUX.*perf.conj*	mango=*nom.sg.masc*	eat.*perf.sg.masc*

Hamid, having bathed, ate the mango.
Hamid, after bathing, ate the mango.

$$\begin{bmatrix} \text{PRED} & \text{' } \textit{nahaanaa} \langle \text{SUBJ} \rangle \text{ '} \\ \text{SUBJ} & \boxed{1} \begin{bmatrix} \text{PRED} & \text{' } \textit{Haamed} \text{ '} \\ \text{CASE} & \textit{nom} \end{bmatrix} \\ \text{TNS-ASP} & \begin{bmatrix} \text{TENSE} & \textit{past} \end{bmatrix} \\ \text{ADJUNCT} & \left\{ \begin{bmatrix} \text{PRED} & \text{' } \textit{khaanaa} \langle \text{SUBJ, OBJ} \rangle \text{ '} \\ \text{SUBJ} & \boxed{1} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{aam} \text{ '} \\ \text{CASE} & \textit{nom} \end{bmatrix} \\ \text{TNS-ASP} & \begin{bmatrix} \text{ASPECT} & \textit{perfective} \\ \text{MOOD} & \textit{conjunctive} \end{bmatrix} \end{bmatrix} \right\} \end{bmatrix}$$

Figure 8.25: F-Structures of '*Haamed aam khaa kar nahaayaa*'

$$\begin{bmatrix} \text{PRED} & \text{' } \textit{khaanaa} \langle \text{SUBJ, OBJ} \rangle \text{ '} \\ \text{SUBJ} & \boxed{1} \begin{bmatrix} \text{PRED} & \text{' } \textit{Haamed} \text{ '} \\ \text{CASE} & \textit{nom} \end{bmatrix} \\ \text{TNS-ASP} & \begin{bmatrix} \text{TENSE} & \textit{past} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{aam} \text{ '} \\ \text{CASE} & \textit{nom} \end{bmatrix} \\ \text{ADJUNCT} & \left\{ \begin{bmatrix} \text{PRED} & \text{' } \textit{nahaanaa} \langle \text{SUBJ} \rangle \text{ '} \\ \text{SUBJ} & \boxed{1} \\ \text{TNS-ASP} & \begin{bmatrix} \text{ASPECT} & \textit{perfective} \\ \text{MOOD} & \textit{conjunctive} \end{bmatrix} \end{bmatrix} \right\} \end{bmatrix}$$

Figure 8.26: F-Structures of '*Haamed ney nahaa kar aam khaayaa*'

The f-structures of sentences in (280) and (281) are shown in Figure 8.25 and Figure 8.26, which assign verbal conjunctive phrase an adverbial adjunct to the main phrase, such that the subject for both phrases is the same. The adverbial adjunct has perfective aspect and conjunctive mood.

The second type 'verbal conjunction progressive' represents the overlap of two actions, and one action is performed while the other action is in progress. This is formed by appending an oblique form of auxiliary '*hoo-ey*' to the oblique repetitive verb form, as shown in (282).

(282) An action accompanies another progressive action
Verbal Conjunction Progressive = Verb Repetitive Form *-tey* + *hoo-ey*

(283) حامد آم کھاتے ہوئے نہایا

Haamed	*aam*	*khaa-tey*	*hoo-ey*	*nahaa-yaa*
Hamid=*nom.masc*	mango=*nom.masc*	eat.*m.obl*	AUX.*m.obl*	bath.*perf.sg.masc*

Hamid, while eating the mango, bathed.

$$\begin{bmatrix} \text{PRED} & \text{' } \textit{nahaanaa} \langle \text{SUBJ} \rangle \text{ '} \\ \text{SUBJ} & \boxed{1} \begin{bmatrix} \text{PRED} & \text{' } \textit{Haamed} \text{ '} \\ \text{CASE} & \textit{nom} \end{bmatrix} \\ \text{TNS-ASP} & \begin{bmatrix} \text{TENSE} & \textit{past} \end{bmatrix} \\ \text{ADJUNCT} & \begin{Bmatrix} \text{PRED} & \text{' } \textit{khaanaa} \langle \text{SUBJ, OBJ} \rangle \text{ '} \\ \text{SUBJ} & \boxed{1} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{' } \textit{aam} \text{ '} \\ \text{CASE} & \textit{nom} \end{bmatrix} \\ \text{TNS-ASP} & \begin{bmatrix} \text{ASPECT} & \textit{progressive} \\ \text{MOOD} & \textit{conjunctive} \end{bmatrix} \end{Bmatrix} \end{bmatrix}$$

Figure 8.27: F-Structures of '*Haamed aam khaatey hooey nahaayaa*'

(284) Progressive Verbal Conjunction – Adverbial Participle
میں نے بھاگتے ہوئے لڑکا دیکھا

mayN=ney	*[bhaag-tey*	*hoo-ey]*	*laRk-aa*	*deykh-aa*
I=*erg*	run=*repeat.obl*	AUX.*obl*	boy-*sg.masc*	saw.*perf.sg.masc*

I, while running, saw a boy.

(285) Verbal Stative Noun – Adjective
میں نے بھاگتا ہوا لڑکا دیکھا

mayN=ney	*[bhaag-taa*	*hoo-aa]*	*laRk-aa*	*deykh-aa*
I=*erg*	run=*repeat.sg.masc*	AUX-*sg.masc*	boy-*sg.masc*	saw.*perf.sg.masc*

'I saw a running boy', i.e., I saw a boy, who was running.

This 'verbal conjunction progressive' is similar in construction to the 'verbal stative noun – اسمِ حالیہ', which is formed using various forms of auxiliary '*hoo-aa*' that agree in number and gender with the repetitive forms. Both have similar construction, but the 'verbal conjunction progressive' acts as an adverbial participle, does not require agreement, appears only in oblique-form and describes a simultaneous action performed by the subject, while 'verbal stative noun' acts as an adjective, requires agreement in number and gender like an adjective and appears for

both the subject and object. This contrast between the two is shown in sentences (284) and (285).

8.5 Conclusion

In this Chapter, the modeling of the verbal structure is presented for commonly used Urdu tenses, aspects and moods. Urdu language has rich verb morphology, which requires agreement with the subject and/or object nouns. The verb morphology as well as auxiliaries, describe various features for tense, aspect and mood. It is shown that the LFG based c-structures and f-structures can handle such diverse modeling requirements of Urdu verbal syntax.

Chapter 9

URDU PARSING BY CHUNKING BASED ON CLOSED WORD CLASSES USING ORDERED CONTEXT FREE GRAMMAR

Parsing is to find constituent structure of a sentence in a language using given grammar and it is an important requisite for various natural language processing applications. For machine translation systems, parsing the sentences of the source language is important for the proper syntactic and afterwards semantic understanding of the source language sentences. Unless the proper understanding of the structure of the source language is attained, the knowledge conveyed by the sentence may not be extracted and then reliable machine translation may not be achieved.

The parsing techniques for context free grammar (CFG), also known as rule-based parsing techniques, are well understood (Grune and Jacobs 1994). Various parsing techniques are available, each of which is suited to some particular conditions. The Earley parser, Tomita parser and chart parser are widely used for natural language parsing. However, to find a complete unambiguous CFG parsing for natural language processing (NLP) is a difficult task, until semantic disambiguation is done. The statistical parsing techniques for the natural language parsing are achieving good results (Manning and Schütze 2003). However, most of statistical techniques need a large corpus and a considerable amount of manual work input for tagging and parsing of corpus as an example data for statistical tagger and parser. Steven Abney proposed to approach parsing of natural languages by starting with finding correlated chunks of words (Abney 1991). Chunking is to divide text into syntactically related non-overlapping groups of words. Ramshaw and Marcus used chunking through a machine learning method (Ramshaw and Marcus 1995). Their approach categorized every non-NP chunk as VP chunk. Buchholz et al. found various chunks: NP, VP, PP, ADJP and ADVP (Buchholz, Veenstra et al. 1999). Veenstra worked with NP, VP and PP chunks (Veenstra 1999). In 2000, a conference on Computational Natural Language Learning (CoNLL)[1] "shared task" was held on chunk parsing. JMLR Special Issues[2] was published on shallow parsing in 2002.

[1] For CoNLL please see online: http://cnts.uia.ac.be/conll2000/chunking

[2] Available online: http://www.ai.mit.edu/projects/jmlr/papers/special/ shallowparsing.html

This Chapter explores parsing by chunking based on morphologically closed word classes in Urdu and using a novel Ordered Context Free Grammar (OCFG). The OCFG rules use the linguistic features of these Urdu words to make chunks of neighbor words. Full parsing is achieved after chunks of basic phrases have been made. Because the Lexical Functional Grammar (LFG) has been used for the representation of lexical and syntactic information, instead of a simple CFG, therefore, when chunking and parsing drive parse tree (i.e., c-structure), the unification is also performed at the same time to make f-structures. The constraint equations along with completeness, consistency and coherence conditions for the well formedness of f-structures are also helpful, during and after full parsing, to endorse the correctness of the parsing.

9.1 Ordered Context Free Grammar

This work presents a novel Ordered Context Free Grammar (OCFG), which is an extension of the context-free-grammar, and each rule OCFG has an order and type associated with it, just like probabilistic context-free-grammar is an extension of CFG (Yao and Lua 1998) and has a probability value associated with each rule. The formal definition is given in (286).

(286) **Definition**: An ordered context free grammar (OCFG) is a four tuple {W, N, S, R}, where W = {w_1, w_2, … , w_u} is a set of terminal symbols like words in a sentence, N = {n_1, n_2, … , n_v} is a set of non-terminal symbols like noun-phrase in a sentence, S = { n } is a set of one symbol, which is the goal symbol, and R = {r_1, r_2, … , r_w} is a set of grammar rule, where each rule has a unique order number and type (left, right or recursive)
(Rizvi and Hussain 2004).

Each rule ‘r’ has an order number associated with it, which is used as a priority during parsing. In addition to order, each rule has a left, right or recursive type. A left-rule is applied from left to right, and a right-rule is applied from right to left. A recursive rule is applied recursively. A rule can have neither empty left-hand side nor empty right-hand side. This means empty productions are not allowed.

In order to validate the proposed method a computer program is written and tested for arithmetic expressions, programming language and natural language parsing applications. In this section, parsing of arithmetic expression is presented to show that method not only can parse but also takes care of association involved in arithmetic expressions. In next sections, the OCFG will be used for ‘parsing by chunking’ for Urdu language.

This OCFG grammar for arithmetic expression is shown in (287). In the absence of order-number and rule-type, this grammar is normal CFG, which is ambiguous, and cannot be used for the parsing of arithmetic expressions (Aho, Sethi et al. 1986). However, OCFG based parsing is not ambiguous. Precedence of operators is handled by applying rule such that higher precedence operator is parsed ahead of lower precedence operator. This means that the expression having operator for multiplication '*' or division '/' will be evaluated ahead of expression involving operator for addition '+' or subtraction '–', because the rule for '*' or '/' has a lower order number. Associativity is handled by rule type: left or right. The sub-expression in braces is handled by recursive rule type.

(287)

E ← ID O3 E	0	Right/Recur
E ← ID	1	Right
E ← N	2	Right
E ← (E)	3	Left/Recur
E ← E O2 E	4	Left
E ← E O1 E	5	Left

The symbols used for the grammar in (287) are as follows:

ID	Any identifier that starts with a letter
N	Numeric Value
O1	+ (addition), – (subtraction)
O2	* (multiplication), / (division)
O3	= (assignment)

(288) A = B + C * 5.3
3 + 4 * (6 – 7) / 5 + 3

The expressions shown in (288) are parsed to explain the method. In the first step the input strings are converted to token objects. The parser sends tokens array to rules one by one, and rule object applies itself. The rule # 0 has recursion. Therefore, it finds the first two object ID and O3 and if found, it sends all tokens following O3 to parser object for finding the parse. The parser takes the shorter array and uses the same set of rules one by one to reduce it to E. The shorter array is shown using square brackets in Figure 9.1. After recursive call, the rules are applied on the array shown within square bracket, until it reduces to E, then recursion will return and parsing before recursion call is resumed. Now, by using rule # 2 from right-to-left the N object is used to construct E. Then using rule # 1 twice, two more E objects are created from ID objects. Then the rule # 4 is used ahead of rule # 5, which takes care of precedence requirement. Finally, the expression 'E' is constructed successfully within square bracket, where recursive call returns the parse in the previous array as an E object. Which by using rule # 0, constructs the final expression for assignment.

A = B + C * 5.3	Input expression
ID O3 ID O1 ID O2 N	Token objects
ID = [ID O1 ID O2 N]	Rule 0 : Recursion
ID = [ID O1 ID O2 E]	Rule 2
ID = [ID O1 E O2 E]	Rule 1
ID = [E O1 E O2 E]	Rule 1
ID = [E O1 E]	Rule 4
ID = [E]	Rule 5
ID = E	Recursion returns
E	Rule 0

Figure 9.1: Parsing of an Arithmetic Expression 'A = B + C * 5.3' using OCFG

The grammar (287) is applied to another expression, which has a sub-expression in brackets and its parse is shown in Figure 9.2. The sub-expression within braces is also handled using a recursive call, in order to ensure that the inner sub-expressions be evaluated ahead of out-side brace expression.

3 + 4 * (6 – 7) / 5 + 3	Input expression
N O1 N O2 (N O1 N) O2 N O1 N	Token objects
E O1 E O2 (E O1 E) O2 E O1 E	6 times, Rule 2
E O1 E O2 ([E O1 E]) O2 E O1 E	Rule 3 (Recursion)
E O1 E O2 ([E]) O2 E O1 E	Rule 5
E O1 E O2 (E) O2 E O1 E	Recursion returns
E O1 E O2 E O2 E O1 E	Rule 3
E O1 E O2 E O1 E	Rule 4 : Left
E O1 E O1 E	Rule 4
E O1 E	Rule 5 : Left
E	Rule 5

Figure 9.2: Parsing of an Arithmetic Expression '3+4*(6–7)/5+3' using OCFG

9.2 Tokenization

Before looking into chunking and parsing of Urdu sentences, the tokenization of Urdu text is discussed. Tokenization is a trivial task for most of the languages where space character is used to separate two lexical items. Urdu language has many lexical entries that contain a space, and therefore, the space character cannot be used to separate words. For example, consider the following sentences:

(289) ٹائر پنکچر ہو گیا ہے
Taaeyr pankchar hao gayaa hay.
The tyre has been punctured.

(290) میں ٹیلی فون خریدنے کے لیے گیا تھا
meyN Taylee phaon xareedney key leeey gayaa thaa.
I went for buying a telephone (set).

The character sequences: '*pankchar haonaa*', '*Teylee phaon*' and '*key leeey*' may be treated as a single word, which makes tokenization difficult. Alternatively, these may be composed at syntactic level, which makes syntactic-rules more complex. For example, the verbal-words such as '*pankchar haonaa*' and '*aenteezaar karnaa*' may be composed at syntactic level as N-V complex predicates, but handling them at syntactic level is not easy, because not all nouns can combine with all verbs. Such words with spaces are normally listed in dictionaries (Feroz-ud-Din 2000) as single words, and Urdu grammars (Mustafa 1973; Abdul-Haq 1991; Schmidt 1999) do no present syntax rules to compose them.

Unicode[1] character set and Urdu Zabta Takhti (UZT) character set (Afzal and Hussain 2001), both contain two types of spaces. In Unicode character set normal space is represented with hexadecimal value 0x0020, and another zero-width-non-joiner space has value 0x200C. In UZT, the second space is given name hard-space and represented using hexadecimal value 0x41. The function of both zero-width-non-joiner space and hard-space is to represent space in character sequence that represents single word. However, the current electronically available Urdu text uses only normal space, such as the books written in word processor 'Inpage', the newspaper Jang text, various Urdu books and websites using Pakistan data management system's Urdu word processor 'Urdu 98', the Unicode based text at BBC Urdu news website. Before tokenization of such a text, pre-processing of sentences in text is required. A simple algorithm for the tokenization, by replacing soft-space with hard-space, for an Urdu sentence is as follows:

0. given a sorted lexicon containing collocations (i.e., words with soft-spaces) such that the collocations having more spaces and greater length are on top
1. for each such collocation, do
2. search the collocation in the sentence
3. if found; replace normal-space with hard-space for this collocation
4. separate words in the sentence as token at normal-spaces

9.3 Part of Speech (POS) Tagging

Part of speech tagging, also known as POS tagging, is an important task to be carried out priror to doing chunking and parsing. POS tagging is the task of assigning to each word in the corpora a label, known as tag, to indicate the category of that word with respect to its morphological and/or syntactic variation. Andrew Hardie has proposed a tagset for the POS tagging of Urdu language (Hardie 2004) as part of

[1] For information about Unicode standard please see: http://www.unicode.org

EMILLE project[1], which assigns a separate tag to each morphosyntactic variation of a word, according to EAGLES guidelines (Leech and Wilson 1999). Hardie's tagset is more useful, if one wants to handle morphology and syntax using context-free-grammar rules for handling morphosyntactic variations. The LFG based approach followed in this thesis, is first define broad syntactic categories, then the morphological and few syntactic variations, associated with each of the lexical entries, are taken as attribute-values. Therefore, this reserach proposes a smaller tagset, primarily collected from Urdu grammar books (Platts 1884; Mustafa 1973; Abdul-Haq 1991; Schmidt 1999). The tags and the rules for each category, to be used for the chunking have been summarized in the next sub-sections.

Case Markers

Urdu case markers, as discussed in Chapter 7, mark core and oblique grammatical relations. These markers always follow a noun or a noun-phrase. Therefore, these may be used to make a case marked noun-phrase chunk.

Case Marker		Case	Tag
نے	*ney*	ergative	CM_ney
کو	*kao*	dative, accusative	CM_kao
سے	*sey*	agentive, instrumental, locative, …	CM_sey

Postpositions

Urdu postposition are different from case-markers, because these do not mark basic grammatical relation, and these are not controlled by the verbal predicate. These are adjuncts to the main sentence. These, like case-markers, always follow nouns or noun-phrases. Therefore, these may be used to make post-positional phrase chunk. As shown in the list of post-positions given below, that most of the post-positions are composed of two or more basic units, because these contain 'space' character. Therefore, tokenization and tagging of post-position, must take care of the 'space' issue, in order to achieve better chunking results.

Postposition		English Preposition	Tag
میں	*meyN*	in, into, at (place)	PP
پر	*par*	on	PP
سے	*sey*	from, by, …	PP
تک	*tak*	up to, till	PP
کے لیے	*key leeey*	for	PP
کے اندر	*key aandar*	within, contained in	PP

[1] For information about EMILLE project, please see: http://www.emille.lancs.ac.uk/about.php

Postposition		English Preposition	Tag
کے اُوپر	*key aoopar*	on, at, over	PP
کے باہر	*key baahar*	outside, out	PP
کے نیچے	*key neechay*	below, beneath, under	PP
سے اُوپر	*sey aoopar*	above	PP
کے پیچھے	*key peechhay*	behind	PP
کے پاس	*key paas*	near to	PP
کے بعد	*key baAd*	after	PP
سے پہلے	*sey pahley*	before	PP
کے ساتھ	*key saath*	with, beside, along	PP
کے مطابق	*key mottaabeq*	according to	PP
کے متعلق	*key motaAleq*	about	PP
کے خلاف	*key xelaaf*	against	PP
کے سوا	*key sewwaa*	except	PP
کی طرح	*kee ttaraH*	like, similar to	PP
کی طرف	*kee ttaraf*	to, toward	PP
کے بارے میں	*key baarey meyN*	about	PP
کی جگہ	*kee jagah*	in place of	PP
کے علاوہ	*key Aelaawah*	apart from	PP
کے طور پر	*key ttaor par*	as an alternate for	PP
کی وجہ سے	*kee wajah sey*	caused by	PP
کے ذریعے	*key ZareeAey sey*	through, by means of	PP
کے سبب	*key sabab*	due to	PP
کے قریب	*key qareeb*	near to	PP
کے درمیان	*key darmeyaan*	among, between	PP
سے باہر	*sey baahar*	beyond	PP
کے دوران	*key daoraan*	during	PP
کی خاطر	*kee xaatter*	for	PP
کے سامنے	*key saamney*	opposite to	PP
سے زیادہ	*sey Zeyaadah*	more than	PP

Possession Markers

Urdu possession markers come in between two nouns or noun-phrases and these describe possessive relation. Therefore, whenever the possession markers are found in a text, there is a high probability that there will be two nouns or noun-phrases around them. These may, therefore, be used to make a possessive noun-phrase chunk. It may also be noted that many postpositions, contain '*kee*' and '*key*', which are treated differently from these possession markers. The easiest way to handle this ambiguity is that first postpostion are tagged and then remaining '*kaa*', '*kee*' and '*key*' are tagged as possession markers. However, possessive phrase chunks are made ahead of postpositional or case marker phrase chunks.

Possession Marker			Tag
کا	*kaa*	of	PM
کی	*kee*	of	PM
کے	*key*	of	PM

Conjunctions

A conjunction is a part of speech that connects two words, phrases, or clauses together. **Coordinating conjunctions**, also called coordinators, are conjunctions that join two or more items of the same syntactic category. More frequently, Coordinating conjunctions can be used to make chunks of noun-phrases or to make chunks of separate sentences. However, sometimes these can also be used to break apart verbs and adjective phrases.

Coordinating Conjunctions		English	Tag
اور	*aaor*	and	CJC
یا	*yaa*	or	CJC

Subordinating conjunctions, also called subordinators, are conjunctions that introduce a dependent clause. These are mainly used to break apart large sentence into smaller clause chunks, which are subsequently easier to parse.

Subordinating Conjunctions		English	Tag
البته	*aalbatah*	however	CJS
مگر	*magar*	but	CJS
لیکن	*leyken*	nevertheless	CJS
لہذا	*leyhaaZaa*	therefore	CJS
اس لیے	*aes leeey*	consequently	CJS
تو	*tao*	then	CJS
تب ہی	*tab hee*	after that	CJS
پھربھی	*pher bhee*	yet, despite, in spite of	CJS
کہ	*keh*	that	CJS
تا کہ	*taa keh*	for the reason	CJS
حالانکہ	*Haalaankeh*	in spite of this situation	CJS
جبکہ	*jabkeh*	whereas	CJS
کیونکہ	*keeonkeh*	because	CJS

Correlative conjunctions are pairs of conjunctions which work together to coordinate two phrases or clauses. These are, likewise, used to break apart larger sentence into smaller chunks, which are subsequently easier to parse.

Correlative Conjunctions		English Equivalent		Tag
تو	اگر	if	then	CJR1, CJR2
وہ، سو	جو	whoever	he, she, it	CJR1, CJR2
اُس	جس	whomever	he, she	CJR1, CJR2
وہ	جن	whom, whose	they	CJR1, CJR2
لہٰذا	چونکہ	because	therefore	CJR1, CJR2
یا	یا تو	either	or	CJR1, CJR2
نہ ہی	نہ، نہ تو	neither	nor	CJR1, CJR2
بلکہ	نہ صرف	not only	but also	CJR1, CJR2
ویسا، ویسی، ویسے	جیسا، جیسی، جیسے	whatever	the same	CJR1, CJR2
اُتنا، اُتنی، اُتنے	جتنا، جتنی، جتنے	whatever, as much	the same	CJR1, CJR2
وہاں	جہاں	where	there	CJR1, CJR2
اُدھر	جدھر	wherever	there	CJR1, CJR2
تب	جب	when	then	CJR1, CJR2
اُس وقت تک	جب تک	until	–	CJR1, CJR2
تب	جب کبھی	whenever	then	CJR1, CJR2

Interjections

An interjection is a word added to a sentence to convey emotion. It is not grammatically related to any other part of the sentence.

Interjections		Tag
انشاءاللہ	*aenshaa aallah*	IJ
ماشاءاللہ	*maashaa aallah*	IJ
شاباش	*shaabaash*	IJ
شاندار	*shaandaar*	IJ
او	*aoo*	IJ
اے	*aey*	IJ
اوہ	*aooh*	IJ
اُف	*aof*	IJ
واہ	*waah*	IJ
خوب	*xoob*	IJ
ارے	*aarey*	IJ

Pronouns

The word used instead of a noun is called a pronoun. The ‘subject pronouns’ are those that stand for three persons forms, when these forms take the position of the subjects. The subject pronoun forms may appear in nominative and ergative cases, which means that these may be used with ergative marker ‘*ney*’ or these may appear alone in the nominative case.

Subject Pronouns		English	Tag
میں	*meyN*	I	PNS
ہم	*ham*	we	PNS
تو	*too*	you	PNS
تم	*tom*	you	PNS
آپ	*aap*	you	PNS
وہ	*woh*	he, she, they, it (far)	PNS
یہ	*yeh*	it (near)	PNS
اِس	*aes*	he, she, this	PNS
اُس	*aos*	he, she, that	PNS
اِن	*aen*	they, these	PNS
اُن	*aon*	they, those	PNS

The ‘object pronouns’ are those that are used in place of three persons, when these appear as the object. These pronoun forms cannot appear with other cases markers like ergative or agentive markers, because these already bear dative or accusative case.

Object Pronouns		English	Tag
مجھے	*mojhey*	me	PNO
ہمیں، ہم کو	*hameyN, ham kao*	us	PNO
تجھے	*tojhey*	you	PNO
تمھیں، تم کو	*tomheyN, tom kao*	you	PNO
آپ کو	*aap kao*	you	PNO
اِسے، اِس کو	*aesey, aes kao*	him, her, it	PNO
اُسے، اُس کو	*aosey, aos kao*	him, her, it	PNO
اِنہیں، اِن کو	*aenheyN*	them	PNO
اُنہیں، اُن کو	*aonheyN*	them	PNO

The ‘possessive pronouns’ are those that are used in place of nouns to show possessive relationship, these pronouns must be followed by a noun or a noun-phrase to complete the possessive relationship.

Possessive Pronouns		English	Tag
میرا، میری، میرے	*meyraa, meyree, meyrey*	my, mine	PNP
ہمارا، ہماری، ہمارے	*hamaaraa, hamaaree, hamaarey*	our, ours	PNP
تیرا، تیری، تیرے	*teyraa, teyree, teyrey*	your, yours	PNP
تمھارا، تمھاری، تمھارے	*tomhaaraa, tomhaaree, tomhaarey*	your, yours	PNP
اِس کا، اِس کی، اِس کے	*aes kaa, aes kee, aes key*	his, her, hers, its	PNP
اُس کا، اُس کی، اُس کے	*aos kaa, aos kee, aos key*	his, her, hers, its	PNP
اِن کا، اِن کی، اِن کے	*aen kaa, aen kee, aen key*	his, her, hers, its	PNP
اُن کا، اُن کی، اُن کے	*aon kaa, aon kee, aon key*	his, her, hers, its	PNP

A ‘reflexive pronoun’ is a pronoun that is preceded by the noun or pronoun to which it refers (its antecedent) within the same clause.

Reflexive Pronoun		English	Tag
اپنے آپ کو	*aapney aap kao*	myself, ourselves, himself, herslef	PNR
خود	*xaod*	myself, ourselves, himself, herslef	PNR
آپس	*aapas*	themselves	PNR
ایک دوسرے	*aeyk doosrey*	one another	PNR

An ‘indefinite pronoun’ is a pronoun that does not refer to a specific person, place or thing.

Indefinite Pronoun		English	Tag
سَب	*sab*	all	PNI
کوئی	*kaoee*	anyone, anybody, anything	PNI
کچھ	*kochh*	some, something, few	PNI
یہاں،وہاں	*yahaan, wahaan*	here, there	PNI
اِدھر،اُدھر	*aedhar, aodhar*	here, there	PNI
ہر ایک	*har aeyk*	each, everyone, everybody	PNI
دونوں	*daonaoN*	both	PNI
کئی	*kaee*	many, serveral	PNI
باقی	*baaqee*	others	PNI

Negation Markers

The negation markers are used to make negative senteces. These are like adverbs, as these add negation to the meaning of actual sentence.

Negation Markers		English	Tag
نہ	*nah*	no	NM
نہیں	*naheeN*	no, nil	NM
مَت	*mat*	no	NM
کبھی نہیں	*kabhee naheeN*	never	NM

Question Markers (k-words)

The qeustion markers are used to make interrogative senteces. These k-words add question to the meaning of actual sentence.

Question Markers		English	Tag
کیا	*keyaa*	what	QM
کیوں	*keeooN*	why	QM
کیسے	*keysey*	how	QM
کون	*kaon*	who	QM

Question Markers		English	Tag
کب	*kab*	when	QM
کہاں	*kahaaN*	where	QM
کِدھر	*kedhar*	where	QM
کِس جگہ	*kes jagah*	where	QM
کِس کا، کِس کی، کِس کے	*kes kaa, kes kee, kes key*	whose	QM
کِس لیے	*kes leeey*	for what	QM
کِس وقت	*kes waqt*	at what time	QM
کِس طرف	*kes ttaraf*	in which direction	QM
کِس سمت	*kes semmat*	in which direction	QM

Auxiliaries

An auxiliary, also known as helping verb, auxiliary verb, or verbal auxiliary, is a verb functioning to give further semantic or syntactic information about the main or full verb in a sentence. The auxiliaries in Urdu convey tense, aspect and mood information, as described in Chapter 8.

Auxiliaries			Tag
ہے، ہو، ہوں، ہیں	*hay, hao, hooN, hayN*	be (present), is, are	AUX
تھا، تھی، تھے	*thaa, thee, they*	be (past), was, were	AUX
گا، گی، گے	*gaa, gee, gey*	will, shall	AUX
چکا، چکی، چکے	*chokaa, chokee, chokey*	Perfective Aspect	APA
لیا، لی، لے	*leeaa, lee, ley*	Perfective Aspect	APA
رہا، رہی، رہے	*rahaa, rahee, rahey*	Progressive Aspect	APrA
رہتا، رہتی، رہتے	*rahtaa, rahtee, rahtey*	Repetitive Aspect	ARA
کرتا، کرتی، کرتے	*kartaa, kartee, kartey*	Repetitive Aspect	ARA
لگا، لگی، لگے	*lagaa, lagee, lagey*	Inceptive Aspect	AIA
والا، والی، والے	*walaa, walee, waley*	Inceptive Aspect	AIA
پڑا، پڑی، پڑے	*paRaa, paRee, paRey*	Compulsive Mood	ACoM
سکا، سکی، سکے	*sakaa, sakee, sakey*	Capacitive Mood	ACaM
سکتا، سکتی، سکتے	*saktaa, saktee, saktey*	Capacitive Mood	ACaM
چاہیئے	*chaaheeey*	Suggestive Mood	ASM
ہوا، ہوئی، ہوئے	*hooaa, hooee, hooey*	Declarative Mood (happen)	ADM
دینا (مصدر)	VN–*ney* + *deynaa*	Permissive Mood	APeM
منع کرنا (مصدر)	*manaA karnaa*	Prohibitive Mood	APrM

Numbers, Date and Time, Currency

The cardinal numbers and ordinal numbers are open class words and finite state morphology may be used to generate these numbers. These numbers follow adjectives or noun phrases. However, names of days, names of weeks, and other time and date

nouns may be treated as close class of words. Similarly, country names, city names, currencies may be made a close class of words.

ایک، دو، تین، ----	cardinal numbers	NC
پہلا، دوسرا، تیسرا، ---	ordinal numbers	NO

Focus and Topic Markers

ہی	*hee*	only	FM
بھی	*bhee*	also	TM

Titles

Title		Tag
صدر	President	TLE
صدرِ پاکستان	President of Pakistan	TLE
وزیرِاعظم	Prime Minister	TLE

Verbal Morphemes

In Chapter 7, the two proposals to compose verb phrase were given, i.e., the verb phrase, VP, may be composed by combining a verb form, V, with various auxiliaries, AUX, or alternatively it may be composed by combining verb base, VB, with the verbal morphome, VM, which contains all the auxiliaries lumped into actual morpheme. These VM may be treated as the closed classes of words, because these can be attached to many different VB's. A list of few VM is shown as follows:

taa hooN	*ee*	*ee thaa*	*chokey hao*	*ao gey*
tee hoon	*aa*	*aa thee*	*chokee hao*	*ao gee*
tey hayN	*eeN*	*ee theeN*	*chokaa hay*	*ey gaa*
tee hayN	*ey*	*ey thay*	*chokee hay*	*ey gee*
taa hay	*ee hay*	*chokaa hooN*	*ooN gaa*	*eyN gey*
tee hay	*aa hay*	*chokee hooN*	*ooN gee*	*eyN gee*
tey hao	*ee hayN*	*chokey hayN*	*eyN gey*	*ao*
tee hao	*ey hayN*	*chokee hayN*	*eyN gee*	*ooN*

9.4 Chunking

The morphologically closed word classes have been discussed in the previous section. Case markers, possession markers are used to make noun phrase chunk, NP. Postpositions are used to make postpositional phrase chunk, PPP. The verbal morphemes and auxiliaries are used to make verb phrase chunk, VP. The conjunctions are used to make recursive rules for breaking apart larger sentences into smaller sentences. The interjections, negation markers, question k-words, are are not directly

useful in chunking and these are used as adjunct phrase, AJP, in the main sentence. Therefore, closed word classes are found quite useful as an aid for chunking based on their linguistic characteristics.

The chunking scheme presented in this research utilizes an ordered context free grammar (OCFG) presented in the earlier sections of this chapter. List of recursive chunking OCFG rules is given in (291), while non-recursive OCFG rules are given in (292).

(291) 1 S ← S CJC S

2 S ← S CJS S

3 S ← CJR1 S CJR2 S

(292) 4 NP ← PNS

5 NP ← (TLE | NO | NC) (Adj) N

6 PPP ← (N | NP) PP

7 NP ← PNP (N | NP)

8 NP ← (PNS | N | NP) PM (N | NP)

9 NP ← (PNS | NP) CM_ney

10 NP ← (PN | NP) CM_kao

11 NP ← (PN | NP) CM_sey

12 NP ← (NP,) NP CJC NP

13 V_1 ← V (AUX)*

14 V_2 ← VB VM

15 AJP ← Adv | PPP

After chunking of longer sentences into smaller sentences is achieved by using rules in (291), then for the smaller sentences, NP, PPP and VP chunks are made using the rules in (292). After chunking is complete, the parsing rules shown in (293) are used to achieve full parsing.

(293) 16 S ← NP (NP) V_{INF} (AIA | ACoM) AUX

17 S ← NP (NP) V_{ROOT} (APrA | APA) AUX

18 S ← NP (NP) V_{REPEAT} ARA AUX

19 S ← (AJP) (NP_{NOM}) (NP_{INFO} | NP_{LOC}) (AUX | ADM)

20 S ← (AJP)* NP (AJP)* (NP)* (AJP)* (V_1 | V_2) (AJP)*

9.5 Algorithm for Parsing through Chunking

The outlines of the algorithm adopted for parsing by chunking and then using ordered context free grammar is as follows:

1. Tokenize sentence into words.
2. Tag tokens by starting with morphologically closed classes of words.
3. Tag remaining tokens with morphologically open classes of words, by using linguistic guess from already tagged closed class tokens.
4. Apply chunking rules in the given order to make chunks.
5. Use parsing rules on 'tagged and chunked' sentence to achieve full parsing.

9.6 Parsing by Chunking: Illustrative Examples

To illustrate the basic steps involved in the working of the algorithm, we present parsing of the Urdu sentence (294):

(294) وہ اپنی بہن کے گھر جا رہی ہے
woh aapnee behen key ghar jaa rahee hay.
She is going to her sister's house.

Tokenization of sentence results in:

woh	*aapnee*	*behen*	*key*	*ghar*	*jaa*	*rahee hay*

Tagging with closed classes results finds a pronoun (PNS) '*woh*', a possessive pronoun (PNP) '*aapnee*', a possession marker (PM) and a verb morpheme (VM) '*rahee hay*':

woh	*aapnee*	*behen*	*key*	*ghar*	*jaa*	*rahee hay*
PNS	PNP		PM			VM

Tagging with open classes using knowledgeable guess using linguistic knowledge is done next. The possessive postposition '*key*' must follow a noun, search through lexicon shows that '*behen*' is a noun. Verb morpheme must follow a verb base 'jaa'. The remaining one word must be a noun that follows possessive postposition. The lexical entries confirm guesses and the result is as follows:

woh	*aapnee*	*behen*	*key*	*ghar*	*jaa*	*rahee hay*
PNS	PNP	N	PM	N	VB	VM

Chunking rules are applied next. The PNP is combined with the following noun to form a noun phrase (NP) using rule 7 and then rule 8 is used to form another possessive noun phrase. The VB and VM are combined to form verb phrase by using rule 12. The result of chunking is:

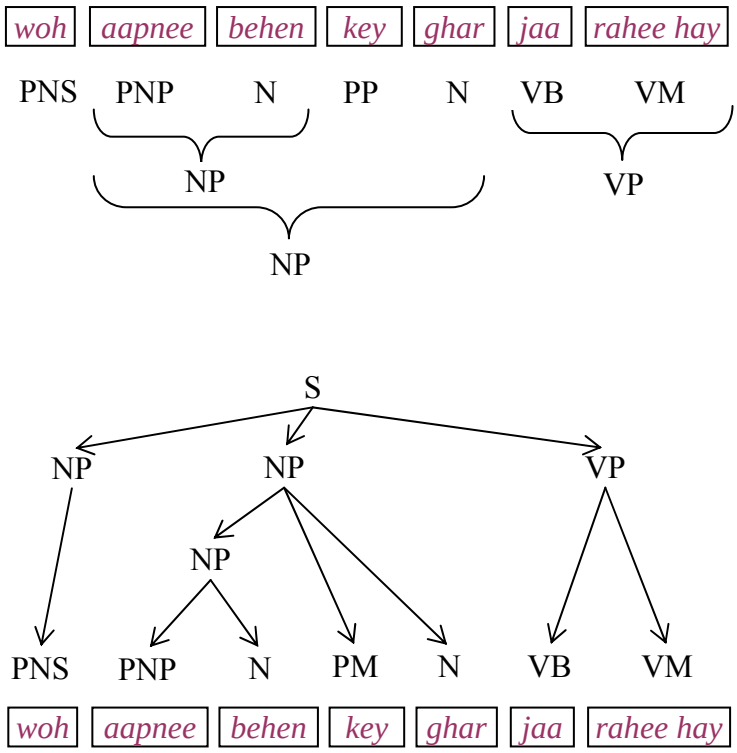


Figure 9.3: Parse Tree of Sentence '*woh aapnee behen key ghar jaa rahee hay*'

After chunking, the remaining are only three tokens, NP, NP and VP. Therefore, now parsing is achieved by using rule 20, this parse would have been difficult in the absence of chunking. The Parse Tree is produced as shown in Figure 9.3.

(295) کمرے میں تختہ سیاہ، میز اور کرسی ہے
kamrey meyN takhtah seeah, meyz aaor korsee hay
There is a blackboard, a table and a chair in the room.

To explain the working of the parsing by chunking method, we analyze another sentence shown in (295). Tokenization of sentence results in seven tokens as shown below. Actually there are eighth token including the comma, which is not shown here. The only token having space in it is '*takhtah seeah*':

کمرے	میں	تختہ سیاہ	میز	اور	کرسی	ہے
kamrey	*meyN*	*takhtah seeah*	*meyz*	*aaor*	*korsee*	*hay*

Tagging with closed classes finds that only three tokens belongs to this class, i.e., a coordinating conjunction (CJC) '*aaor*', a postposition (PP) '*meyN*' and a verb auxiliary (AUX) '*hay*'. The rest of tokens are not found in the lexicon portion representing closed classes.:

kamrey	*meyN*	*takhtah seeah*	*meyz*	*aaor*	*korsee*	*hay*
	PP			CJC		AUX

Tagging with open classes using knowledgeable guess is done next. We know that PP must follow a noun, N, and a conjunction must make a list of similar tokens. When the length of the sentence is small and CJC is surrounded with a list of nouns, then the result of breaking sentence into smaller sentence is false and in that condition rule13 is used. The lexicon search finds four nouns:

kamrey	*meyN*	*takhtah seeah*	*meyz*	*aaor*	*korsee*	*hay*
N	PP	N	N	CJC	N	AUX

Next after tagging is finished, chunking rules are applied. This one PPP and one NP chunk, by using rule 6 and rule 13, respectively, as shown:

kamrey	*meyN*	*takhtah seeah*	*meyz*	*aaor*	*korsee*	*hay*
N	PP	N	N	CJC	N	AUX
PPP		NP				

After chunking, the resulting three tokens are two noun phrases (NP)'s followed by a verb auxiliary AUX. A parsing rule 19 is used to generate complete parse of the sentence. Tree produced is shown in Figure 9.4.

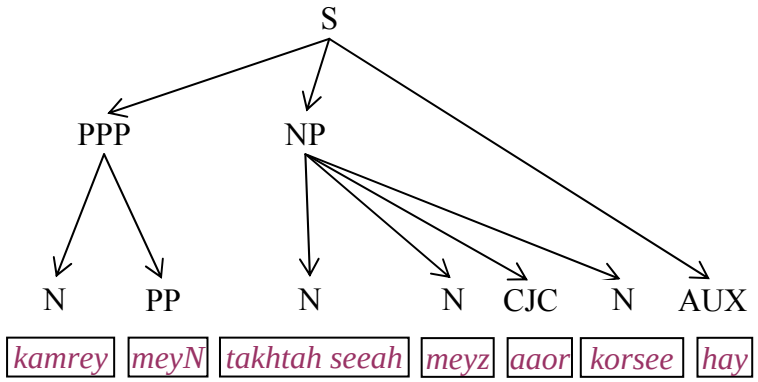


Figure 9.4: Final Parse Tree of Sentence '*kamrey meyN takhtah seeah, meyz aaor korsee hay*'

9.6.1 Handling Longer Sentences

In this section, it is shown that how parsing by chunking method can be used to parse longer sentences taken from news websites, such as newspaper Jang (http://www.jang.net) and BBC (http://www.bbc.co.uk/urdu). A sample sentence taken from newspaper Jang is shown here:

وزیراعظم پاکستان شوکت عزیز نے انٹرنیشنل کرکٹ کونسل کویقین دہانی کرائی ہے کہ اگر پاکستان کو بھارت کی جگہ آئی سی سی چیمپئنز ٹرافی کی میزبانی مل جائے تو حکومت پاکستان آئی سی سی کو ٹیکس میں چھوٹ دے گی

After transliteration it is:

wazeer e aAzzam paakestaan shaokat Aazeez ney aenTarneyshnal karekeT kaonsal kao yaqeen dahaanee karaaee hay keh aagar paakestaan kao bhaarat kee jagah aaee see see chaympeeanz Taraafee kee meyzbaanee mel jaaey tao Hakoomat e paakestaan aaee see see kao teyks meyN chhooT dey gee

Recursive chunking rule (S ← S CJS S) separates this bigger sentences into two smaller chunks by separating sentence at subordinating conjunction, CJS, '*keh* – that', thus the result is two smaller sentences as shown below:

S	*wazeer e aAzzam paakestaan shaokat Aazeez ney aenTarneyshnal karekeT kaonsal kao yaqeen dahaanee karaaee hay*
CJS	*keh*
S	*aagar paakestaan kao bhaarat kee jagah aaee see see chaympeeanz Taraafee kee meyzbaanee mel jaaey tao Hakoomat e paakestaan aaee see see kao teyks meyN chhooT dey gee*

Recursive chunking rule (S ← CJR1 S CJR2 S) separates the second part of above sentences into two more smaller chunks by separating sentence at pair of correlative conjunctions, CJR1, '*agar* – if', and , CJR2, '*tao* – then', thus the result is the smaller chunks as shown below:

S	*wazeer e aAzzam paakestaan shaokat Aazeez ney aenTarneyshnal karekeT kaonsal kao yaqeen dahaanee karaaee hay*
CJS	*keh*
CJR1	*aagar*
S	*paakestaan kao bhaarat kee jagah aaee see see chaympeeanz Taraafee kee meyzbaanee mel jaaey*
CJR2	*tao*
S	*Hakoomat e paakestaan aaee see see kao teyks meyN chhooT dey gee*

Now by applying the chunking rules for postpositional phrase, PPs, noun phrases, NPs, and for verb phrases VPs, the resultant chunks of the original sentence are shown as follows:

S	NP	NP	TLE [*wazeer e aAzzam paakestaan*] N [*shaokat Aazeez*]
		CM_erg	CM_erg [*ney*]
	NP	N [*aenTarneyshnal karekeT kaonsal*] CM_kao [*kao*]	
	VP	VB [*yaqeen dahaanee kar* VM [*aaee hay*]	
CJS	*keh*		
CJR1	*aagar*		
S	NP	N [*paakestaan*] CM_kao [*kao*]	
	PPP	N [*bhaarat*] PP [*kee jagah*]	
	NP	N [*aaee see see chaympeeanz Taraafee*] PM [*kee*] N [*meyzbaanee*]	
	VP	VB [*mel jaa*] VM [*ey*]	
CJR2	*tao*		
S	NP	N [*Hakoomat e paakestaan*]	
	NP	N [*aaee see see*] CM_kao [*kao*]	
	PPP	N [*Teyks*] PP [*meyN*]	
	VP	VB [*chhooT dey*] VM [*gee*]	

The major limitation for adopting this algorithm for general parsing is to tackle compound nouns containing spaces. In this research, these compound nouns, like '*aenTarneyshnal karekeT kaonsal*', '*Hakoomat e paakestaan*', '*aaee see see chaympeeanz Taraafee*', etc. having been included in the lexicon with 'hard spaces', so that these are treated as single nouns. Similarly, the title '*wazeer e aAzzam paakestaan*' has been treated as a single word in the lexicon.

9.7 Results and Analysis

The method presented is tested on various Urdu sentence categories like declarative, interrogative, negative, permissive, etc. taken from text books and news websites. The list of sentences is given in Appendix C, alongwith chunking rules used for each of the sentence shown. The parsing results are shown in Table 9.1.

Table 9.1: Parsing by Chunking Results

Sentence Set	Successful Parses
Basic Sentences	85%
News Website Sentences	75%

The method has been tested on the sentences given in the Appendix C and on all the example sentences given in this document. Please note that compound words are manually added in lexicon as single token. The given results may not be considered precise, until the method is tested on a standard large corpus of sentences and after a better tokenization algorithm is developed. For news website sentences, manual tokenization of compound nouns and their inclusion in the lexicon is performed before the processes of chunking and parsing are carried out.

The chunking method makes NP, PPP and VP chunks based on case markers, possession markers, postpositions and verbal auxiliaries and morphemes. It utilizes broad POS categories divided into closed and open class words. The open class words are collected in separate files to reduce the search space. Although, tagging closed class words ahead of open class words requires two passes on the input token array, but it results in 50% improvement in efficiency, because the search space is reduced by using separate files for closed class words, nouns, verbs and adjectives. Moreover, closed class words provide useful hint, like for a case marker, its predecessor must be a noun, therefore searching is required only for a noun. Moreover, after NP, PPP and VP chunks have been formed, the remaining parsing rules are less and simpler.

The parsing method presented does not require calculations and storage requirements for finding a parsing table, which is required in tabular parsing methods. Neither it generates all the possible parses of the given sentence, as generated by some methods, e.g., chart parser. It needs to store only the ‘n’ token objects in an array and ‘m’ rule objects. Therefore, space efficiency of this method may be considered good.

9.8 Conclusions

The language oriented parsing method presented in this Chapter for Urdu language through the mechanism of chunking utilizes linguistic characteristics of the morphologically closed word classes in Urdu language to make chunks. The simple tokenization algorithm presented in this chapter, manually includes compound Urdu words with soft spaces into lexicon. However, a better tokenization algorithm is needed to be developed for Urdu based script. Tagging is initiated with closed classes of words, which not only reduces search space, but also useful in guessing and chunking neighbor open class words through linguistics characteristics. The chunking results in a shallow parsing of the sentence and reduces number of rules for the final parsing stage. Proper identification of NP, PPP and VP phrases through chunking also results in the reduction in ambiguity for most of the sentences containing case markers and postpositions. However, for declarative (or news) mood sentences lexical ambiguity is not resolved. Reduction in ambiguity in natural language parsing results in more reliable machine translation.

The method generates only one parse tree for a given sentence, therefore, lexical and syntactic ambiguous sentences for which more than one parses are acceptable may not be handled by this method. Moreover, this method shows poor results for verbal conjunctions and also for sentences having long distance dependencies. To improve the accuracy of the method it is suggested that LFG feature based unification during parsing may be carried out to make sure proper agreement. Alternatively, some statistical technique may also be adopted for the tagging and chunking of Urdu text.

Chapter 10

CONCLUSIONS

10.1 Summary and Conclusions

The work has been done on the modeling of computational grammar for the formation of Urdu words and sentences. The frequently used constructions in Urdu have been investigated under the framework of Lexical Functional Grammar (LFG) and proposals have been presented for handling Urdu specific issues. The grammar formulation proposed in this work can be utilized for many natural language processing applications, such as, grammar checker, machine translation, text summarization, text categorization, information extraction, speech processing and knowledge engineering.

The morphological analysis of verbs, nouns and adjectives has been performed and implemented using Xerox finite state lexicon compiler 'LEXC' and Xerox finite state tool 'XFST'. The 'XFST' is a morphological analysis tool, which is useful for analyzing the lexical data and morphological information, and it builds a finite state network usually referred to as a 'lexical transducer'. The lexical transducer 'looks-up' surface morphological form of a word into a lexicon and finds lexical form of a word and 'looks-down' lexical word and gives corresponding morphological form.

For the syntactical analysis, most frequently used sentence constructions in Urdu have been modeled. Mainly, Lexical Functional Grammar 'LFG' has been used for the mathematical formulation of Urdu grammar, the implementation of which has been carried out using Xerox Linguistic Environment 'XLE'. For the development and testing of language grammars, XLE is a useful tool, which can be used to incorporate morphological analysis from XFST, and by implementing syntactic rules the parsing of sentences into c-structures and f-structures is achieved. Some Urdu syntactic concepts has also been modeled under Head-driven Phrase Structure Grammar 'HPSG', which also serve as a comparison between LFG and HPSG.

Urdu verb has a rich morphology and its verb forms can be divided into five categories, i.e., infinitive, perfective, repetitive, subjunctive and imperative. Urdu has three stem forms named as the root form, the causative form 1 and the causative form 2. Each of these three stem forms, by the attachment of various morphemes, results in 20 verb forms, making a total of 60 verb forms for a single Urdu verb. A finite-state-

automaton has been presented in Chapter 3 to represent these 60 forms. The attributes and corresponding value sets have been selected for representing verbal information in Urdu Lexicon. As in most languages, these attributes are person, gender, number, tense, aspect, mood, verb form, and honor form. As compared with English, the 'honor form' attribute for imperative verb forms is additionally required in Urdu, and similarly verb form, mood and aspect attribute in Urdu have some different values.

Urdu nouns and adjective morphology has been investigated and the attributes necessary to represent lexical information relating to nouns and adjectives have been collected. A noun in Urdu bears a 'gender' attribute for all nouns, which can take either 'feminine' or 'masculine' value, unlike English, which does not have such an attribute for inanimate nouns and unlike German, which take additional 'neutral' value. However, only some nouns in Urdu have overt gender morpheme, therefore for most of the Urdu nouns 'gender' attribute is required to be adopted from traditional dictionaries. The nouns have nominative form if they appear without a case-marker or post-position, have oblique form if they appear with a case-marker or post-position and have vocative form in subjunctive mood. Again, not all nouns have visible morpheme to distinguish nominative, oblique and vocative forms. The adjectives also have 'gender', 'number' and 'form' morphemes, which require agreement with the noun. The attributes required to represent lexical information related to various noun categories or characteristics and corresponding values they take have been collected. The semantic class of the noun, which tells the type of noun, i.e., animate, instrument, location, etc. is also selected, which is found useful in classifying different cases.

The review and implementation of algorithms for constructing a computational lexicon has been carried out. Some hash functions have been implemented for constructing a lexicon without morphological considerations. Similarly, some deterministic-finite-state-automaton minimization algorithms have been implemented to construct lexicon using 'lexical transducers'. A comparison between the two approaches showed that hash table implementation requires more memory space, however, it has fast access time, requires lesser morphological knowledge and is more dynamically adjustable, while lexical transducers based implementation requires morphological analysis, lesser space in memory and it has fast access time.

The formulation of the noun-phrase syntax in Urdu has been carried out. As Urdu is rich case-marked language, therefore nouns accompany various case-markers and post-position to form phrase that fill various grammatical roles in the argument structure of the verb. To better differentiate various roles adopted by noun-phrases a classification of case-markers and post-position has been proposed. This classification is based on the difference in modeling and conceptualization, such as whether a noun phrase should be handled morphologically or syntactically, whether it should be

controlled by verb's argument-structure or not, whether it should be attached to a core function or an oblique function. To resolve some of the ambiguities, the semantic class of nouns, such as animate, instrumental, location, etc. is employed. Similarly, to distinguish the various roles represented by the case marker '*sey*', the noun's semantic class has been found useful. It has been proposed that the possession-markers require different formulation from the case-markers, because they require two noun phrases, i.e., the possessor and the possessee, they require agreement in 'gender' and 'number' and they are not controlled by the argument-structure of a verb. It also has been proposed, in this work, that the argument-structure of some causative form 2 verbs might control four noun-phrases, i.e., an agent marked with '*ney*', an intermediate agent marked with '*sey*', an indirect object phrase marked with '*kao*' and a nominative object. This analysis assumed that the intermediate agent, similar to an agent of a passive sentence, if omitted from a sentence then it is semantically implied.

The modeling of the verbal syntax in Urdu has been presented in Chapter 8. The main features represented by a verb in many languages are tense, aspect and mood, which are represented in Urdu in its own way. These have been presented and modeled through LFG in this work. The verb agreement in Urdu has many dimensions for the dependency, due to which verbs and auxiliary verbs change their form. The tense, aspect and mood features represented by various verb morphemes and auxiliaries have been identified and phrase structure rules for the formation of sentences has been presented. It was proposed in this work that computationally a verb in Urdu might be separated into two lexical parts: (i) the root or stem of a verb, which carries the principal meaning of a verb and contains information about the transitivity and argument-structure; (ii) the inflectional morphemes and auxiliary verbs, which carry information about tense, mood and aspect. It was shown that the computational equations were simpler using this scheme. The perfect, progressive, repetitive and inceptive aspects in Urdu have been modeled under LFG. The declarative, permissive, prohibitive, imperative, capacitive and suggestive moods in Urdu have also been formulated under LFG by presenting c-structures and f-structures.

A parsing algorithm, which makes chunks with the help of morphologically closed word classes in Urdu, was proposed and implemented using a novel Ordered Context Free Grammar (OCFG). The proposed OCFG rules have additional attributes, i.e., order and type associated with each rule. The order of a rule employs linguistic features of words to make chunks with neighbor words, e.g., the case-marker make chunks with nouns to make noun phrases. The final parse is achieved after chunks of basic phrases have been made. While chunking and parsing drive parse tree (i.e., c-structure), the unification may be carried out simultaneously to make f-structures.

A roman-script has also been proposed, which is used for the transcription of Urdu sentences in this thesis. The characters of this roman-script are selected in such a way that computerized transfer of text to this roman-script from Urdu-script is possible and vice versa. It is also taken care that the mapped characters in these scripts be phonetically the same or as close as possible.

10.2 Future Directions

A standard large corpus of Urdu text in Urdu script may be developed, which may contain sentences from various constructions in Urdu. The same corpus may be made available in the roman script, using which an automatic conversion from roman script to Urdu script and vice versa may be tested. The corpus may be utilized for automatic tagging, chunking and parsing applications and for comparing and evaluating these applications for various proposals. The corpus may be utilized for automatic or semi-automatic extraction of world knowledge from the given text. The same corpus may also be made bilingual, using which various statistics-based and example-based machine translation studies may be made between Urdu and other languages. Moreover, this bilingual parallel corpus may be utilized for machine translation testing and validation studies. It can be utilized to evaluate and compare two machine translation systems.

Moreover, text corpora taken from various sources, such as, newspapers, literary work, editorial work, older Urdu books, text books, Islamic books and TV plays may be developed and may be compared for various difference, which may exist, between these corpora. Using these corpora, a more systematic computational linguistic study may be made, such as, for the usage of case markers and post-positions '*ney*' and '*kao*'.

A specialized morphological analyzer, based on finite state transducers, for Urdu text may be developed, that covers various aspects of Urdu morphology. In this thesis, only inflectional morphology has been studied. The Urdu morphological analyzer may cover the basic verb, noun and adjective forms covered in this thesis, as well as it may cover derivational morphology, such as, formation of nouns from verbs, verbs from nouns, or adjectives from nouns. The work of (Kaplan and Kay 1994) may be utilized to cover irregular morphological constructions in Urdu. The morphological analyzer may cover various other morphological conversions of nouns to verbs, like, N-V complex predicate formation. It may also cover construction of compound nouns and adjectives. The morphological analyzer may be built with such an interface that its output may be utilized with other modules, e.g., its output may be utilized by a parser or syntax analyzer.

LFG based Grammar implementation of syntax may be further improved by studying and analyzing more sentence constructions from Urdu texts, by collecting

more example data for the particular construction, e.g., by collecting more usages of case markers and post-positions in Urdu. Many other sentence constructions in Urdu that are not covered in the thesis may be studied and the rules for those may be incorporated in the syntax grammar, which may include conditional statements, correlatives, complementizers, multi-gap constructions, anaphora resolution etc.

The parsing algorithm presented based on OCFG needs further improvement. The tokenization and tagging algorithms needs enhancements. The chunking may be improved by incorporating LFG based unification information and if the unification fails, the parse may be rejected. For robust testing of the parsing algorithm, based on Urdu chunking, a standard corpus of Urdu text may be useful. Statistical chunking techniques may be implemented to validate the rule order and results based on OCFG. Alternatively, the standard parsing techniques may also be employed, like chart parsing, along with specialized Urdu rules to eliminate unwanted parse trees.

As discussed in Chapter 3, the context free grammar (CFG) may be used to model natural languages, however, it will require more part of speech (POS) categories as well as more rules. To model the same linguistics' phenomena, lexical functional grammar (LFG) based modeling requires fewer POS categories and fewer rules. However, LFG has an overhead of attribute unification. A detailed time and space complexity study may be made to compare implementation of natural language grammars using CFG, LFG or HPSG.

The ideas of Urdu computational grammar developed for machine translation may be utilized for various other Urdu NLP applications, such as, grammar checker, text summarization, question-answer systems, expert systems, text categorization, information extraction, intelligent search applications, speech processing and knowledge engineering.

Appendix A

ROMAN SCRIPT FOR URDU LANGUAGE

To use Latin characters as a script for writing languages that use other script characters is commonly referred to as 'roman script'. Various character sets of 'roman script' for representing Urdu and Hindi languages already exist, but mostly to transfer text bi-directionally in these script, using a computer, is difficult, especially without a dictionary. In this appendix, a new character set is being proposed so that the computational transfer of text is possible between Urdu script and proposed roman script.

Urdu is written in Arabic-Persian script with some additional characters, while Hindi is written in the Devanagari Script. Otherwise, Urdu and Hindi have common syntactic structure and most of the commonly used vocabulary is the same. For the proposed roman script, the mapping is also given for Hindi language, however, the transfer of text between roman script and Hindi may require a small dictionary to disambiguate some Urdu characters.

The characters are selected so that these are phonetically close to English characters so that the English reader may read the script easily, however, at some places it was not possible to reduce the ambiguity in the script, especially for vowel sounds. The following Tables give mapping of characters between Urdu, Roman and Hindi scripts.

Table A.1: Mapping of Unambiguous Consonants

Urdu	Unicode	Hindi	Unicode	IPA	Unicode	Roman	Hex	Dec
ب	0628	ब	092C	b	0062	*b*	62	98
بھ	0628	भ	092D	b^h	0062+02B0	*bh*	62+68	98+104
پ	067E	प	092A	p	0070	*p*	70	112
پھ	067E	फ	092B	p^h	0070+02B0	*ph*	70+68	112+104
ت	062A	त	0924	t	0074	*t*	74	116
تھ	062A	थ	0925	t^h	0074+02B0	*th*	74+68	116+104
ٹ	0679	ट	091F	ʈ	0288	*T*	54	84
ٹھ	0679	ठ	0920	$ʈ^h$	0288+02B0	*Th*	54+68	84+104
ج	062C	ज	091C	ʤ	02A4	*j*	6A	106

Urdu	Unicode	Hindi	Unicode	IPA	Unicode	Roman	Hex	Dec
جھ	062C	झ	091D	dʒʰ	02A4+02B0	*jh*	6A+68	106+104
چ	0686	च	091A	ʧ	02A7	*ch*	63+68	99+104
چھ	0686	छ	091B	ʧʰ	02A7+02B0	*chh*	63+68+68	99+104+68
خ	062E	ख़	0959	x	0078	*x*	4B	75
د	062F	द	0926	d	0064	*d*	64	100
دھ	062F	ध	0927	dʰ	0064+02B0	*dh*	64+68	100+104
ڈ	0688	ड	0921	ɖ	0256	*D*	44	68
ڈھ	0688	ढ	0922	ɖʰ	0256+02B0	*Dh*	44+68	68+104
ر	0631	र	0930	r	0072	*r*	72	114
ڑ	0691	ड़	095C	ɽ	027D	*R*	52	82
ڑھ	0691	ढ़	095D	ɽʰ	027D+02B0	*Rh*	52+68	82+104
س	0633	स	0938	s	0073	*s*	73	115
ش	0634	ष	0937	ʃ	0283	*sh*	9A	154
غ	063A	ग़	095A	ɣ	0263	*G*	47	71
ف	0641	फ़	095E	f	0066	*f*	66	102
ق	0642	क़	0958	q	0071	*q*	71	113
ک	06A9	क	0915	k	006B	*k*	6B	107
کھ	06A9	ख	0916	kʰ	006B+02B0	*kh*	6B+68	107+104
گ	06AF	ग	0917	g	0261	*g*	67	103
گھ	06AF	घ	0918	gʰ	0261+02B0	*gh*	67+68	103+104
ل	0644	ल	0932	l	006C	*l*	6C	108
م	0645	म	092E	m	006D	*m*	6D	109
ن	0646	न	0928	n	006E	*n*	6E	110

Table A.1: Mapping of Ambiguous Consonants

Urdu	Unicode	Hindi	Unicode	IPA	Unicode	Roman	Hex	Dec
و	0648	व	0935	ʋ	028B	*w*	77	119
ھ	06BE,0647	ह	0939	ɦ	0266	*h*	68	104
ہ	06C1	ह	0939	ɦ	0266	*h*	68	104
ء	0621, 0654	Separates 2 vowels (hyphen)				–	–	–
ی	06CC	य	092F	j	006A	*y*	79	121
ے	06D2	य	092F	j	006A	*Y*	59	89

Table A.3: Mapping of Consonants in Urdu but not in Hindi

Urdu	Unicode	Hindi	Unicode	IPA	Unicode	Roman	Hex	Dec
ث	062B	स़	0938+093C	–	–	*th* *C*	74+68 43	116+104 67

ح	062D	ह़	0939+093C	ɦ	0127	*H*	48	72
ذ	0630	ज़	095B	z	007A	*Z*	5A	90
ز	0632	ज़	095B	z	007A	*z*	7A	122
ژ	0698	ज़	095B	ʒ	0292	*zh* *X*	7A+68 58	122+104 88
ص	0635	स	0938	s	0073	*S*	53	83
ض	0636	ज़	095B	z	007A	*J*	4A	74
ط	0637	त	0924	t	0074	*tt*	74+74	116+116
ظ	0638	ज़	095B	z	007A	*zz*	7A+7A	122+122
ع	0639	–	–	–	–	*A*	41	65
ں	06BA	ँ	0901	–	–	*N*	4E	78

Table A.4: Mapping of Consonants in Hindi but not in Urdu

Urdu	Unicode	Hindi	Unicode	IPA	Unicode	Roman	Hex	Dec
–	–	श	0936	ʃ	0283	*Sh*	53+68	83+104
–	–	ञ	091E	ɲ	0272	*nn*	6E+6E	110+110
–	–	ण	0923	ɳ	0273	*N*	4E	78
–	–	ङ	0919	ŋ	014B	*ng*	6E+67	110+103
–	–	ऩ	0929	_	–	*nNn*	6E+4E+6E	110+78+110
–	–	ळ	0933	_	–	*L*	4C	76
–	–	ऴ	0934	_	–	*lLl*	6C+4C+6C	108+76+108
–	–	ऱ	0931	_	–	*rr*	72+72	114+114

Table A.5: Mapping of Vowels

Urdu	Unicode	Hindi	Unicode	IPA	Unicode	Roman	Comment	Hex	Dec
ا	0627	अ	0905	ə	0259	*a*	word initial only	61	97
زَبَر	064E	ा	093E	ə	0259	*a*	after consonant	61	97
آ	0622	आ	0906	a	0061	*aa*	word initial only	61+61	97+97
زَبَر+ا	064E+0627	_	–	a	0061	*aa*	after consonant	61+61	97+97
ا+زِیر	0627+0650	इ	0907	i	0069	*ae*	word initial only	61+65	97+101
زِیر	0650	ि	093F	i	0069	*e*	after consonant	65	101
ا+زِیر+ی	0627+0650 +06CC	ई	0908	_	–	*aee*	word initial only	61+65+65	97+101+ 101
زِیر +ی	0650+06CC	ी	0940	_	–	*ee*	after consonant	65+65	101+101
ئ	0626	ी	0940	_	–	*ee*	word final only	65+65	101+101
ا+پیش	0627+064F	उ	0909	_	–	*ao*	word initial only	61+6F	97+111

Urdu	Unicode	Hindi	Unicode	IPA	Unicode	Roman	Comment	Hex	Dec
پیش	064F	ु	0941	_	–	*o*	after consonant	6F	111
ا+ پیش+و	0627+064F +0648	ऊ	090A	_	–	*oo*	word initial only	6F+6F	111+111
پیش+و	064F+0648	ू	0942	_	–	*oo*	after consonant	6F+6F	111+111
ؤ	0624	ू	0942	_	–	*oo*	word final only	6F+6F	111+111
ر+زِیر+ی		ऋ	090B	_	–	*re*	–	72+65	114+101

Table A.6: Mapping of Diphthongs

Urdu	Unicode	Hindi	Unicode	IPA	Unicode	Roman	Hex	Dec
زِیر+ے	0650+06D2	ए	090F	_	–	*ey*	65+79	101+121
زِیر+ے	0650+06D2	े	0947	_	–	*ey*	65+79	101+121
زَبَر+ے	064E+06D2	ऐ	0910	_	–	*ay*	61+79	97+121
زَبَر+ے	064E+06D2	ै	0948	_	–	*ay*	61+79	97+121
زَبَر+و	064E+0648	ओ	0913	_	–	*ao*	61+6F	97+111
زَبَر+و	064E+0648	ो	094B	_	–	*ao*	61+6F	97+111
ا+زَبَر+و	0627+064E +0648	औ	0914	_	–	*ao*	61+6F	97+111
ا+زَبَر+و	0627+064E +0648	ौ	094C	_	–	*ao*	61+6F	97+111
دوزِیر	064D	–	–	_	–	–	_	–
دوزَبَر	064B	–	–	_	–	–	_	–
دوپیش	064C	–	–	_	–	–	_	–

Appendix B

ALGORITHMS FOR WORD REPRESENTATION

In this Appendix, algorithms related to word insertion into a trie and the minimization of DFA (Mihov; Daciuk 1998; Ciura and Deorowicz 2001), which are used and implemented in Chapter 6 are given as follows.

B.1 Algorithm for Word Insertion into a Trie

Algorithm for inserting a sequence of word strings into a tire is:

1: FOR each word do the following steps
2: SET currentNode = RootNode
3: FOR each character in a word
4: IF there exist a branch matching current character
5: *currentNode = nextNode* (matching character)
6: ELSE Add new Node for the current character

B.2 Algorithm for Word Insertion into an Acyclic DFSA

Algorithm for inserting a sequence of word strings into an acyclic deterministic finite state automata is:

1: Sort the given list of words alphabetically
2: WHILE there is a word DO
3: Find Prefix of word already in Automaton (Algorithm B.2.1)
4: Add Rest of the word to Automaton (Algorithm B.2.2)
5: END WHILE

Algorithm B.2.1: Algorithms to find "prefix of a word already in the automaton", is as follows:

1: Start from start state
2: FOR each character in the word
3: IF there exist a transition matching character
4: Add character to prefix
5: Move to next State
6: ELSE go to step 7
7: The character consumed so far are the prefix already in the automaton. Stop.

Algorithm B.2.2: Algorithm to "add rest of the word to automaton" is:

1: SET *currentState* = last state in step 5 of Prefix algorithm
2: FOR each character in the rest of the word
3: IF NOT last character, do steps 4, 5
4: Add transition for character from *currentState* to *newState*
5: SET currentState = newState
6: ELSE add transition for character from *currentState* to *finalState*

B.2 Algorithm for Word Insertion into an Acyclic Minimal DFSA

Algorithm for inserting a sequence of word strings into an acyclic minimal deterministic finite state automata is:

1: Sort the given list of words alphabetically
2: WHILE there is a word DO steps 3-5
3: Find Prefix of word already in Automaton (Algorithm B.2.1)
4: Add Rest of the word to Automaton (Algorithm B.2.2)
5: Minimize (Algorithm B.3.1)

For minimal acyclic DFSA, there could be more than one final state. Therefore states are divided into two classes – (a) the terminal final state (TFS), having no out-going transition and there is only one TFS in an automaton (b) the intermediate final state (IFS) that can have out-going transitions as well as in-coming transitions and there can be many IFS in an automaton. Each state is stored with a flag to tell whether it is a final state or not. The algorithm to find "prefix of the word" needs slight modification for such final states and the algorithm to Check for minimization is as follows.

Algorithm B.3.1:

1: Maintain a list of recently traversed states for the current word while "*finding prefix*" and "*adding rest of the word*". The length of the list is equal to one greater than length of word.
2: FOR each state in the list starting from last to first
3: IF there is an equivalent state in the automaton
4: Replace transitions that are coming to current state from current state to the state already in the automaton and delete the current state.
5: ELSE add the current state to automaton.

Appendix C

SAMPLE SENTENCES FOR PARSING

In this Appendix, the sentences used for the parsing by chunking algorithm presented in Chapter 10 have been listed.

C.1 Basic Sentences

This section presents chunking of basic sentences. In the following list, each sentence on left side is presented with its transliteration and on the right side corresponding tokens and chunking is described.

1.	یہ شاہد کی بیوی ہے *yeh shaahed kee beewee hay*	[PNS] [N PM N] AUX → NP NP AUX
2.	وہ موٹی ہے *woh maoTee hay*	PNS N AUX → NP NP AUX
3.	اب وہ بہت خوش ہے *aab woh bohat xaosh hay*	Adv PNS [Adj NP] AUX → AJP NP NP AUX
4.	آج شب برات ہے *aaj shab-baraat hay*	N N AUX → NP NP AUX
5.	وہ اپنی بہن کے گھر جا رہی ہے *woh aapnee behan key ghar jaa rahee hay*	PNS [[PNP N] PM N] [VB VM] → NP NP V_2
6.	وہ بس کا انتظار کر رہی ہے *woh bas kaa aentezzaar kar rahee hay*	PNS [N PM N] [VB VM] → NP NP V_2
7.	اس کی بیوی بس میں ہے *aes kee beewee bas meyN hay*	[PNS PM N] [N PP] AUX → NP PPP AUX
8.	اس کی بہن کا گھر سمن آباد میں ہے *aes kee behan kaa ghar samanaabaad meyN hay*	[[PNS PM N] PM N] [N PP] AUX → NP PPP AUX
9.	بڑی بس سمن آباد جا رہی ہے *yeh bas samanaabaad jaa rahee hay*	[Adj N] N [VB VM] → NP NP NP V_2
10.	بس رک رہی ہے *bas rok rahee hay*	N [VB VM] → NP V_2
11.	پیلی ٹیکسی آ رہی ہے *Teyksee aa rahee hay*	[Adj N] [VB VM] → NP V_2
12.	شاہد کا بیٹا سڑک کے کنارے کھڑا ہے *shaahed kaa beyTaa saRak key kenaarey khaRaa hay*	[N PM N] [N PM N] [VB VM] → NP NP V_2
13.	شاہد کا بیٹا اپنا ہاتھ ہلا رہا ہے *shaahed kaa beyTaa aapnaa haath helaa rahaa hay*	[N PM N] [PNP N] [VB VM] → NP NP V_2
14.	شاہد کا بیٹا ٹیکسی میں بیٹھ رہا ہے *shaahed kaa beyTaa Teyksee meyN bayTh rahaa hay*	[N PM N] [N PP] [VB VM] → NP PPP V_2
15.	ٹیکسی سمن آباد جا رہی ہے *Teyksee samanaabaad jaa rahee hay*	N N [VB VM] → NP NP V_2

16.	اب وہ اس کی ٹیکسی میں بیٹھ رہی ہے *aab woh aes kee Teyksee meyN bayTh rahee hay*	Adv PNS [PNS PM N] PP [VB VM] → Adv NP [NP PP] V_2 → AJP NP PPP V_2
17.	شاہد کا بیٹا اپنے ماموں کے گھر پہنچ گیا ہے *shaahed kaa beyTaa aapney maamooN key ghar pohanch gayaa hay*	[N PM N] [PNP N] PM N [VB VM] → NP [NP PM NP] V_2 → NP NP V_2
18.	اس تصویر کو دیکھو *aes taSweer kao dekhao*	PNS [N CM_kao] [VB VM] → NP NP V_2
19.	آدمی پہیہ مرمت کر رہا ہے *aadmee paheeyaa maramat kar rahaa hay*	N N [VB VM] → NP NP V_2
20.	لڑکا پمپ کے قریب ہے *laRkaa pamp key qareeb hay*	N [N PM N] AUX → NP NP AUX
21.	بوڑھا آدمی درخت کے سائے میں بیٹھا ہے *booRhaa aadmee daraxt key saaey meyN bayThaa hay*	[Adj N] [N PM N] PP [VB VM] → NP [NP PP] V_2 → NP PPP V_2
22.	ہمارا سکول گاوں سے باہر ہے *hamaaraa sakool gaaooN sey baahar hay*	[PNP N] [N PP] AUX → NP PPP AUX
23.	سکول کے پانچ کمرے ہیں *sakool key paanch kamrey hayN*	N PM [NC N] AUX → [NP PM NP] AUX → NP AUX
24.	روشنی اور تازہ ہوا کے لیے کمروں میں کھڑکیاں ہیں *raoshnee aor taazah hawaakey leeey kamraoN meyN kheRkeeyaaN hayN*	N CJC [Adj N] PP N [PP N] AUX → [[NP CJC NP] PP] PPP NP AUX → PPP PPP NP AUX
25.	کمروں میں ٹاٹ بچھے ہیں *kamraoN meyN TaaT bechhey hayN*	[N PP] N [VB VM] → PPP NP V_2
26.	دیواروں پر چارٹ لگے ہیں *deewaaraoN par chaarT lagey hayN*	[N PP] N [VB VM] → PPP NP V_2
27.	کمرے میں تختہ سیاہ میز اور کرسی ہے *kamrey meyN taxtah-seeyah meyz aor korsee hay*	[N PP] [N N CJC N] AUX → PPP NP AUX
28.	ہمارے سکول میں ایک کھلا میدان ہے *hamaarey sakool meyN aeyk kholaa meydaan hay*	[[PNP N] PP] [NC Adj N] AUX → PPP NP AUX
29.	سکول کے صحن میں پھولوں کے پودے ہیں *sakool key SeHan meyN phoolaoN key paodey hayN*	[[N PM N] PP] [N PM N] AUX → PPP NP AUX
30.	ہم پودوں کی حفاظت کرتے ہیں *ham paodooN kee Hefaazzat kartey hayN*	PNS [N PM N] [VB VM] → NP NP V_2
31.	ہم بڑے شوق سے سکول جاتے ہیں *ham baRey shaoq sey sakool jaatey hayN*	PNS [[Adj N] PP] N [VB VM] → NP NP PPP V_2
32.	استاد ہمیں پیار اور محنت سے پڑھاتے ہیں *aostaad hameyN peyaar aor meHnat sey paRhaatey hayN*	N PNS [[N CJC N] PP] [VB VM] → NP NP PPP V_2
33.	ہم اپنے استادوں کا ادب کرتے ہیں *ham aapney aostaadooN kaa aadab kartey hayN*	N [[PNP N] PM N] [VB VM] → NP NP V_2
34.	میں کتاب خریدتی ہوں *meyN ketaab xareedtee hooN*	N N [VB VM] → NP NP V_2
35.	میں نے کتاب خریدی ھے *meyN ney ketaab xareedee hay*	[N CM_erg] N [VB VM] → NP NP V_2
36.	میں کتاب خرید رہا تھا *meyN ketaab xareed rahaa thaa*	N N [VB VM] → NP NP V_2
37.	حامد کتاب خرید چکا تھا *Haamed ketaab xareed chokaa thaa*	N N [VB VM] → NP NP V_2
38.	انجم نے صدف کو مصالحہ چکھایا *anjom ney Saddaf kao meSaalHah chakhaayaa*	[N CM_erg] [N CM_kao] N [VB VM] → NP NP NP V_2
39.	وہ کتاب پڑھنے والا ہے *woh ketaab paRhney waalaa hay*	N N V_{INF} AIA AUX → NP NP V_{INF} AIA AUX

C.2 News Website Sentences

This section presents chunking of news website sentences. These sentences are relatively complex and require manual tokenization before performing chunking. In the following list, a few sentences on left side are presented with the transliteration and on the right side chunking is described.

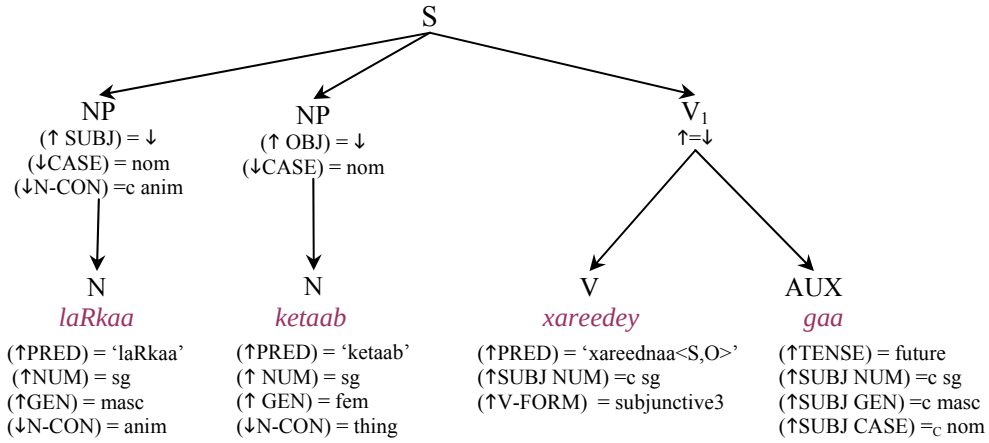
شمالی کوریا نے جمعہ کو ایٹمی بحران کے حل کیلئے امریکا کو غیرمشروط مذاکرات کی پیشکش کی ہے اور کہا ہے کہ اگر امریکا بحران کا حل چاہتا ہے تو فوری مذاکرات کرے *saomaalee kaoryaa ney jommaAah kao aeTmee boHraan key Hal keyleeey aamreekaa kao Gayr mashroott moZaakraat key peyshkash kee hay aaor kahaa hay keh aagar aamreekaa boHraan kaa Hal chaahtaa hay tao faoree moZaakraat karey*	S CJS CJR1 S CJR2 S
ملک کی معاشی صورت حال کا ذکر کرتے ہوئے صدرمشرف نے کہا کہ پاکستان کشکول لے کر نہیں گھوم رہا اوراقتصادی طورپر ابھرتا ہوا ملک ہے *molk kee maAaashee Saorat e Haal kaa Zekar kartey hooey Sadar mosharaf ney kahaa keh paakestaan kashkaol ley kar naheeN ghoom rahaa aaor aeqteSaadee ttaor par aobhartaa hooaa molk hay*	S CJS S CJC S
صدر نے کہا کہ وہ حکومت سے کہتے ہیں کہ مہنگائی کنٹرول کرو اور قیمتوں میں کمی لاؤ *Sadar ney kahaa key woh Hakoomat sey kehtey hayN keh mehngaaee kanTraol karao aaor qeematooN meyN kamee laaoo*	S CJS S CJS S CJC S
منگل کو پاکستان ٹیم کو اس وقت شدید دھچکہ لگا جب آل راؤنڈر عبدالرزاق گھٹنے کی انجری کی وجہ سے ورلڈ کپ سے باہر ہوگئے *mangal kao paakestaan Teem kao aos waqt shadeed dhachkah lagaa jab aal raaoonDar Aabdaolrazaaq ghooTney kee aenjree kee wajah sey warlD kap sey baahar hao gaaey*	S CJS S
ڈاکٹروں نے فریکچر کی تشخیص کی اور عبدالرزاق کو تین ہفتے آرام کا مشورہ دیا ہے جبکہ انہیں فزیو تھراپی کیلئے مزید تین ہفتے درکار ہوں گے *DaakTarooN ney farekchar kee tashxeeS kee aaor Aabdaolrazaaq kao teen haftey aaraam kaa mashwarah deeyaa hay jabkeh aenheeN fezeeao tharaapee keeleeey mazeed teen haftey darkaar haoN gey*	S CJC S CJS S
ایک سوال کے جواب میں انضمام نے کہا کہ ان کو اوپننگ کرنے کا مشورہ دینے والے فضول بات کر رہے ہیں البتہ وہ بیٹنگ آرڈر میں اوپر کھیلنے کی کوشش ضرور کریں گے *aeyk sawaal key jawaab meyN aenJamaamolHaq ney kahaa keh aen kao aopanneng karney kaa mashwarah deyney waaley faJool baat kar rahey hayN aalbatah woh beyTeng aarDar meyN aopar khelney kee kaoshesh Jaroor kareeN gey*	S CJS S CJS S
انضمام الحق نے کہا کہ میچ میں جیت بولر دلواتے ہیں لیکن سارے گیارہ کھلاڑیوں کو اچھی کارکردگی کا مظاہرہ کرنا ہوگا *aenJamaamolHaq ney kahaa keh meych meyN jeet baolar delwaatey hayN leyken saarey gayaarah khelaaReeyooN kao aachchee kaarkardagee kaa moJaaharah karnaa hao gaa*	S CJS S CJS S

Appendix D

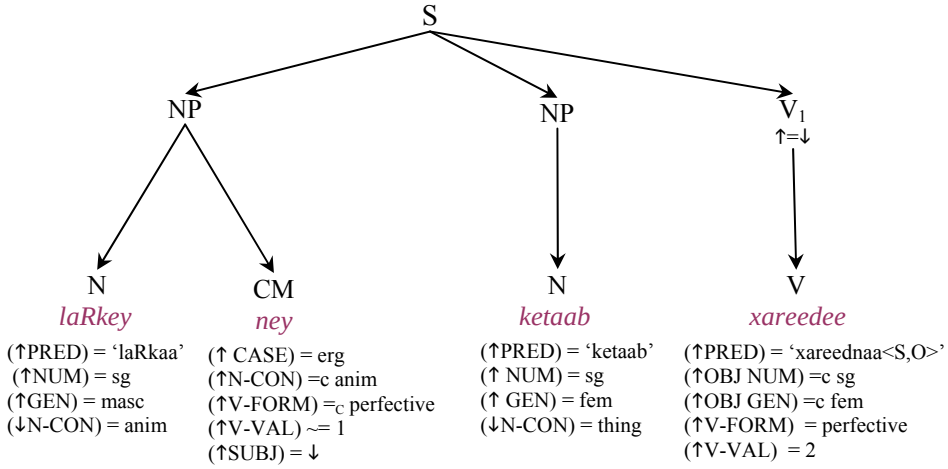
CONSTITUENT STRUCTURES

In this Appendix, constitutent structures (c-structures) for corresponding feature-structures (f-structures) shown in Chapter 7 are given to elaborate the creation of f-structures.

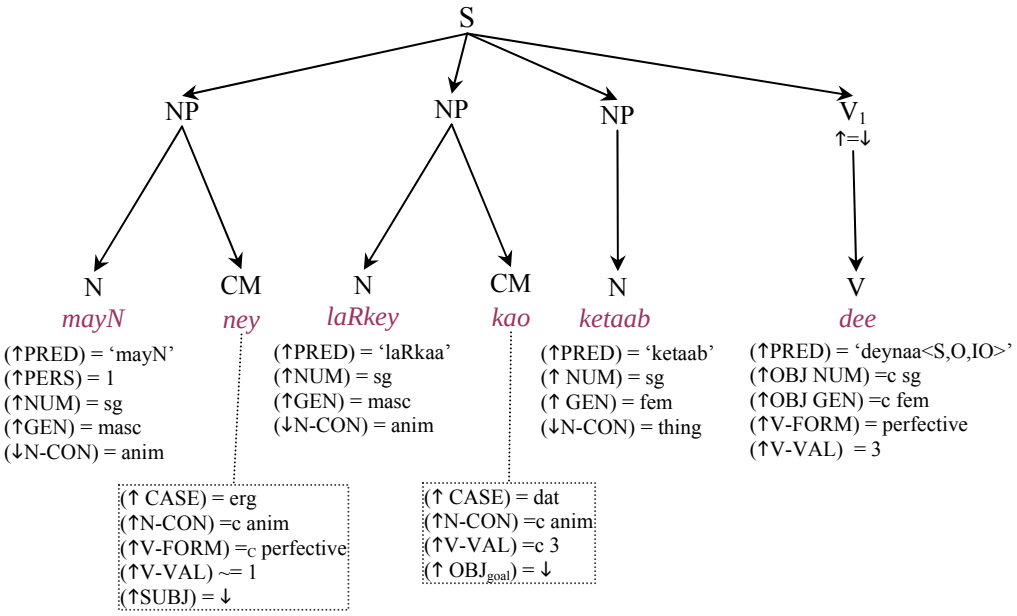
D.1 C-Structure for the F-Structure in Figure 7.5



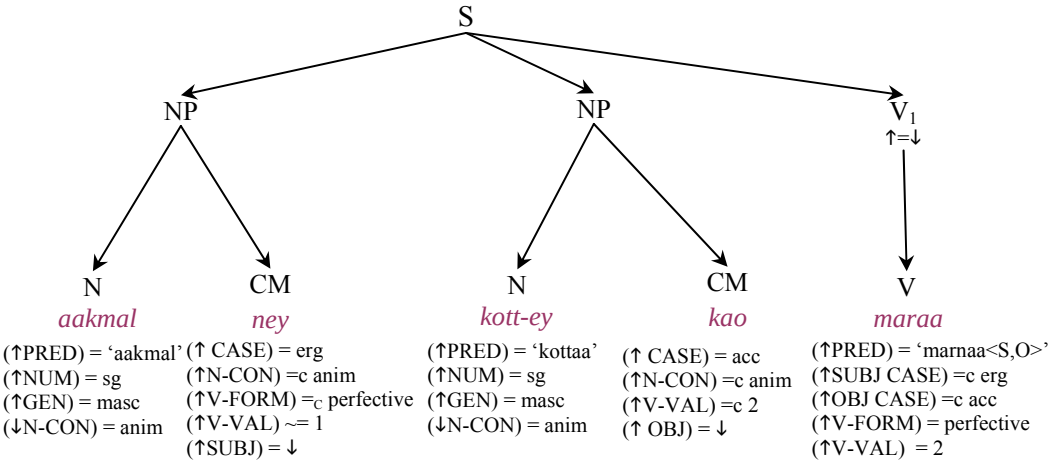
D.2 C-Structure for the F-Structure in Figure 7.6



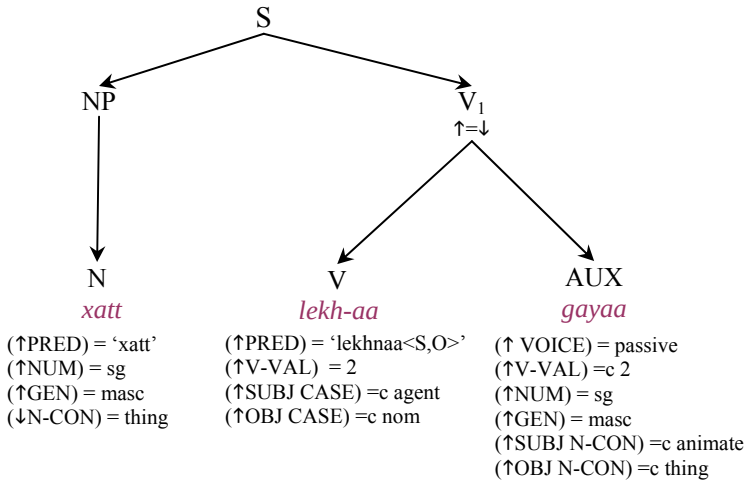
D.3 C-Structure for the F-Structure in Figure 7.7



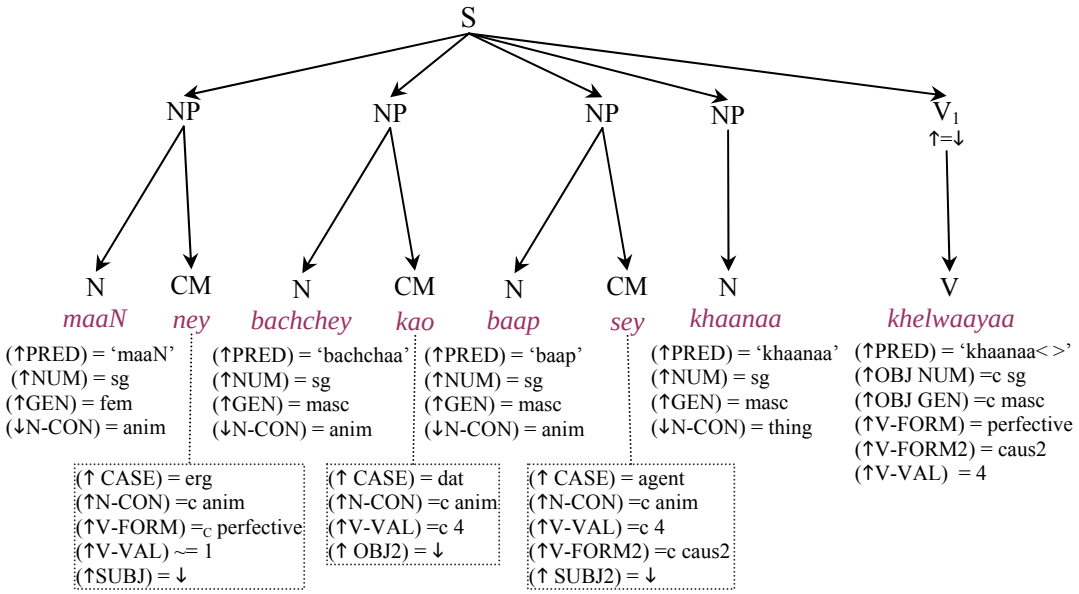
D.4 C-Structure for the F-Structure in Figure 7.8



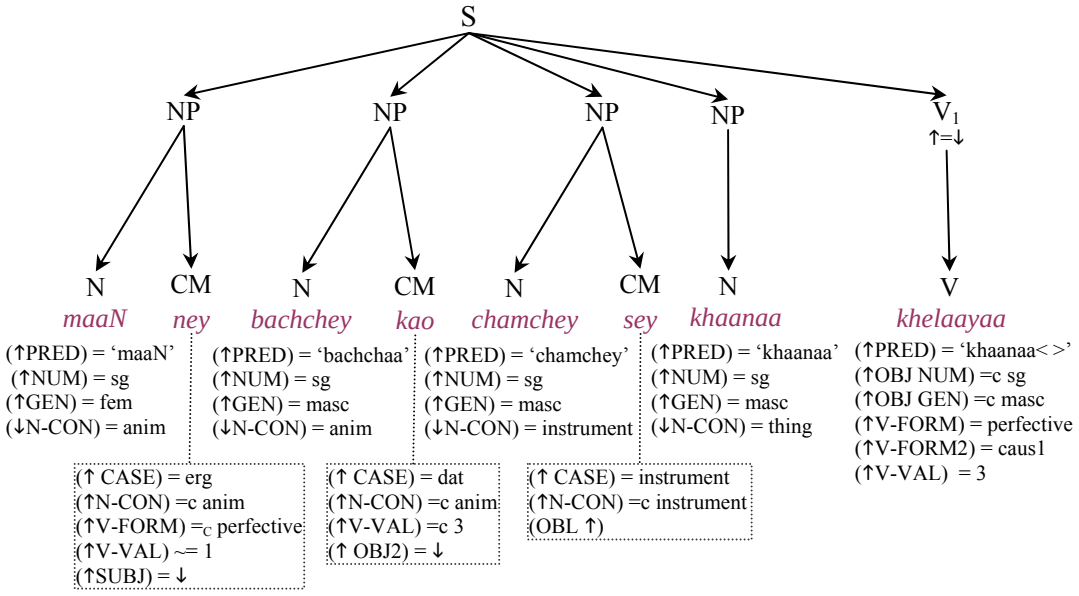
D.5 C-Structure for the F-Structure in Figure 7.9



D.6 C-Structure for the F-Structure in Figure 7.18



D.7 C-Structure for the F-Structure in Figure 7.19



Appendix E

URDU GRAMMAR IMPLEMENTATION

The concepts of Urdu grammar proposed in this research work are implemented using Xerox linguistic tools. The Xerox Finite State Tool (XFST) and Finite State Lexicon Compiler (LEXC) are used to implement Urdu morphology. The Xerox Linguistic Environment (XLE) is used for syntactical analysis based on Lexical Functional Grammar (LFG). The XLE does tokenization and morphological analysis of the given sentences using output of LEXC, and then used syntax rules to parse sentences into c-structures and f-structures. In this appendix, the morphological and syntactical rules for Urdu grammar are listed in Xerox linguistic tools format.

E.1 Morphology Implementation

The finite state Lexicon Compiler (LEXC) compiles input source file into a lexical transducer using command '*compile-source filenam*e'. After the finite state transducer is successfully compiled, it may be saved using command '*save-source filename*', then various commands may be used to analyse the transducer. XFST also takes input of LEXC and may be used to analyse and extend the automata for irregular forms. The following is the source file that may be input to LEXC.

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Urdu Morphology

Multichar_Symbols
+Verb
+Root +Caus1 +Caus2 +Repeat +Perf +Inf +Subj +Impr  ! Verb Forms
+Rude +Formal +Polite +Request ! 2nd Person Honorific Forms

+Noun
+Common +Proper ! Noun class
+Abstract +Group +Spatial +Temporal
+Instrument +Animate +Thing ! Noun Concept
+Mass +Count ! Noun Type
+Nominative +Oblique +Vocative ! Noun Form
+Arabic +Persian +Hindi +Turkish +English ! Base Language

+Adjective

+Aux
+Future +Present +Past ! Tense
+Perfect +Progress +Cont +Incept ! Aspect
+Comp +Decl  ! Mood
```

```
! Common Symbols
+Fem +Masc ! Gender
+Sg +Pl ! Number
+1st +2nd +3rd ! Person

LEXICON Root
Verbs;
Nouns;
Adjectives;
Aux_Verb;

! verb root forms
! from verb root form we go to stem form
! VerbRoot1 = verb roots that can generate Caus1 but not Caus2 Form
! VerbRoot2 = verb roots that can generate Caus1 and Caus2 Form
! VerbRoot3 = verb roots that can generate Caus2 but not Caus1 Form
! VerbRoot4 = verb roots that cannot generate Causative Forms

LEXICON Verbs

! VerbRoot1  = verb roots that can generate Caus1 but not Caus2 Form
dokhnaa+Verb:dokh                VerbRoot1;     ! pain
behnaa+Verb:beh                  VerbRoot1;     ! flow

! VerbRoot2  = verb roots that can generate Caus1 and Caus2 Form
hansnaa+Verb:hans                VerbRoot2;     ! laugh
paRhnaa+Verb:paRh                VerbRoot2;     ! read
maangnaa+Verb:maang              VerbRoot2;     ! ask for
lekhnaa+Verb:lekh                VerbRoot2;     ! write
kaatnaa+Verb:kaat                VerbRoot2;     ! cut
pohanchnaa+Verb:pohanch          VerbRoot2;     ! reach
darnaa+Verb:dar                  VerbRoot2;     ! fear
chalnaa+Verb:chal                VerbRoot2;     ! walk
deykhnaa+Verb:deykh              VerbRoot2;     ! look
chakhnaa+Verb:chakh              VerbRoot2;     ! taste
lagnaa+Verb:lag                  VerbRoot2;     ! touch

! VerbRoot3  = verb roots that can generate Caus2 but not Caus1 Form
kholnaa+Verb:khol                VerbRoot3;     ! open
bolnaa+Verb:bol                  VerbRoot3;     ! speak
nekalnaa+Verb:nekal              VerbRoot3;     ! come out
xareednaa+Verb:xareed            VerbRoot3;     ! buy
poochhnaa+Verb:poochh            VerbRoot3;     ! ask about, ask

! VerbRoot4  = verb roots that do not have causative forms

bataanaa+Verb:bataa              VerbRoot4;     ! tell
deynaa+Verb:d                    VerbRoot4a;    ! give
leynaa+Verb:l                    VerbRoot4a;    ! take
deynaa+Verb:dee                  VerbRoot4b;    ! give
leynaa+Verb:lee                  VerbRoot4b;    ! take

! Irregular and Other Verb Roots that are not covered above
chalaanaa+Verb:chalaa          VerbStem; ! drive
banaanaa+Verb:banaa            VerbStem; ! make
nahaanaa+Verb:nahaa            VerbStem; ! bathe
aanaa+Verb:aa                  VerbStem; ! come
bolaanaa+Verb:bolaa            VerbStem; ! call (to come)
bolwaanaa+Verb+Caus2:bolwaa    VerbStem; ! make someone call someone
rehnaa+Verb:reh                VerbStem; ! stay
kehnaa+Verb:keh                VerbStem; ! say
```

```
cahnaa+Verb:cah                    VerbStem; ! want
gernaa+Verb:ger                    VerbStem; ! fell
khaansnaa+Verb:khaans              VerbStem; ! cough

saonaa+Verb:sao                    VerbStem1; ! sleep
raonaa+Verb:rao                    VerbStem1; ! weep

khaanaa+Verb:khaa                  VerbStem1; ! eat
khelaanaa+Verb+Caus1:khelaa        VerbStem1; ! eat causative form 1
khelwaanaa+Verb+Caus2:khelwaa      VerbStem1; ! eat causative form 2

seenaa+Verb:see                      VerbStem2;  ! sew

karnaa+Verb:kee                       VerbStem3a; ! do
karnaa+Verb:kar                       VerbStem3b; ! do
safar=karnaa+Verb:safar=kee           VerbStem3a; ! travel
safar=karnaa+Verb:safar=kar           VerbStem3b; ! travel
aenteZaar=karnaa+Verb:aenteZaar=kee   VerbStem3a; ! wait
aenteZaar=karnaa+Verb:aenteZaar=kar   VerbStem3b; ! wait


jaa+Verb+Perf:ga                      GendNumb3; ! go
jaanaa+Verb:jaa                       VerbStem5; ! go

LEXICON VerbRoot1  !
+Caus1:aa              VerbStem1;
0:0                    VerbStem1;

LEXICON VerbRoot2 !
+Caus1:aa              VerbStem1;
+Caus2:waa             VerbStem1;
0:0                    VerbStem;

LEXICON VerbRoot3 !
+Caus2:waa             VerbStem1;
0:0                    VerbStem;

LEXICON VerbRoot4 !
0:0                    VerbStem;

LEXICON VerbStem
+Root:0                          #;
+Inf:n                 Infinitive;
+Repeat:t              Repetitive;
+Perf:0                Perfective;
+Subj:0                Subjunctive;
+Impr:0                Imperative;

LEXICON Infinitive
0:0                    GendNumb1;

LEXICON Repetitive
0:0                    GendNumb2;

LEXICON Perfective
0:0                    GendNumb2;

LEXICON Subjunctive
+1st+Sg:ooN          #;
+1st+Pl:eyN          #;
+2nd+Rude:0          #;
```

```
+2nd+Formal:ao       #;
+2nd+Polite:eyN      #;
+2nd+Request:eeey    #;
+3rd+Sg:ey           #;
+3rd+Pl:eyN          #;

LEXICON Imperative
+2nd+Rude:0          #;
+2nd+Formal:ao       #;
+2nd+Polite:eyN      #;
+2nd+Request:eeey    #;

LEXICON VerbStem1
+Root:0                      #;
+Inf:n                  Infinitive;
+Repeat:t               Repetitive;
+Perf:0                 Perfective2;
+Subj:0                 Subjunctive;
+Impr:0                 Imperative;

LEXICON Perfective2
0:0                   GendNumb3;

LEXICON VerbStem2
+Root:0                      #;
+Inf:n                  Infinitive;
+Repeat:t               Repetitive;
+Perf:0                 Perfective3;
+Subj:0                 Subjunctive2;
+Impr:0                 Imperative2;

LEXICON Perfective3
0:0                   GendNumb4;

LEXICON Subjunctive2
+1st+Sg:ooN          #;
+1st+Pl:eyN          #;
+2nd+Rude:0          #;
+2nd+Formal:ao       #;
+2nd+Polite:eyN      #;
+2nd+Request:jeeey   #;
+3rd+Sg:ey           #;
+3rd+Pl:eyN          #;

LEXICON Imperative2
+2nd+Rude:0          #;
+2nd+Formal:ao       #;
+2nd+Polite:eyN      #;
+2nd+Request:jeeey   #;

LEXICON GendNumb1
+Masc:aa           #;
+Fem:ee            #;
+Obl:ey            #;

LEXICON GendNumb2
+Sg+Masc:aa        #;
+Sg+Fem:ee         #;
+Pl+Masc:ey        #;
+Pl+Fem:eeN        #;
```

```
LEXICON GendNumb3
+Sg+Masc:yaa       #;
+Sg+Fem:ee         #;
+Pl+Masc:ey        #;
+Pl+Fem:eeN        #;

LEXICON GendNumb4
+Sg+Masc:aa        #;
+Sg+Fem:0          #;
+Pl+Masc:ey        #;
+Pl+Fem:N          #;

LEXICON VerbStem3a
+Perf:0                 Perfective3;
+Subj+2nd+Request:jeeey     #;
+Impr+2nd+Request:jeeey     #;

LEXICON VerbStem3b
+Root:0                     #;
+Inf:n                  Infinitive;
+Repeat:t               Repetitive;
+Subj+1st+Sg:ooN            #;
+Subj+3rd+Sg:ey             #;
+Subj+1st+Pl:eyN            #;
+Subj+3rd+Pl:eyN            #;
+Subj+2nd+Rude:0            #;
+Subj+2nd+Formal:ao         #;
+Subj+2nd+Polite:eyN        #;
+Impr+2nd+Rude:0            #;
+Impr+2nd+Formal:ao         #;
+Impr+2nd+Polite:eyN        #;

LEXICON VerbRoot4a
+Perf:0               GendNumb2a;
+Subj+1st+Sg:ooN            #;
+Subj+3rd+Sg:ey             #;
+Subj+1st+Pl:eyN            #;
+Subj+3rd+Pl:eyN            #;
+Subj+2nd+Formal:ao         #;
+Subj+2nd+Polite:eyN        #;
+Subj+2nd+Rude:ey           #;
+Impr+2nd+Formal:ao         #;
+Impr+2nd+Polite:eyN        #;
+Impr+2nd+Rude:ey           #;

LEXICON VerbRoot4b
+Inf:n                  GendNumb1;
+Repeat:t               GendNumb2;
+Subj+2nd+Request:jeeey  #;
+Impr+2nd+Request:jeeey  #;

LEXICON VerbStem5
+Root:0                     #;
+Inf:n                  GendNumb1;
+Repeat:t               GendNumb2;
+Subj+1st+Sg:ooN            #;
+Subj+3rd+Sg:ey             #;
+Subj+1st+Pl:eyN            #;
+Subj+3rd+Pl:eyN            #;
+Subj+2nd+Rude:0            #;
+Subj+2nd+Formal:ao         #;
```

```
+Subj+2nd+Polite:eyN          #;
+Subj+2nd+Request:eeey        #;
+Impr+2nd+Rude:0              #;
+Impr+2nd+Formal:ao           #;
+Impr+2nd+Polite:eyN          #;
+Impr+2nd+Request:eeey        #;

LEXICON GendNumb2a
+Sg+Masc:eeaa      #;
+Sg+Fem:ee         #;
+Pl+Masc:eeey      #;
+Pl+Fem:eeN        #;

LEXICON Nouns
! CAT 1a: animate nouns that end in suffix -aa that generetates to
fem
laRkaa+Noun+Animate:laRk             N_Cat1a;  ! boy
daadaa+Noun+Animate:daad             N_Cat1a;  ! grand father
kotaa+Noun+Animate:kot               N_Cat1a;  ! dog
bakraa+Noun+Animate:bakr             N_Cat1a;  ! (masc.) goat

! CAT 1b: animate nouns that end in suffix -ah that generates to fem
bachchah+Noun+Animate:bachch         N_Cat1b;  ! child

! CAT 2: inanimate masc nouns that do not end with a suffix
xatt+Noun+Thing+Masc+Count:xatt      N_Cat2;   ! letter
jahaaz+Noun+Thing+Masc+Count:jahaaz  N_Cat2;   ! plane
den+Noun+Temporal+Masc:den           N_Cat2;   ! day
sawaal+Noun+Abstract+Masc:sawaal     N_Cat2;   ! question
saRak+Noun+Saptial+Masc:saRak        N_Cat2;   ! road
teyl+Noun+Thing+Masc+Mass:teyl       N_Cat2;   ! oil
seyb+Noun+Thing+Sg+Masc+Count:seyb   N_Cat2;   ! apple

! CAT 3: inanimate fem nouns that do not end with a suffix
ketaab+Noun+Thing+Fem+Count:ketaab   N_Cat3;  ! book
pencel+Noun+Instrument+Fem:pencel    N_Cat3;  ! pencil
baat+Noun+Abstract+Fem:baat          N_Cat3;  ! talk
madad+Noun+Abstract+Fem:madad            #;  ! help
SobaH+Noun+Temporal+Fem:SobaH        N_Cat3;  ! morning
moddat+Noun+Temporal+Fem:moddat      N_Cat3;  ! long (duration)

! CAT 4a: inanimate masc nouns that end in suffix -aa
khaanaa+Noun+Thing+Masc+Mass:khaan   N_Cat4a;  ! food
taalaa+Noun+Thing+Masc+Count:taal    N_Cat4a;  ! lock
chhoraa+Noun+Instrument+Fem:chhor    N_Cat4a;  ! knife (bigger)

! CAT 4b: inanimate masc nouns that end in suffix -ah
darwaazah+Noun+Spatial+Masc+Count:darwaaz  N_Cat4b; ! door
waAdah+Noun+Abstract+Masc:waAdah           N_Cat4b;  ! promise
kamrah+Noun+Spatial+Masc+Count:kamr        N_Cat4b;  ! room
penjrah+Noun+Instrument+Masc+Count:penjr   N_Cat4b;  !
chamchah+Noun+Instrument+Masc+Count:chamch N_Cat4b; ! spoon
maSaalHah+Noun+Thing+Masc+Mass:maSaalH     N_Cat4b; ! spice,
seasoning

! CAT 4b: animate masc nouns that end in suffix -ah and don't be fem
parendah+Noun+Animate:parend             N_Cat4b;  ! bird

! CAT 5: inanimate fem nouns that end in suffix -ee
cheThee+Noun+Thing+Fem+Count:cheTh       N_Cat5;  ! note
seeRhee+Noun+Spatial+Fem+Count:seeRh     N_Cat5;  ! stairs
```

```
chhoree+Noun+Instrument+Fem:chhor        N_Cat5;  ! knife (smaller)

! Nouns: Proper Names
Akmal+Noun+Proper+Animate+Masc+Sg+3rd:aakmal      #;
Hamid+Noun+Proper+Animate+Masc+Sg+3rd:haamed      #;
Hameed+Noun+Proper+Animate+Masc+Sg+3rd:hameed     #;
Zafar+Noun+Proper+Animate+Masc+Sg+3rd:zzafar      #;
Mozafar+Noun+Proper+Animate+Masc+Sg+3rd:mozzafar  #;
America+Noun+Proper+Masc:amreekah                 #;
Anjum+Noun+Proper+Animate+Fem+Sg+3rd:aanjom       #;
Sadaf+Noun+Proper+Animate+Fem+Sg+3rd:Sadaf        #;


!Nouns : animate nouns that do not end in suffix
maaN+Noun+Animate+Fem+Sg:maaN      #;  ! mother . add pl+oblique
baap+Noun+Animate+Masc+Sg:baap     #;  ! father . add pl+oblique


LEXICON N_Cat1a  ! Noun Category 1-a
+Sg+Masc:aa               #;
+Sg+Masc+Oblique:ey       #;
+Pl+Masc:ey               #;
+Pl+Masc+Oblique:ooN      #;
+Pl+Masc+Vocative:ao      #;
+Sg+Fem:ee                #;
+Pl+Fem:aN                #;
+Pl+Fem+Oblique:oN        #;
+Pl+Fem+Vocative:eeao     #;

LEXICON N_Cat1b  ! Noun Category 1-b
+Sg+Masc:ah               #;
+Sg+Masc+Oblique:ey       #;
+Pl+Masc:ey               #;
+Pl+Masc+Oblique:ooN      #;
+Pl+Masc+Vocative:ao      #;
+Sg+Fem:ee                #;
+Pl+Fem:aN                #;
+Pl+Fem+Oblique:oN        #;
+Pl+Fem+Vocative:eeao     #;

LEXICON N_Cat2   ! Noun Category 2
+Sg+Masc:0                #;
+Sg+Masc+Oblique:0        #;
+Pl+Masc:0                #;
+Pl+Masc+Oblique:ooN      #;

LEXICON N_Cat3   ! Noun Category 3
+Sg+Fem:0                 #;
+Sg+Fem+Oblique:0         #;
+Pl+Fem:eyN               #;
+Pl+Fem+Oblique:ooN       #;

LEXICON N_Cat4a   ! Noun Category 4-a
+Sg+Masc:aa               #;
+Sg+Masc+Oblique:ey       #;
+Pl+Masc:ey               #;
+Pl+Masc+Oblique:ooN      #;

LEXICON N_Cat4b   ! Noun Category 4-b
+Sg+Masc:ah               #;
+Sg+Masc+Oblique:ey       #;
```

```
+Pl+Masc:ey               #;
+Pl+Masc+Oblique:ooN      #;

LEXICON N_Cat5   ! Noun Category 5
+Sg+Fem:ee               #;
+Sg+Fem+Oblique:ee       #;
+Pl+Fem:eeaN             #;
+Pl+Fem+Oblique:eeooN    #;

LEXICON Adjectives
! CAT 1-a: Adjectives that end in a suffix -aa
achchaa+Adjective:achch       Adj_Cat1a;   !  good
neelaa+Adjective:neel         Adj_Cat1a;   !  blue
haraa+Adjective:har           Adj_Cat1a;   !  green
teesraa+Adjective:teesr       Adj_Cat1a;   !  third
kaRwaa+Adjective:kaRw         Adj_Cat1a;   !  harsh

! CAT 1-b: Adjectives that end in a suffix -ah
taazah+Adjective:achch        Adj_Cat1b;   !  fresh

! CAT 2: Adjectives that do not end with a suffix
goal+Adjective:goal           Adj_Cat2;    ! round
sorkh+Adjective:sorkh         Adj_Cat2;    ! red
laal+Adjective:laal           Adj_Cat2;    ! red
baasee+Adjective:baasee       Adj_Cat2;    ! old
shareer+Adjective:shareer     Adj_Cat2;    ! naughty
meHnatee+Adjective:meHnatee   Adj_Cat2;    ! hard-worker

LEXICON Adj_Cat1a   ! Adjective Category 1-a
+Sg+Masc:aa               #;
+Sg+Masc+Oblique:ey       #;
+Pl+Masc:ey               #;
+Pl+Masc+Oblique:ey       #;
+Sg+Fem:ee                #;
+Sg+Fem+Oblique:ee        #;
+Pl+Fem:ee                #;
+Pl+Fem+Oblique:ee        #;

LEXICON Adj_Cat1b   ! Adjective Category 1-b
+Sg+Masc:ah               #;
+Sg+Masc+Oblique:ey       #;
+Pl+Masc:ey               #;
+Pl+Masc+Oblique:ey       #;
+Sg+Fem:ee                #;
+Sg+Fem+Oblique:ee        #;
+Pl+Fem:ee                #;
+Pl+Fem+Oblique:ee        #;

LEXICON Adj_Cat2   ! Adjective Category 2
+Sg+Masc:0               #;
+Sg+Masc+Oblique:0       #;
+Pl+Masc:0               #;
+Pl+Masc+Oblique:0       #;
+Sg+Fem:0                #;
+Sg+Fem+Oblique:0        #;
+Pl+Fem:0                #;
+Pl+Fem+Oblique:0        #;

LEXICON Aux_Verb
gaa+Aux+Future:g             Aux_Suffix1;  ! future tense
chokaa+Aux+Perfect:chok      Aux_Suffix1;  ! perfective aspect
```

```
rahaa+Aux+Progress:rah       Aux_Suffix1;  ! progressive aspect
chalaa+Aux+Cont:chal         Aux_Suffix1;  ! continuing progressive
lagaa+Aux+Incept:lag         Aux_Suffix1;  ! inceptive aspect
waalaa+Aux+Incept:waal       Aux_Suffix1;  ! inceptive aspect
paRaa+Aux+Comp:paR           Aux_Suffix1;  ! compulsive mood

thaa+Aux+Past:th             Aux_Suffix2;  ! past tense
hooaa+Aux+Decl:hoo           Aux_Suffix2;  ! declarative mood

hay+Aux+Present:h            Aux_Suffix3;  ! present tense

LEXICON Aux_Suffix1
+1st+Sg+Masc:aa             #;
+2nd+Rude+Masc:aa           #;
+3rd+Sg+Masc:aa             #;
+1st+Pl+Masc:ey             #;
+3rd+Pl+Masc:ey             #;
+2nd+Formal+Masc:ey         #;
+2nd+Polite+Masc:ey         #;
+1st+Sg+Fem:ee              #;
+2nd+Rude+Fem:ee            #;
+3rd+Sg+Fem:ee              #;
+1st+Pl+Fem:ee              #;
+3rd+Pl+Fem:ee              #;
+2nd+Formal+Fem:ee          #;
+2nd+Polite+Fem:ee          #;

LEXICON Aux_Suffix2
+1st+Sg+Masc:aa             #;
+2nd+Rude+Masc:aa           #;
+3rd+Sg+Masc:aa             #;
+1st+Pl+Masc:ey             #;
+3rd+Pl+Masc:ey             #;
+2nd+Formal+Masc:ey         #;
+2nd+Polite+Masc:ey         #;
+1st+Sg+Fem:ee              #;
+2nd+Rude+Fem:ee            #;
+3rd+Sg+Fem:ee              #;
+2nd+Formal+Fem:ee          #;
+1st+Pl+Fem:eeN             #;
+3rd+Pl+Fem:eeN             #;
+2nd+Polite+Fem:eeN         #;

LEXICON Aux_Suffix3
+1st+Sg+Masc:ooN           #;
+1st+Sg+Fem:ooN            #;
+1st+Pl+Masc:ayN           #;
+1st+Pl+Fem:ayN            #;
+3rd+Sg+Masc:ay            #;
+3rd+Pl+Masc:ay            #;
+3rd+Sg+Fem:ay             #;
+3rd+Pl+Fem:ay             #;
+2nd+Rude+Masc:ay          #;
+2nd+Rude+Fem:ay           #;
+2nd+Formal+Masc:ao        #;
+2nd+Formal+Fem:ao         #;
+2nd+Polite+Masc:ayN       #;
+2nd+Polite+Fem:ayN        #;
```

E.2 Morphology Syntax Interface Implementation

The Xerox Linguistics Environment (XLE) takes compiled finite state transducer input for the morphology and to interface it with syntax, the following code is used.

```
"Morphology-Syntax Interface Mapping"

PIEAS URDU MORPHOLOGY (1.0)

# Urdu Morph Config
TOKENIZE:
/home/jafar/xleHome/bin/default-parse-tokenizer.fsmfile
/home/jafar/xleHome/bin/default-gen-tokenizer.fst

ANALYZE:
urdu-morphology.fst

PARAMETERS:
*NOCAP
----
PIEAS URDU_MORPH RULES (1.0)

"Sublexical Rules"

   N -->  N-S_BASE: ^ = ! ;
          N-T_BASE: ^ = ! ;
          N-F_BASE*: ^ = ! ,
          C-F_BASE*: ^ = ! .

   V -->  V-S_BASE: ^ = ! ;
          V-T_BASE: ^ = ! ;
          V-F_BASE*: ^ = ! ,
          C-F_BASE*: ^ = ! .

   Adj --> Adj-S_BASE: ^ = ! ;
           Adj-T_BASE: ^ = ! ;
           N-F_BASE*: ^ = ! ,
           C-F_BASE*: ^ = ! .

   Aux --> AUX-S_BASE: ^ = ! ;
           AUX-T_BASE: ^ = ! ;
           AUX-F_BASE*: ^ = ! ,
           C-F_BASE*: ^ = ! .

----
MORPHOLOGY-BASED URDU LEXICON (1.0)

"Suffix '-S' representing Stems"

  -LUnknown N-S xle @(PRED %stem);
            Adj-S xle ^ = ! ;
            AUX-S xle ^ = ! ;
            ONLY.

  -Lunknown N-S xle   @(PRED %stem);
            Adj-S xle ^ = ! ;
            AUX-S xle ^ = ! ;
            ONLY.
```

```
" Verbs with Sub-categorisation frames .. "

hansnaa   V-S xle @(V-SUBJ %stem); ONLY.  "laugh"

paRhnaa   V-S xle @(V-SUBJ-OBJ %stem); ONLY.    "read"

maangnaa  V-S xle @(V-SUBJ-OBJ %stem); ONLY.    "ask for"

lekhnaa     V-S xle { @(V-SUBJ-OBJ %stem)
              | @(V-SUBJ-OBJ-INST %stem) }.     "write"

kaatnaa     V-S xle { @(V-SUBJ-OBJ-INST %stem)
                    | @(V-SUBJ-OBJ %stem) }.      "cut"

pohanchnaa  V-S xle @(V-SUBJ-OBJ %stem);  ONLY. "reach"

darnaa      V-S xle { @(V-SUBJ %stem)
              | @(V-SUBJ-OBJ %stem) };    ONLY. "fear"

chalnaa     V-S xle @(V-SUBJ %stem);      ONLY. "walk"

deykhnaa    V-S xle { @(V-SUBJ-OBJ %stem)
                    | @(V-SUBJ-COMP %stem) }.     "look"

chakhnaa    V-S xle @(V-SUBJ-OBJ %stem);  ONLY. "taste"

chakhaanaa  V-S xle @(V-SUBJ-OBJ-OBJ2 %stem);   ONLY. "taste, caus1"

chakhwaanaa V-S xle @(V-SUBJ-SUBJ2-OBJ2-OBJ %stem);   ONLY. "taste,
caus2"

lagnaa      V-S xle @(V-SUBJ-OBJ %stem);  ONLY.     "touch"

kholnaa     V-S xle { @(V-SUBJ-OBJ %stem)
                    ! @(V-SUBJ-OBJ-INST %stem) }.     "open"

bolnaa      V-S xle { @(V-SUBJ %stem)
              | @(V-SUBJ-COMP %stem) }.           "speak"

nekalnaa    V-S xle @(V-SUBJ-OBJ %stem);  ONLY. "come out"

xareednaa   V-S xle @(V-SUBJ-OBJ %stem);  ONLY. "buy"

poochhnaa   V-S xle { @(V-SUBJ %stem)
                    | @(V-SUBJ-COMP %stem) }.     "ask about, ask a
question"

chalaanaa   V-S xle @(V-SUBJ-OBJ %stem);  ONLY. "drive"

banaanaa    V-S xle @(V-SUBJ-OBJ %stem);  ONLY. "make"

nahaanaa    V-S xle @(V-SUBJ %stem);      ONLY. "bathe"

aanaa V-S xle @(V-SUBJ %stem);      ONLY. "come"

bolaanaa    V-S xle @(V-SUBJ-OBJ %stem);  ONLY. "call"

bolwaanaa   V-S xle @V-SUBJ-GOAL-OBJ(%stem);
                          ONLY. "make someone call someone"
```

```
rehnaa      V-S xle @(V-SUBJ %stem);      ONLY.      "stay"

kehnaa      V-S xle { @(V-SUBJ %stem)
              | @(V-SUBJ-COMP %stem) }.            "say"

cahnaa      V-S xle { @(V-SUBJ-OBJ %stem)
              | @(V-SUBJ-COMP %stem) }.            "want"

gernaa      V-S xle @(V-SUBJ %stem);      ONLY.     "fell"

khaansnaa   V-S xle @(V-SUBJ %stem);      ONLY. "cough"

khaanaa     V-S xle @(V-SUBJ-OBJ %stem);  ONLY. "eat"

khelaanaa   V-S xle @(V-SUBJ-OBJ-OBJ2 %stem);   ONLY. "eat - caus1"

khelwaanaa  V-S xle @(V-SUBJ-SUBJ2-OBJ2-OBJ %stem);
                            ONLY. "eat - caus2"

khaanaa N-S xle @(PRED %stem);          ONLY.   "food"

saonaa      V-S xle @(V-SUBJ %stem);      ONLY. "sleep"

raonaa      V-S xle @(V-SUBJ %stem);      ONLY. "weep"

seenaa      V-S xle @(V-SUBJ-OBJ %stem);  ONLY. "sew"

deynaa      V-S xle @(V-SUBJ-GOAL-OBJ %stem);   ONLY. "give"

leynaa      V-S xle @(V-SUBJ-GOAL-OBJ %stem);   ONLY. "take"

karnaa      V-S xle @(V-SUBJ-OBJ %stem);  ONLY. "do"

safar=karnaa      V-S xle @(V-SUBJ %stem);      ONLY. "travel"

aenteZaar=karnaa  V-S xle @(V-SUBJ %stem);      ONLY. "wait"

jaanaa      V-S xle @(V-SUBJ-OBJ %stem);  ONLY. "go"

"Suffix '-T' representing Tag"

  +Verb       V-T xle; ONLY.
  +Noun       N-T xle; ONLY.
  +Adjective  Adj-T xle; ONLY.
  +Aux        Aux-T xle; ONLY.

" --- Common Features --- "

" ... Number ... "

  +Sg C-F xle @(NUMBER sg);   ONLY.

  +Pl C-F xle @(NUMBER pl);   ONLY.

" ... Gender ... "

  +Fem      C-F xle @(GENDER fem);  ONLY.

  +Masc     C-F xle @(GENDER masc); ONLY.
```

```
" ... Person ... "

  +1st      C-F xle @(PERSON 1);    ONLY.

  +2nd      C-F xle @(PERSON 2);    ONLY.

  +3rd      C-F xle @(PERSON 3);    ONLY.

" ... Others ... "

" --- Verb Features --- "

" ... Tense ... "

  +Pres       V-F xle @(V-TENSE present);
            AUX-F xle @(V-TENSE present);   ONLY.

  +Past       V-F xle @(V-TENSE past);
            AUX-F xle @(V-TENSE past);      ONLY.

  +Future AUX-F xle @(V-TENSE future);
            V-F xle @(V-TENSE future)
            (^ V-FORM) =c subjunctive
            ((SUBJ ^) GEND) = (! GEND);     ONLY.

" ... Verb Form ... "

  +Root    V-F xle (^ V-FORM) = root;     ONLY.

  +Repeat  V-F xle (^ V-FORM) = repetitive;     ONLY.

  +Perf    V-F xle (^ V-FORM) = perfective;     ONLY.

  +Inf         V-F xle (^ V-FORM) = infinitive ;      ONLY.

  +Obl         V-F xle (^ V-FORM2) = oblique ;  ONLY.

  +Caus1   V-F xle (^ V-FORM2) = causative1 ;   ONLY.

  +Caus2   V-F xle (^ V-FORM2) = causative2 ;   ONLY.

  +Subj        V-F xle (^ V-FORM) = subjunctive
                       ((SUBJ ^) NUM) = (! NUM)
                       ((SUBJ ^) PERS) = (! PERS);   ONLY.

  +Impr  V-F xle (^ V-FORM) = imperative ;      ONLY.


" ... 2nd Person Honorific Verb Forms ... "

  +Rude     V-F xle (^ V-HFORM) = rude ;  ONLY.

  +Formal   V-F xle (^ V-HFORM) = formal ;      ONLY.

  +Polite   V-F xle (^ V-HFORM) = polite ;      ONLY.

  +Request  V-F xle (^ V-HFORM) = request ;     ONLY.

"Noun Features"

" ... Noun Class ... "
```

```
  +Proper   N-F xle (^ N-SEM N-CLASS) = proper
                @(NUMBER sg);     ONLY.

  +Common   N-F xle (^ N-SEM N-CLASS) = common; ONLY.

" ... Noun Type ... "

  +Count   N-F xle (^ N-SEM N-TYPE) = count;    ONLY.

  +Mass        N-F xle (^ N-SEM N-TYPE) = mass; ONLY.


" ... Noun Concept (N-CONCEPT)... "

  +Abstract N-F xle (^ N-SEM N-CONCEPT) = abstract;   ONLY.

  +Group    N-F xle (^ N-SEM N-CONCEPT) = group;      ONLY.

  +Spatial  N-F xle (^ N-SEM N-CONCEPT) = spatial;    ONLY.

  +Temporal N-F xle (^ N-SEM N-CONCEPT) = temporal;   ONLY.

  +Instrument     N-F xle (^ N-SEM N-CONCEPT) = instrument; ONLY.

  +Animate  N-F xle @(N-CONCEPT animate); ONLY.

  +Thing    N-F xle (^ N-SEM N-CONCEPT) = thing;      ONLY.

" ... Noun Form (N-FORM) ... "

  +Nominative     N-F xle (^ N-FORM) = nominative;    ONLY.

  +Oblique  N-F xle (^ N-FORM) = oblique; ONLY.

  +Vocative N-F xle (^ N-FORM) = vocative;      ONLY.

" ... Noun Base Language ... "

  +Arabic   N-F xle (^ N-SEM N-LANG) = arabic;  ONLY.

  +Persian  N-F xle (^ N-SEM N-LANG) = persian; ONLY.

  +Hindi    N-F xle (^ N-SEM N-LANG) = hindi;   ONLY.

  +Turkish  N-F xle (^ N-SEM N-LANG) = turkish; ONLY.

  +English  N-F xle (^ N-SEM N-LANG) = english; ONLY.

"Auxiliary Features"

" ... Aspect ... "

  +Perfect  AUX-F xle @(V-ASPECT perfective);   ONLY.

  +Progress  AUX-F xle @(V-ASPECT progressive); ONLY.

  +Cont  AUX-F xle (^ TNS-ASP ACTION) = continuous;   ONLY.

  +Incept  AUX-F xle @(V-ASPECT inceptive);     ONLY.
```

```
  +Comp  AUX-F xle @(V-ASPECT compulsive);      ONLY.

" ... Mood ... "

  +Decl  AUX-F xle @(V-MOOD declarative); ONLY.

----
```

E.3 Syntax Implementation

The Xerox Linguistics Environment (XLE) takes LFG based syntax rules to generate c-structures and f-structures. The following is the listing of rules to analyse Urdu sentences.

```
PIEAS URDU CONFIG (1.0)
  ROOTCAT   S.
  FILES Pronouns.lfg Templates.lfg VerbMorphemes.lfg .
  LEXENTRIES (all all).
  RULES      (PIEAS URDU_SYN) (PIEAS URDU_MORPH).
  TEMPLATES  (PIEAS URDU).
  MORPHOLOGY (PIEAS URDU).
  FEATURES   (PIEAS URDU).
  GOVERNABLERELATIONS  SUBJ SUBJ2 OBJ OBJ2 OBL-?+ COMP.
  SEMANTICFUNCTIONS    ADJUNCT TOPIC FOCUS.
  EPSILON   e.
----
PIEAS URDU FEATURES (1.0)
NUM: -> $ { sg pl }.
PERS: -> $ { 1 2 3 }.
GEND: -> $ { fem masc }.
CASE: -> $ { nom erg dat acc agent mutual instrument temporal
movement adverbial}.
N-SEM: -> << [ N-CONCEPT N-TYPE N-CLASS N-LANG H-MOOD DIST ].
N-TYPE: -> $ { count mass }.
N-CLASS: -> $ { common proper }.
N-LANG: -> $ { arabic persian hindi turkish english }.
N-CONCEPT: -> $ {abstract group spatial temporal instrument animate
thing}.
H-MOOD: -> $ { rude formal polite request }.
DIST: -> $ { near far }.
N-FORM: -> $ { nominative oblique vocative }.
V-FORM: -> $ { root perfective repetitive infinitive subjunctive
imperative }.
V-FORM2: -> $ { oblique causative1 causative2 }.
V-HFORM: -> $ { rude formal polite request }.
V-TENSE: -> $ { present past future }.
V-VAL: -> $ { 1 2 3 4 }.
TNS-ASP: -> << [ TENSE MOOD ASPECT ACTION VOICE].
TENSE: -> $ { present past future }.
MOOD: -> $ { indicative subjunctive permissive imperative }.
ASPECT.
ACTION.
VOICE: -> $ { active passive }.
P-CASE.
SPEC: -> $ { definite indefinite }.

----
```

```
PIEAS URDU TEMPLATES (1.0)

   GENDER(_G_) = (^ GEND) = _G_.

   NUMBER(_N_) = (^ NUM) = _N_.

   PERSON(_P_) = (^ PERS) = _P_.

   PRED(_P_) = (^ PRED) = '_P_'.

   V-SUBJ(_P_) = (^ PRED)='_P_<(^ SUBJ)>'
                    (^ V-VAL) = 1.

   V-SUBJ-OBJ(_P_) = (^ PRED)='_P_<(^ SUBJ)(^ OBJ)>'
                    (^ V-VAL) = 2.

   V-SUBJ-COMP(_P_) =  (^ PRED)='_P_<(^ SUBJ)(^ COMP)>'
                    (^ V-VAL) = 2.

   V-SUBJ-XCOMP(_P_) =  (^ PRED)='_P_<(^ SUBJ)(^ XCOMP)>'
                    (^ V-VAL) = 2.

   V-SUBJ-OBJ-OBJ2(_P_) = (^ PRED)='_P_<(^ SUBJ)(^ OBJ2)(^ OBJ)>'
                    (^ V-VAL) = 3.

   V-SUBJ-OBJ-INST(_P_) = (^ PRED)
                       ='_P_<(^ SUBJ)(^ OBJ)(^ OBL-sey-inst)>'
                             (^ V-VAL) = 3.

   V-SUBJ-SUBJ2-OBJ(_P_) = (^ PRED)='_P_<(^ SUBJ)(^ SUBJ2)(^ OBJ)>'
                    (^ V-VAL) = 3.

   V-SUBJ-SUBJ2-OBJ2-OBJ(_P_) = (^ PRED)
                     ='_P_<(^ SUBJ)(^ SUBJ2)(^ OBJ2)(^ OBJ)>'
                    (^ V-VAL) = 4.

   N-CASE(_C_) = (^ CASE) = _C_.

   N-FORM(_F_) = (^ N-FORM) = _F_.

   POSTPOSITION( _P_ _C_ ) = (^ PRED) = '_P_<(^ OBJ)>'
                               (^ P-CASE) = _C_.

   V-TENSE(_T_) = (^ TNS-ASP TENSE) = _T_.

   V-VOICE(_V_) = (^ TNS-ASP VOICE) = _V_.

   V-ASPECT(_A_) = (^ TNS-ASP ASPECT) = _A_.

   V-MOOD(_M_) = (^ TNS-ASP MOOD) = _M_.

   N-CONCEPT(_C_) = (^ N-SEM N-CONCEPT) = _C_.

   N-H-MOOD(_M_) = (^ N-SEM H-MOOD) = _M_.

   DIST( _D_ ) = (^ N-SEM DIST) = _D_.

   ADJUNCT = ! $ (^ ADJUNCT).

   GF = {   (^ SUBJ)=!
        | (^ OBJ)=!
```

```
        | (^ OBJ2)=!
        }.

   OBLF = {   (^ OBL-sey-inst)=!
            | (^ OBL-sey-temp)=!
          | (^ SUBJ2)=!
          "| (^ OBJ)=!"
          }.

----

PIEAS URDU_SYN RULES (1.0)

   S --> {
         S_verb
         |
         S_perf
         }.

   S_verb --> NP#1#5: { @GF
                |
                 @OBLF
               },

               (PP#1#3:
               @ADJUNCT
               ),

               { V2: ^ = !
                |
                 V1: ^ = !
                 { " for "
                   (^ V-FORM) = perfective "perfect"
                   { (^ V-VAL) = 2 "transitive verb"
                    |(^ V-VAL) = 3 "or ditransitive verb"
                    |(^ V-VAL) = 4}
                   " we need "
                   (^ SUBJ CASE) =c erg "ergative suject"
                   {
                     (^ OBJ CASE) ~= acc
                     (^ OBJ GEND) = (! GEND) "object-verb agreement"
                     (^ OBJ NUM) = (! NUM)
                    |
                     (^ OBJ CASE) =c acc
                     (^ GEND) = masc "default sg-masc agreement"
                     (^ NUM) = sg
                   }

                  |

                   (^ V-FORM) = repetitive
                   { (^ V-VAL) = 2 "transitive verb"
                    |(^ V-VAL) = 3 "or ditransitive verb"
                    |(^ V-VAL) = 4}
                   (^ SUBJ CASE) =c nom
                   (^ SUBJ GEND) =c (! GEND) "suject-verb agreement"
                   (^ SUBJ NUM) =c (! NUM)
                   (^ SUBJ N-SEM N-CONCEPT) =c animate

                  |
```

```
                (^ V-VAL) = 1 "intranstive verb"
                (^ SUBJ CASE) =c nom
                (^ SUBJ GEND) =c (! GEND) "suject-verb agreement"
                (^ SUBJ NUM) =c (! NUM)
                (^ SUBJ N-SEM N-CONCEPT) =c animate

             }
          }.

   S_perf --> NP: (^ SUBJ)=!
                  (^ CASE) =c nom
                  (^ N-SEM N-CONCEPT) =c animate;
              NP: (^ OBJ)=!
                  (^ CASE) =c nom;
              V:  ^ = !
                  (^ V-FORM) =c root;
              Aux: ^ = !
                  (^ TNS-ASP ASPECT) =c perfective
                  (^ SUBJ PERS) =c (! PERS)
                  (^ SUBJ NUM) =c (! NUM);
              Aux: ^ = !
                  (^ SUBJ PERS) =c (! PERS)
                  (^ SUBJ NUM) =c (! NUM).


   V2 --> VS VM.    " without finite state morphology, auxiliaries
lumped into morphmes which are being joined at syntax level "

   V1 --> V : ^ = !; " with finite state morphology "
        (Aux*: ^ = !).

   NP --> (Adj:
          (^ NUM) =c (! NUM)
          (^ GEND) =c (! GEND)
          ) " optional Adjective "
        { N: ^ = !; "either a case marked noun"
          Case: ^ = !
        |
          Pronoun: ^ = !; "or a case marked pronoun"
          Case: ^ = !
        |
          N: (^ CASE) = nom "or an unmarked noun"
             { (OBJ ^) | (SUBJ ^)}
        |
          Pronoun: (^ CASE) = nom "or an unmarked pronoun"
                 { (OBJ ^) | (SUBJ ^)}
             }.

   PP --> N: (^ OBJ) = !;
        PostPos: ^ = !
                   (ADJUNCT ($) ^).

----
URDU LEX LEXICON (1.0)

" ~~~~~~~~~~~~~~~~
    Case Clitics
  ~~~~~~~~~~~~~~~~ "

ney     Case * @(N-CASE erg)
             @(N-FORM oblique)
```

```
             (^ N-SEM N-CONCEPT) =c animate
             (SUBJ ($) ^).

kao     Case * @(N-FORM oblique)
             {
               " either 'kao' marks a dative case "
               @(N-CASE dat)
               (^ N-SEM N-CONCEPT) =c animate

               { " either it becomes 'goal' or 'indirect object' OBJ2
"
                 (OBJ2 ($) ^)

               | " or "

                 " sometimes dative case acts as a SUBJ "
                 (SUBJ ($) ^)
               }

             | " or "

               " 'kao' marks an accusative case "
               @(N-CASE acc)
               " which acts as a direct object "
               (OBJ ($) ^)

               {
                 (^ N-SEM N-CONCEPT) =c animate " if animate then ok
"
                |
                 (^ N-SEM N-CONCEPT) =c thing " but if a thing "
                 (^ SPEC) =c definite " it requires a specifier "
               }

             }.

sey     Case * @(N-FORM oblique)
             " either 'sey' marks an animated noun "
             {
               @(N-CASE agent)
               (^ N-SEM N-CONCEPT) =c animate
               {
                 (SUBJ ($) ^) " it is subject in the absence of
ergative case "
                |
                 (SUBJ2 ($) ^) " else it is secondary subject in the
presense of 'ney' "
               }

             | " or "

               @(N-CASE mutual) " Both Subject & Object are animate
and a mutual verb "
               ((OBJ ^) SUBJ N-SEM N-CONCEPT) =c animate
               ((OBJ ^) OBJ N-SEM N-CONCEPT) =c animate
               (OBJ ($) ^)

             | " or "

               " marks an instrumental noun "
```

```
                @(N-CASE instrument)
                (^ N-SEM N-CONCEPT) =c instrument
                (OBL-sey-inst ($) ^)


              | " or "

                " marks a temporal noun "

                @(N-CASE temporal)
                (^ N-SEM N-CONCEPT) =c temporal
                (OBL-sey-temp ($) ^)
              }.


VERBMORPHEMES LEX LEXICON (1.0)

" ~~~~~~~~~~~~~~~~
    Verb Morphemes
  ~~~~~~~~~~~~~~~~  "

ee          VM * @(V-TENSE past)
                 (^ OBJ NUM) = sg
                 (^ OBJ GEND) = fem
                 (^ SUBJ CASE) =c erg.

aa          VM * @(V-TENSE past)
                 {
                   (^ OBJ CASE) ~= acc
                   (^ OBJ GEND) = masc "object-verb agreement"
                   (^ OBJ NUM) = sg
                 |
                   (^ OBJ CASE) =c acc
                   (^ GEND) = masc       "default sg-masc agreement"
                   (^ NUM) = sg
                 }
                 (^ SUBJ CASE) =c erg.

ey          VM * @(V-TENSE past)
                 (^ OBJ NUM) = pl
                 (^ OBJ GEND) = masc
                 (^ SUBJ CASE) =c erg.

eeN         VM * @(V-TENSE past)
                 (^ OBJ NUM) = pl
                 (^ OBJ GEND) = fem
                 (^ SUBJ CASE) =c erg.

ee-hay      VM * @(V-TENSE present)
                 @(V-ASPECT perfect)
                 (^ OBJ NUM) = sg
                 (^ OBJ GEND) = fem
                 (^ SUBJ CASE) =c erg
                 (^ OBJ CASE) ~= acc.

aa-hay      VM * @(V-TENSE present)
                 @(V-ASPECT perfect)
                 (^ SUBJ CASE) =c erg
                 { (^ OBJ NUM) = sg
                 | (^ OBJ NUM) = pl
                   (^ OBJ CASE) =c acc}
```

```
                { (^ OBJ GEND) = masc
                | (^ OBJ GEND) = fem
                  (^ OBJ CASE) =c acc}.

eeN-hayN  VM * @(V-TENSE present)
               @(V-ASPECT perfect)
               (^ OBJ NUM) = pl
               (^ OBJ GEND) = fem
               (^ OBJ CASE) ~= acc
               (^ SUBJ CASE) =c erg.

ey-hayN   VM * @(V-TENSE present)
               @(V-ASPECT perfect)
               (^ OBJ NUM) = pl
               (^ OBJ GEND) = masc
               (^ SUBJ CASE) =c erg.

ee-thee   VM * @(V-TENSE past)
               @(V-ASPECT perfect)
               (^ OBJ NUM) = sg
               (^ OBJ GEND) = fem
               (^ SUBJ CASE) =c erg.

aa-thaa   VM * @(V-TENSE past)
               @(V-ASPECT perfect)
               (^ SUBJ CASE) =c erg
               { (^ OBJ NUM) = sg
               | (^ OBJ NUM) = pl
                 (^ OBJ CASE) =c acc}
               { (^ OBJ GEND) = masc
               | (^ OBJ GEND) = fem
                 (^ OBJ CASE) =c acc}.

eeN-theeN VM * @(V-TENSE past)
               @(V-ASPECT perfect)
               (^ OBJ NUM) = pl
               (^ OBJ GEND) = fem
               (^ SUBJ CASE) =c erg.

ey-they   VM * @(V-TENSE past)
               @(V-ASPECT perfect)
               (^ OBJ NUM) = pl
               (^ OBJ GEND) = masc
               (^ SUBJ CASE) =c erg.

ooN-gaa   VM * @(V-TENSE future)
               (^ SUBJ PERS) = 1
               (^ SUBJ NUM) = sg
               (^ SUBJ GEND) = masc
               (^ SUBJ CASE) =c nom.

ooN-gee   VM * @(V-TENSE future)
               (^ SUBJ PERS) = 1
               (^ SUBJ NUM) = sg
               (^ SUBJ GEND) = fem
               (^ SUBJ CASE) =c nom.

eyN-gey    VM * @(V-TENSE future)
               {(^ SUBJ PERS) = 1
                |(^ SUBJ PERS) = 3}
               (^ SUBJ NUM) = pl
```

```
                (^ SUBJ GEND) = masc
                (^ SUBJ CASE) =c nom.

eyN-gee    VM * @(V-TENSE future)
                { (^ SUBJ PERS) = 1
                | (^ SUBJ PERS) = 3 }
                (^ SUBJ NUM) = pl
                (^ SUBJ GEND) = fem
                (^ SUBJ CASE) =c nom.

ao-gey     VM * @(V-TENSE future)
                (^ SUBJ PERS) = 2
                { (^ SUBJ NUM) = sg
                | (^ SUBJ NUM) = pl }
                (^ SUBJ GEND) = masc
                (^ SUBJ CASE) =c nom.

ao-gee     VM * @(V-TENSE future)
                (^ SUBJ PERS) = 2
                { (^ SUBJ NUM) = sg
                | (^ SUBJ NUM) = pl }
                (^ SUBJ GEND) = fem
                (^ SUBJ N-SEM N-CONCEPT) =c animate
                (^ SUBJ CASE) =c nom.

ey-gaa     VM * @(V-TENSE future)
                (^ SUBJ PERS) = 3
                (^ SUBJ NUM) = sg
                (^ SUBJ GEND) = masc
                (^ SUBJ N-SEM N-CONCEPT) =c animate
                (^ SUBJ CASE) =c nom.

ey-gee     VM * @(V-TENSE future)
                (^ SUBJ PERS) = 3
                (^ SUBJ NUM) = sg
                (^ SUBJ GEND) = fem
                (^ SUBJ N-SEM N-CONCEPT) =c animate
                (^ SUBJ CASE) =c nom.

aa-gayaa   VM * @(V-TENSE past)
                @(V-VOICE passive)
                @(V-ASPECT perfect)
                 (^ SUBJ CASE) =c agent.

" ~~~~~~~~~~~~~~~~~~ "

khaayaa-naheeN-jaataa V2 *  @(V-SUBJ-OBJ khaanaa)
                            (^ SUBJ CASE) =c agent.


baat-kee   V2 * @(V-SUBJ-OBJ baat-karnaa)
             (^ SUBJ CASE) =c erg
             (^ OBJ CASE) =c mutual
             (^ SUBJ N-SEM N-CONCEPT) =c animate
             (^ OBJ N-SEM N-CONCEPT) =c animate.

madad-maangee  V2 * @(V-SUBJ-OBJ madad-maangnaa)
             (^ SUBJ CASE) =c erg
             (^ OBJ CASE) =c mutual
             (^ SUBJ N-SEM N-CONCEPT) =c animate
             (^ OBJ N-SEM N-CONCEPT) =c animate.
```

```
waAdah-keeaa  V2 * @(V-SUBJ-OBJ waAdah-karnaa)
             (^ SUBJ CASE) =c erg
             (^ OBJ CASE) =c mutual
             (^ SUBJ N-SEM N-CONCEPT) =c animate
             (^ OBJ N-SEM N-CONCEPT) =c animate.

sawaal-poochhaa  V2 * @(V-SUBJ-OBJ sawaal-poochhnaa)
             (^ SUBJ CASE) =c erg
             (^ OBJ CASE) =c mutual
             (^ SUBJ N-SEM N-CONCEPT) =c animate
             (^ OBJ N-SEM N-CONCEPT) =c animate.

----

PRONOUN LEX LEXICON (1.0)

" ~~~~~~~~~~~~~~~~
     Pronouns
  ~~~~~~~~~~~~~~~~ "

mayN Pronoun * @(PRED %stem)
               @(NUMBER sg)
               @(PERSON 1)
               @(N-CASE nom)
               @(N-CONCEPT animate).

ham Pronoun * @(PRED %stem)
              @(NUMBER sg)
              @(PERSON 3)
              @(N-CONCEPT animate).

too Pronoun * @(PRED %stem)
              @(NUMBER sg)
              @(PERSON 2)
              @(N-H-MOOD rude)   "Honor Mood"
              @(N-CONCEPT animate).

tom  Pronoun * @(PRED %stem)
               { @(NUMBER sg)
                 |
                 @(NUMBER pl) }
               @(PERSON 2)
               @(N-H-MOOD formal)
               @(N-CONCEPT animate).

aap  Pronoun * @(PRED %stem)
               { @(NUMBER sg)
                 |
                 @(NUMBER pl) }
               @(PERSON 2)
               { @(N-H-MOOD polite)
                 |
                 @(N-H-MOOD respect) }
               @(N-CONCEPT animate).

aes  Pronoun * @(PRED %stem)
               @(NUMBER sg)
               @(PERSON 3)
               (^ CASE) ~= nom
               @(DIST near)
```

```
                { @(N-CONCEPT animate)
                  |
                  @(N-CONCEPT thing)}.

aes  Det      * (^ SPEC) = definite
                @(DIST near).

aos  Det      * (^ SPEC) = definite
                @(DIST far).

yeh  Pronoun * @(PRED %stem)
                @(NUMBER sg)
                @(PERSON 3)
                (^ CASE) = nom
                @(DIST near)
                { @(N-CONCEPT animate)
                  |
                  @(N-CONCEPT thing) }.

aos  Pronoun * @(PRED %stem)
                @(NUMBER sg)
                @(PERSON 3)
                (^ CASE) ~= nom
                @(DIST far)
                { @(N-CONCEPT animate)
                  |
                  @(N-CONCEPT thing) }.

woh  Pronoun * @(PRED %stem)
                @(NUMBER sg)
                @(PERSON 3)
                { @(GENDER fem)
                 | @(GENDER masc) }
                (^ CASE) = nom
                @(DIST far)
                { @(N-CONCEPT animate)
                  |
                  @(N-CONCEPT thing) }.

aenhooN  Pronoun * @(PRED %stem)
                @(NUMBER pl)
                @(PERSON 3)
                (^ FORM) = oblique
                (^ CASE) =c erg
                @(DIST near)
                @(N-CONCEPT animate).

aonhooN  Pronoun * @(PRED %stem)
                @(NUMBER pl)
                @(PERSON 3)
                (^ FORM) = oblique
                (^ CASE) =c erg
                @(DIST far)
                @(N-CONCEPT animate).

aen  Pronoun * @(PRED %stem)
                @(NUMBER pl)
                @(PERSON 3)
                (^ FORM) = oblique
                (^ CASE) ~= nom
                @(DIST near)
```

```
                @(N-CONCEPT animate).

aon  Pronoun * @(PRED %stem)
               @(NUMBER pl)
               @(PERSON 3)
               (^ FORM) = oblique
               (^ CASE) ~= nom
               @(DIST far)
               @(N-CONCEPT animate).
----

" ~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    Post Positions
  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~  "

meyN    PostPos * @(N-FORM oblique)
                  @(POSTPOSITION meyN OBLin)
                " (ADJUNCT ($) ^)".


sey_pp        PostPos * @(POSTPOSITION sey OBLsey)
                  @(N-FORM oblique)
                "(ADJUNCT ($) ^) ".



" ~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    Verb Stems (without FSM)
  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~  "

xareed        VS * @(V-SUBJ-OBJ xareednaa).    " buy "

maar    VS * @(V-SUBJ-OBJ maarnaa)
             { (^ OBJ CASE) =c acc      " beat "
             | (^ OBJ CASE) =c nom }.   " kill "

lekh    VS * @(V-SUBJ-OBJ lekhnaa).       "write"

" ~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    Nouns (without FSM)
  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~  "

skool       N * @(PRED %stem).  " school "

jaldee      N * @(PRED %stem).   " hurriedly "
shaoq       N * @(PRED %stem).   " interest "
tawajah     N * @(PRED %stem).   " concentration "
sardee      N * @(PRED %stem).   " cold "

----
```

REFERENCES

Abdul-Haq, M. (1991). Qwaed-e-Urdu. New Delhi, Anjuman Taraqi-e-Urdu.

Abney, S., Ed. (1991). Parsing by chunks. Principle-Based Parsing, Kluwer Academic Publishers.

Afzal, M. and S. Hussain (2001). Urdu Computing Standards: Urdu Zabta Takhti (UZT) 1.01. IEEE International Multitopic Conference INMIC 2001, Lahore, LUMS.

Aho, A. V., R. Sethi, et al. (1986). Compilers: Principles, Techniques, and Tools. Boston, MA, USA, Addison-Wesley Longman Publishing Co., Inc.

Arnold, D., L. Balkan, et al. (1994). Machine Translation: An Introductory Guide. London, NCC Blackwell.

Arsenault, P. E. (2002). Toward an HPSG Account of Case in Hindi, University of Hyderabad.

Beaven, J. L. (1992). Shake and Bake Machine Translation. Proceedings of the 14th conference on Computational linguistics, Nantes, France, Association for Computational Linguistics.

Beesley, K. R. and L. Karttunen (2003). Finite State Morphology, CSLI Publications.

Bhatt, R. and D. Embick (2003). Causative Derivations in Hindi. Austin.

Bod, R., R. Scha, et al. (2003). Data-Oriented Parsing. California, CSLI Publications.

Bresnan, J., Ed. (1982). The Mental Representation of Grammatical Relations, MIT Press.

Bresnan, J. (2001). Lexical-Functional Syntax. Oxford, Blackwell Publishers.

Bresnan, J. (2001). Lexical Functional Syntax. Surrey, Blackwell.

Buchholz, S., J. Veenstra, et al. (1999). Cascaded Grammatical Relation Assignment. EMNLP/VLC-99, University of Maryland, USA.

Butt, M. (1995). The Structure of Complex Predicates in Urdu. Stanford, California, CSLI Publications.

Butt, M. (2003). The Morpheme That Would'nt Go Away. London.

Butt, M. (2003). Tense and Aspect in Urdu. Paris.

Butt, M. (2005). Theories of Case, Cambridge University Press.

Butt, M. and T. H. King (1999). Licensing Semantic Case, University of Konstanz and Xerox PARC.

Butt, M. and T. H. King (2002). The Status of Case.

Butt, M. and T. H. King (2006). Restriction for Morphological Valency Alternations: The Urdu Causative. Intelligent Linguistic Architectures: Variations on Themes by Ronald M. Kaplan. M. Butt, M. Dalrymple and T. H. King. Stanford, CA, CSLI Publications**:** 235-258.

Butt, M., T. H. King, et al. (2002). The Parallel Grammar Project. Proceedings of the Workshop on Grammar Engineering and Evaluation.

Butt, M., M. i.-E. Niño, et al. (1999). A Grammar Writer's Cookbook. Stanford, CA, CSLI Publications.

Chomsky, N. (1993). Lectures on Government and Binding. Berlin, Walter de Gruyter & Co.

Ciura, M. G. and S. Deorowicz (2001). "How to squeeze a lexicon." Software-Practice and Experience **31**(11): 1077-1090.

Cormen, T. H., C. E. Leiserson, et al. (1994). Introduction to Algorithms. Cambridge, The MIT Press.

Daciuk, J. (1998). Incremental Construction of Finite-State Automata and Transducers, and their Use in the Natural Language Processing, Politechnika Gda?ska.

Dalrymple, M. (2001). Lexical Functional Grammar. New York, Academic Press.

Dorr, B. J. (2000). "A Survey of Current Paradigms in Machine Translation." Advances in Computers.

Feroz-ud-Din (2000). Feroz ul Lughat - Urdu - Jamay. Lahore, Feroz Sons.

Grune, D. and C. Jacobs (1994). Parsing Techniques - A Practical Guide, Ellis Horwood Limited.

Hardie, A. (2004). The Computational Analysis of Morphosyntactic Categories in Urdu. Department of Linguistics, Lancaster University. **Ph.D. Thesis**.

Hopcroft, J. E. and J. D. Ullman (1979). Introduction to Automata Theory, Languages and Computation, Addison Wesley.

Hussain, S. (2004). Finite-State Morphological Analyzer for Urdu. Department of Computer Science. Lahore, National University of Computer and Emerging Sciences. **M.S. (Computer Science)**.

Hutchins, W. J. and H. L. Somers (1997). An Introduction to Machine Translation. London, Academic Press.

Kaplan, R. M. and M. Kay (1994). "Regular Models of Phonological Rule Systems." Computational Linguistics **20**(3): 331-378.

Karttunen, L. "Application of Finite-State Transducers in Natural Language Processing."

Karttunen, L. (1994). Constructing Lexical Transducers. COLING'94.

Khan, M. A. (1995). Text Based Machine Translation. Peshawar, Peshawar University.

Knuth, D. E. (1998). Sorting and Searching, Addison-Wesley.

Leech, G. and A. Wilson, Eds. (1999). Standards for Tagsets. Recommendations for the Morphosyntactic Annotations of Corpora. Syntactic Wordclass Tagging. Dordrecht, Kluwer Academic Publishers.

Luger, G. F. and W. A. Stubblefield (1998). Artificial Intelligence, Addison Wesley.

Manning, C. D. and H. Schütze (2003). Foundations of Statistical Natural Language Processing. London, The MIT Press.

Martin, J. C. (1991). Introduction to Languages and the Theory of Computation, McGraw Hill.

Mihov, S. Direct construction of Minimal Acyclic Finite State Automata.

Mohanan, T. (1990). Argument Structure in Hindi, Stanford University, Department of Linguistics.

Mohanan, T. (1994). Argument Structure in Hindi. Stanford, CA, CSLI Publications.

Mustafa, G. (1973). Jamay ul Qwaed. Lahore, Markazi Urdu Board.

Naruedomkul, K. and N. Cercone (2002). "Generate and Repair Machine Translation." Computational Intelligence **18**(3): 254-269.

Neidle, C. Lexical Functional Grammar.

Nordlinger, R. (1998). Constructive Case: Dependent Marking Nonconfigurationality in Australia. Stanford, CA, CSLI Publications.

Partow, A. from http://www.partow.net/programming/hashfunctions/.

Platts, J. T. (1884). A Dictionary of Urdu, Classical Hindi, and English. Oxford, Oxford University Press.

Pollard, C. J. and I. A. Sag (1987). Information-based Syntax and Semantics, Vol. 1. Stanford University, CSLI Publications.

Pollard, C. J. and I. A. Sag (1994). Head-Driven Phrase Structure Grammar. Chicago, University of Chicago Press.

Ramshaw, L. A. and M. P. Marcus (1995). Text Chunking Using Transformation-Based Learning. Third ACL Workshop on Very Large Corpora, Cambridge MA, USA.

Rizvi, S. M. J. and M. Hussain (2002). "Framework for the Syntactic Machine Translation between English and Urdu Languages." Science International **14**(3): 187-190.

Rizvi, S. M. J. and M. Hussain (2002). A Novel Approach to Account Morphological Behavior of Urdu Verbs to Model Urdu Tenses Using LFG. IEEE INMIC, Karachi, IEEE.

Rizvi, S. M. J. and M. Hussain (2004). Utilization of a Novel Ordered Context Free Grammar for Object Based Parsing and Unification Technique. NCET 2004, SZABIST Karachi.

Rosetta, M. T. (1994). Compositional Translation. Dordrecht, The Netherlands, Kluwer Academic Publishers.

Sag, I. A., T. Wasow, et al. (2004). Syntactic Theory: A Formal Introduction. Stanford, California, CSLI Publications.

Schmidt, R. L. (1999). Urdu: An Essential Grammar. London, Routledge.

Sedgwick, R. (1988). Algorithms, Addison-Wesley Publishing Company.

Trujillo, A. (1999). Translation Engines: Techniques for Machine Translation. London, Springer-Verlag.

Veenstra, J. (1999). Memory-Based Text Chunking. Machine learning in human language technology, Chania, Greece.

Wescoat, T. W. Practical Instructions for Working with the Formalism of Lexical Functional Grammar.

Yao, Y. and K. T. Lua (1998). "A Probabilistic Context-Free Grammar Parser for Chinese." Computer Processing of Oriental Languages **11**(4): 393-407.

PAPERS PUBLISHED DURING THE RESEARCH

[1] S. M. Jafar Rizvi, Mutawarra Hussain, "Framework for the Syntactic Machine Translation between English and Urdu Languages," Science International, Vol. 14, No. 3, pp187, 2002.

[2] S. M. Jafar Rizvi, Mutawarra Hussain, "A Novel Approach to Account Morphological Behavior of Urdu Verbs to Model Urdu Tenses Using LFG," In the Proceedings of IEEE's INMIC 2002, Karachi, 2001.

[3] S. M. Jafar Rizvi, Mutawarra Hussain, "Unified Compression and Encryption Algorithm for Fast and Secure Network Communications," Science International, Vol. 17, No. 2, pp95-99, 2005.

[4] S. M. Jafar Rizvi, Mutawarra Hussain, "Utilization of a Novel Ordered Context Free Grammar for Object Based Parsing and Unification Technique," In the Proceedings of IEEE/ACM NCET 2004, SZABIST, Karachi, 2004.

[5] S. M. Jafar Rizvi, Mutawarra Hussain, "Language Oriented Parsing through Morphologically Closed Word Classes in Urdu," In the Proceedings of IEEE's SCONEST 2004, FJWU, Karachi, 2004.

[6] S. M. Jafar Rizvi, Mutawarra Hussain, "Comparison of Hash Table verses Lexical Transducer based Implementations of Urdu Lexicon," In the Proceedings of IEEE's SCONEST 2004, FJWU, Karachi, 2004.

[7] S. M. Jafar Rizvi, Mutawarra Hussain, "Language Oriented Parsing of Urdu through Chunking and Ordered Context Free Grammar," Program and Paper Abstracts – International Conference on Software Engineering & Applications, ICSEA-2004, Islamabad, pp 21, 2004.

[8] S. M. Jafar Rizvi, Mutawarra Hussain, "Mathematical Modeling based on Head-Driven-Phrase-Structure-Grammar for Urdu Language," presented at Second World Conference on 21st Century Mathematics 2005, Lahore.

[9] S. M. Jafar Rizvi, Mutawarra Hussain, "Investigation of Urdu Case and Tense System under Head driven Phrase Structure Grammar," In the Proceedings of National Conference on Information Technology Applications, NC-ITA-2005, Quetta.

[10] S. M. Jafar Rizvi, Mutawarra Hussain, "Analysis, Design and Implementation of Urdu Morphological Analyzer", In the Proceedings of IEEE SCONEST 2005, NED University of Engineering and Technology, Karachi, 2005.

[11] S. M. Jafar Rizvi, Mutawarra Hussain, "Noun-Case and Verbal Agreement in Grammar Modeling for Urdu Language", In the Proceedings of IEEE International Conference on Natural Language Processing and Knowledge Engineering, NLP-KE 2005, Wuhan, China, 2005.

[12] S. M. Jafar Rizvi, Mutawarra Hussain, "Modeling Case Marking System of Urdu Language by using Semantic Information", In the Proceedings of IEEE International Conference on Natural Language Processing and Knowledge Engineering, NLP-KE 2005, Wuhan, China, 2005.

[13] S. M. Jafar Rizvi, Mutawarra Hussain, "Modeling Urdu Adjectives and Possession Marking using Head driven Phrase Structure Grammar," In the Proceedings of IEEE's International Multi topic Conference, INMIC 2005, FAST-NU, Karachi, 2005.

INDEX