Implementing GF Resource Grammar for Sindhi language

A subtitle that can be quite long if necessary

Master of science Thesis in Intelligent System Design

JHERNA DEVI OAD

Department of Applied Information Technology
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden, 2009
Report No. 2012:070
ISSN: 1651-4769
Implementing a Grammatical Framework Resource grammar for Sindhi

Abstract
Recently, demand for getting data through the world wide web has been dramatically increased. People with different languages can find information in languages which are varied with their own. But if we can manage somehow to translate information from one language to another language, then it might become possible to use the complete information. Grammatical Framework makes it easier to translate text from one language to another. Grammatical Framework is a functional programming language that can be used to develop multilingual natural language processing applications. Grammatical Framework programs implement the grammatical rules of natural language. The resource grammar library is an important part of the Grammatical Framework. The multilingual Grammatical Framework resource library contains a language-independent abstract syntax and a set of language-dependent concrete syntaxes.

The main focus of this thesis work is to develop a resource grammar of Sindhi language using Grammatical Framework. This work tries to cover in detail the morphology and the syntactic structure of Sindhi language. The developed Sindhi resource grammar can be used for multilingual applications, such as translation, language teaching and software localization.
Acknowledgement

I wish to express my sincere gratitude to Prof. Aarne Ranta for giving me this opportunity to work on such an interesting project. I am heartily thankful to my supervisor Shafqat Virk, whose encouragement, supervision and support from the preliminary to the concluding level enabled me to develop an understanding of the Grammatical Framework. Also my appreciation goes to Mutee-u-Rehman Assistant Professor at department of computer science Isra University Hyderabad Sindh Pakistan, his high quality research on Sindhi grammar has made an easy understanding and formalizing a resource grammar of Sindhi language in GF. I would also like to mention that my husband Jagdesh kumar masters student at RWTH Aachen Germany helped me to improve the quality of the thesis. Finally I would like to give a very special acknowledgement and gratitude to my father Ghansho Mal, my mother Sundri Bai, my siblings and my child, who encouraged and supported me to accomplish this goal.
Table of Contents

Abstract .................................................................................................................................. ii
Acknowledgement .................................................................................................................. iii
Chapter 1: Introduction ......................................................................................................... 1
  1.1 Linguistics ................................................................................................................... 1
  1.2 Grammatical Framework ................................................................................................. 1
    1.2.1 Resource Grammar ............................................................................................... 2
    1.2.2 Application Grammar ............................................................................................. 2
    1.2.3 Indo-Iranian Languages in GF Resource Grammar Library ..................................... 3
  1.3 Sindhi Linguistics .......................................................................................................... 4
    1.3.1 Sindhi Script ........................................................................................................... 4
Chapter 2: Sindhi Morphology ............................................................................................... 5
  2.1 Lexicon .......................................................................................................................... 5
    2.1.1 Nouns .................................................................................................................... 5
    2.1.2 Verbs ..................................................................................................................... 8
    2.1.3 Adjectives ............................................................................................................. 10
    2.1.4 Numerals ............................................................................................................. 11
    2.1.5 Adverbs ............................................................................................................... 12
    2.1.6 Pronouns ............................................................................................................. 13
    2.1.7 Conjunctions ....................................................................................................... 14
      2.1.7.1 Coordinate Conjunctions ............................................................................ 14
      2.1.7.2 Subordinate Conjunctions ......................................................................... 15
    2.1.8 Interjunctions ...................................................................................................... 15
Chapter 3: Syntax ................................................................................................................... 16
  3.1 Analytical Syntax .......................................................................................................... 16
    3.1.1 Noun Phrase (NP) ............................................................................................... 16
    3.1.2 Verb Phrase (VP) ................................................................................................. 18
    3.1.3 Adjectival Phrase (AP) ........................................................................................ 21
    3.1.4 Adverbs ............................................................................................................... 22
    3.1.5 Copula (Auxiliary Verb) ..................................................................................... 22
  3.2 Synthetic Syntax ............................................................................................................ 23
    3.2.1 Clauses ................................................................................................................ 23
    3.2.2 Sentences ............................................................................................................. 25
Chapter 4: Applications .......................................................................................................... 26
Chapter 5: Evaluations .......................................................................................................... 30
Chapter 6: Conclusion ........................................................................................................... 30
Chapter 7: Future Work ........................................................................................................ 31
References ............................................................................................................................. 32
Appendix ................................................................................................................................ 34
Chapter 1. Introduction

This thesis project implements Sindhi computational grammar in Grammatical Framework that can be used for text translation in multilingual grammars that has been already implemented in Grammatical Framework. There was not any particular translation tool that could provide the facility for text translation into Sindhi language. Grammatical Framework resource grammar for Sindhi is the one which provides text translation into Sindhi language.

1.1 Linguistics

In this modern world human wants to share and access the knowledge but due to language difference its not convenient to retrieve related information from a resource.

However, in order to understand once idea and thoughts, there is a need to translate one language to another language. It is very important to understand the grammar of both languages including morphology and syntax. Linguistically every language has an appropriate structure including Phonetics, Morphology and syntax. In the 1950, United States originated computational linguistics when computer started to translate Russian text into English [15]. After that computational linguistics started to grow rapidly by developing different applications that provided such facility.

It was very important to develop a central system or tool that could provide facility of multilingual translation. The google has provided such facility but it is limited and it does not cover most of the world languages. Grammatical Framework provides such opportunity and it makes easier to implement different language resource grammars.

1.2 Grammatical Framework

Grammatical Framework (GF, Ranta, 2004) is a functional and natural language processing programming language. GF is designed for writing grammars and GF programs are based on the Grammatical rules of the language. It is based on Martin-Löf’s type-theory (Martin-Löf, 1975) and implemented in Haskell.

It is a structure that manages and defines grammars [13]. Generally GF has a module system consisting of two main modules, one is an abstract syntax and another is concrete syntax. GF consists of a central data structure known as abstract syntax which describes the semantic concept. The concrete syntax that makes possible to describe how abstract syntax rules can be linearized in to different natural language. Grammatical Framework can parse and
translate text into multiple languages simultaneously [5]. GF allows to have more than one concrete syntax for a common abstract simultaneously and results into multilingual grammars. Logical representation of a grammar is an abstract syntax which is independent of any language. It defines a list of different lexical (i.e. Adjectives, verbs, nouns, adverbs, etc.) and syntactical (Adjective phrase, noun phrase, verb phrase adverb phrase, etc.) categories (cat). It also defines a set of tree structure rules (fun). The concrete syntax depends on natural languages. It allocates a linearization (lin) to each function and linearization type (lincat) to each category of the abstract syntax.

GF has a library known as Grammatical Framework resource library covering the morphology and syntax of natural languages. Resource library of Grammatical Framework is used for natural language processing. The Grammatical Framework libraries are of two types, Resource grammar and application grammar. At present GF contains 27 resource grammar libraries of different languages.

1.2.1 Resource Grammar
The Grammatical Framework resource grammar library consist of a set of natural language grammar which is implemented in Grammatical Framework. These grammars are built upon a common tree structure or a common abstract syntax. As a resource for language processing assignments (natural language interfaces, software localization, translation etc) the resource library can be used. The Resource Grammar Library is categorised roughly into two parts, morphological and syntactic. It defines the morphology and comprehensive part of the syntax of languages present in GF. As it has common abstract syntax which make convenient to implement another new language. The resource library provides an opportunity that allows non-linguist application programmers can write multilingual application programs easily [14].

1.2.2 Application Grammar
These grammars are domain specific grammars which encode semantic construction. The domain experts are supposed to write application grammars. Application grammar consists of an abstract syntax that defines the semantics of an application domain linguistically. The resource grammar has abstract syntax that defines the linguistic structure. The application grammar has a concrete syntax that can be expressed as a mapping to the abstract syntax. The application grammars encode the semantics of the particular domain by using resource grammar through abstract interfaces [3].
1.2.3 Indo-Iranian Languages in Grammatical Framework Resource Grammar Library

Grammatical Framework has been used to implement grammars of a number of natural languages. Indo-Iranian languages in Grammatical Framework resource grammar library includes Urdu, Punjabi, Sindhi. These languages can not use common code. Every language need to implement seperate code.

The implemented Sindhi resource grammar using Arabic script like Urdu and Punjabi. Urdu and Punjabi languages consist of 38 alphabets whereas Sindhi alphabet consists of 52 alphabet characters and two special characters. The direction of the writing of Sindhi, Urdu and panajbi is right-left.

Aforementioned languages have common scripts, but few similar alphabet characters. These languages have different grammars. All language has its own grammar rules.

Sindhi noun phrase has 11 case makers and noun phrase is very important part of syntax of the language. Coupla differentiate the tense of sentence, in Sindhi language sentences are incomplete without coupla. Urdu syntax using ezafe constructions like as Persian language. There is not any concept of ezafe construction in Sindhi language.

<table>
<thead>
<tr>
<th>Language</th>
<th>Script</th>
<th>IPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>Thank you</td>
<td></td>
</tr>
<tr>
<td>Sindhi</td>
<td>توہانی مهرباتی</td>
<td>tohaːniː mhrbaːniː</td>
</tr>
<tr>
<td>Urdu</td>
<td>پا شکریا</td>
<td>AːpK aː shukrīya</td>
</tr>
<tr>
<td>Punjabi</td>
<td>شکریا تہاا</td>
<td>tuhāːdā shukrīā</td>
</tr>
</tbody>
</table>

Table 1.2.3: Sindhi, Urdu and Punjabi script
1.3 Sindhi Linguistics

The main objective of this work is to implement GF resource grammar for Sindhi language. Sindhi falls under an Indo-Aryan Group of the Indo-Iranian Language family. Sindhi language is one of the South Asian regional languages, specially the language of Sindh province of Pakistan. Sindhi language is one of the most widely used language of almost 78 million speakers including 41 million speakers in Pakistan, 12 and 25 million respectively from India and other countries.

1.3.1 Sindhi Script

The Sindhi language is written in two scripts, Arabic and Devanagari script. Arabic script consists of 52 alphabet characters known Arabic-Sindhi script, standardized by British government in 1852. It is written from right to left, this script is used by people of Pakistan, where as Sindhi spoken in India has Devanagari script from left to right with additional four letters which express the special implosive sounds. Sindhi language can also be written with characters of the Gurmukhi as well as Gujrati alphabet.

The major characteristics of Sindhi are Rich morphology, verb-compounding, relatively free word-order, which inflects and derivates using suffixes and prefixes [8].

<table>
<thead>
<tr>
<th>پ</th>
<th>ہ</th>
<th>ض</th>
<th>ط</th>
<th>ئ</th>
<th>ۓ</th>
<th>ۓ</th>
<th>ن</th>
<th>ں</th>
<th>ہ</th>
<th>ا</th>
<th>ا</th>
</tr>
</thead>
<tbody>
<tr>
<td>ج</td>
<td>ہ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
</tr>
<tr>
<td>س</td>
<td>ہ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
</tr>
<tr>
<td>ہ</td>
<td>ہ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
</tr>
<tr>
<td>خ</td>
<td>ہ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
</tr>
<tr>
<td>ذ</td>
<td>ہ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
</tr>
<tr>
<td>ط</td>
<td>ہ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
</tr>
<tr>
<td>ز</td>
<td>ہ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
</tr>
<tr>
<td>ف</td>
<td>ہ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
</tr>
<tr>
<td>ض</td>
<td>ہ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
</tr>
<tr>
<td>ط</td>
<td>ہ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
</tr>
<tr>
<td>ئ</td>
<td>ہ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
<td>ۂ</td>
</tr>
</tbody>
</table>

Table 1.1: Sindhi Araic script and IPA (International Phonetic Alphabate)
Chapter 2. Sindhi Morphology

Sindhi language has a very rich morphology, inflection and derivations of the script treated by prefixes and suffixes. It has a large amount of morphological variants. There are primary and secondary words present in Sindhi language. The primary words are indivisible but the secondary words are divisible into compound (combination of two or more primary words) and complex words (addition of prefixes or suffixes) [8]. Morphologically Sindhi language uses reflexive (passive) and objective (impersonal) verbs trunks (Stems), also uses suffixes pronouns along with nouns, post-positions and verbs. [1]

2.1 Lexicon

A lexicon contains word entries and language rules. The words are stored in the lexicon according to the occurrence of parts of speech in Sindhi language. In the Sindhi lexicon the inflectional forms and meaning of a word is stored [12].

The Sindhi lexicon consists of two types of words, one containing simple words which has single meaningful unit and the other, covering complex words which can be divided in more than one meaningful units. A single meaningful unit is known as morpheme.

The words consist of more than one morpheme can be predominate in the language. These words have one root which constitutes core of the word and one or more units as modifiers known as affixes.

The GF resource grammar has a test lexicon. Test lexicon is built through a lexical function known as lexical paradigms; (Bringert et. al 2011). The paradigms are used to make an inflection table containing possible forms of the word according to lexical rules of the language. Test lexicon consists of 450 words with different lexical categories. These lexical categories are selected for testing randomly [9].

2.1.1 Nouns

Sindhi nouns may be simple or complex words. Nouns are classified Grammatically by number (singular and plural), case (Direct, oblique, ablative and vocative) and gender (masculine and feminine). In Sindhi language gender is classified in to two categories one is animate noun and an other is non-animate noun as in Urdu language. Animate nouns are specified with gender inherently where as non-animate nouns are not specified with gender inherently but it is specified artificially.
Table 2.1: Inflectional forms of noun ʰokəro_

<table>
<thead>
<tr>
<th>Cases /Masculine (Boy, ʰokəro)</th>
<th>Singular</th>
<th>Plural</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct (Dir)</td>
<td>ʰokəro</td>
<td>ʰok ra</td>
</tr>
<tr>
<td>Oblique (Obl)</td>
<td>ʰok re</td>
<td>ʰok rn</td>
</tr>
<tr>
<td>Vocative (Voc)</td>
<td>ʰok ra</td>
<td>ʰok ra</td>
</tr>
<tr>
<td>Ablative (Abl)</td>
<td>ʰok re</td>
<td>ʰok rn</td>
</tr>
</tbody>
</table>

GF stores different inflectional forms of noun and lexical details in a record-type structure. The construction of the noun in GF is:

Cat N;

lin Noun = {s : Number => Case => Str ; g : Gender};

It shows a record of a noun having two fields ‘s’ and ‘g’, where as ‘s’ explains ‘s : Number => Case => Str’ (a table from number to case to string). The entity s is a parameter to which number and case are the assigned values. The parameters number and case are given here:

Param number = sg | Pl;

Param Case = Dir | Obl | Voc | Abl;

As mentioned above that GF stores different inflectional forms of noun and lexical details in a record-type structure. So, ‘s’ is forming an inflection table and storing different forms of noun inflecting for gender, case and number, ‘g’ is the type of ‘Gender’ represents that the noun is feminine or masculine.

Parameter number inflects for singular (sg) and plural (Pl). Where as parameter case inflects for directive (Dir), oblique (Obl), ablative (Abl) and vocative (Voc).

As its been mentioned, there is a Grammatical function (fun) list and a categories (cat) list in an abstract syntax. The Grammatical functions (fun) declare words of lexical categories as constant. The linearization function (lin) and linearization type
(lincat) to each category are assigned by concrete syntax. As an example consider the following abstract syntax and concrete syntax code for implementation of the noun ‘boy’.

```plaintext
fun boy_N : N

lin boy_N = mkN01 "ڇڪڏڪ", "hokoro";
```

In GF the abstract syntax code (fun) defines that boy is of type ‘N’ (noun), while its linearization rule (lin) uses lexical paradigm mkN01 and builds all inflection forms of noun. In the above example ‘mkN01’ takes argument ‘ڇڪڏڪ’ (hokoro) and builds an inflection table. The definition of function mkN01 is given as:

```plaintext
Oper     mkN01: Str -> Noun;

Oper     mkN : (x1,_,_,_,_,_,_,x8 : Str) -> Gender -> Noun =
    \sd,so,sv,sa, pd,po,pv, pa, g -> {
        s = table {
            Sg => table { Dir => sd ; Obl => so ; Voc => sv ; Abl => sa
            } ;
            Pl => table { Dir => pd ; Obl => po ; Voc => pv ; Abl => pa
            }.
        } ;
        g = g
    };

mkN01 chokro = let chokr = (tk 1 chokro)
    in mkN (chokro) (chokr+"y") (chokr+"a") (chokr+"y")
    (chokr+"a") (chokr+"n") (chokr+"a") (chokr+"n")
    Masc ;
```

Here ‘chokro’ is as variable which takes a noun then creates its root and adds different information related to the noun to make its inflectional forms.

The above function builds record as mentioned bellow.
It is noted that in the record ‘S’ the inflectional forms of a noun ‘boy’ are stored which are inflecting for case, number and inherent gender. Since there are 14 lexical paradigms which cover all groups of Sindhi nouns, however here in this example only one lexical paradigm of noun is explained.

2.1.2 Verbs
Sindhi verbs have a complex structure. These verbs inflect for aspect (perfective, imperfective), mood (subjunctive, imperative), gender (Masculine, Feminine), tense (Present, Past, Future), number (Singular, Plural) and person (first, second casual, second respect, third near, third distance). Sindhi verbs have single class for all verbs, inflect alike but with minor difference. There are total sixty forms of the verb which have been implemented in GF Sindhi resource grammar library. The following structure builds table and stores different forms of verbs in GF.

cat verb;
lin verb = {s : VForm => Str} ;

Here ‘s’ is inflectional table storing all the inflectional forms of the verb from VForm to String. The table has only one field s. The parameter of the ‘VForm’ is defined as follows:

Param VerbForm = VF VTense PPerson Number Gender
                  | Inf | Inf_Fem | Inf_Obl | Ablative
                  | Root;

The parameter VerbForm has constructors, its use and description is defined as
VF stores different inflecting forms of the verb which inflect for tense, number, person and gender. The VTense, PPerson, number and gender are known as context parameters and they are explained as below.
param
    Gender = Masc | Fem;
    VTense = Subj | Perf | Imperf;
    PPerson = Pers1
        | Pers2_Casual
        | Pers2_Respect
        | Pers3_Near
        | Pers3_Distant;

The verb inflects for the gender (masculine and feminine) and VTense parameter covers the tenses of the Sindhi language. ‘Subj’, ‘Perf’ and ‘Imperf’ forms describe the subjunctive, perfective and imperfective aspects. The present, past and future tenses are quoted here as subjunctive, perfective and imperfective respectively. The verbs also inflect for the 1st person, 2nd person (Casual and Respect) and 3rd person (Near and at Distance).

fun live_V : V;
live_V = mkV "رھڻ", rhn;

Here lexical paradigm is mkV, which takes lemma “رھڻ” live_V and builds finite inflection table by analyzing the lemma. The different forms of the live_V are built by mkV, which are as follows:

\[
\begin{align*}
    \text{s .VF Subj Pers1 Sg.Masc => } & \text{رھڻ} \quad \text{-- rha:n} \\
    \text{s .VF.Imperf .Pers3_Distant .Sg .Masc => } & \text{رھڻدو} \quad \text{-- rhndo} \\
    \text{s .VF.Imperf .Pers3_Distant .Sg .Fem => } & \text{رھڻڈی} \quad \text{-- rhandi:} \\
    \text{s .Inf => } & \text{رھن} \quad \text{-- rhn} \\
    \text{s .Inf_Fem => } & \text{رھنی} \quad \text{-- rhi:} \\
    \text{s .Inf_Obl => } & \text{رھن} \quad \text{-- rh} \\
    \text{s .Ablative => } & \text{رھن} \quad \text{-- rhon} \\
    \text{s .Root => } & \text{رھن} \quad \text{-- rh}
\end{align*}
\]
2.1.3 Adjectives

Sindhi Adjectives inflect for gender, case and number. Sindhi adjectives are agreeing with the noun with respect to the gender, case and number. The inflectional categories of Sindhi adjectives are declinable and indeclinable. The declinable category adjectives inflect for the case and gender of the noun, whereas the indeclinable category of adjectives remain constant. The adjectives can be substituted as an adverb.

\[
\text{Adjective} = \{ s : \text{Number} \Rightarrow \text{Gender} \Rightarrow \text{Case} \Rightarrow \text{Str} \} ;
\]

‘s’ stores inflected forms of the adjective which inflect for the number (Singular, Plural), case (Direct, Oblique, Vocative, Ablative) and gender (Masculine, Feminine).

fun wide_A : A
lin wide_A = mkA "ویکرko"

Here mkA is a lexical paradigm defined as:

mkAdj : Str -> Adjective ;

Here mkAdj accepts a string and stores after analyzing that string. It calls another function adj which develops an inflectional table and stores all inflectional forms of an adjective. In Sindhi each adjective has sixteen inflectional forms.

\[
\begin{align*}
\text{adj} : (x_1, \ldots, x_{16} : \text{Str}) & \rightarrow \{ s : \text{Number} \Rightarrow \text{Gender} \Rightarrow \text{Case} \Rightarrow \text{Str} \} = \\
& \{ \\
& \quad s = \text{table} \{ \\
& \quad \quad \text{Sg} \Rightarrow (\text{cmnAdj} \ msd \ mso \ msv \ msa \ fsd \ fso \ fsv \ fsa).s ; \\
& \quad \quad \text{Pl} \Rightarrow (\text{cmnAdj} \ mpo \ mpv \ mpa \ fpd \ fpo \ fpv \ fpa).s \\
& \quad \} ; \\
& \text{cmnAdj} : (x_1, \ldots, x_8 : \text{Str}) \rightarrow \{ s : \text{Gender} \Rightarrow \text{Case} \Rightarrow \text{Str} \} = \\
& \quad \{ \\
& \quad \quad \text{s} = \text{table} \{ \\
& \quad \quad \quad \text{Masc} \Rightarrow \text{table} \{ \\
& \quad \quad \quad \quad \text{Dir} \Rightarrow \text{sd} ; \\
& \quad \} ; \\
& \end{align*}
\]
The following table is showing different forms of adjective wide.

<table>
<thead>
<tr>
<th>Case</th>
<th>Masculine</th>
<th>Feminine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dir</td>
<td>ویکرو</td>
<td>ویکریون</td>
</tr>
<tr>
<td>Obl</td>
<td>ویکری</td>
<td>ویکرین</td>
</tr>
<tr>
<td>Voc</td>
<td>ویکرا</td>
<td></td>
</tr>
<tr>
<td>Abl</td>
<td>ویکرا</td>
<td></td>
</tr>
</tbody>
</table>

2.1.4 Numerals

Sindhi numerals are written from left to right whereas script is right to left. These numerals possess a variety of numerals including Cardinal numbers and Ordinal numbers. The cardinal numbers describe the quantity and are used for counting in Sindhi whereas ordinal numbers refer to the ranking and ordering. For example:

**Cardinal Number:**

١, bli: _ cat
٤, kta: _ book

**Ordinal Number:**

٣, fi: _ three
٧, kta: _ dogs

سپارلیون گیالیندی آهی، ho _ she st _ seven boli: _ on _ languages گُن: a: lha: _ ژن: di: _ speak
<table>
<thead>
<tr>
<th>Cardinal Number</th>
<th>Sindhi</th>
<th>Ordinal Number</th>
<th>Sindhi</th>
</tr>
</thead>
<tbody>
<tr>
<td>One</td>
<td>هڪ،hk</td>
<td>First</td>
<td>پھربون</td>
</tr>
<tr>
<td>Two</td>
<td>بئ.طبع</td>
<td>Second</td>
<td>پبون،طبع</td>
</tr>
<tr>
<td>Three</td>
<td>تئ.ئى:</td>
<td>Third</td>
<td>پنئون،ئى:</td>
</tr>
<tr>
<td>Four</td>
<td>چار،ca:r</td>
<td>Forth</td>
<td>چوئن،co:nt</td>
</tr>
<tr>
<td>Five</td>
<td>پئن،pnj</td>
<td>Fifth</td>
<td>پئنjon</td>
</tr>
</tbody>
</table>

Table 2.2: Sindhi Numerals

### 2.1.5 Adverbs

The Sindhi has some original adverbs. They never inflect and their structure never changes. Sindhi adverbs are classified into five categories. These are temporal (indicate time of action), locational (indicates place of action), directional (indicates direction of action), manner (indicates manner of action) and degree (indicates degree of action) adverbs.

<table>
<thead>
<tr>
<th>Adverb Categories</th>
<th>Adverb</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Temporal</strong></td>
<td>کدھن،kdhn_when،هاتئ،ha:n_i:_Now،ئى:today</td>
</tr>
<tr>
<td><strong>Location</strong></td>
<td>هئى:Here،کئى:Where،ئى:o_near،پئى:Away</td>
</tr>
<tr>
<td><strong>Directional</strong></td>
<td>کئى:کئى:hnنWhere or in which direction</td>
</tr>
<tr>
<td><strong>Manner</strong></td>
<td>ئئى:ئئى:nIn this manner，like this</td>
</tr>
<tr>
<td><strong>Degree</strong></td>
<td>ئئى:ئئى:nMuch a lot،ئئى:ئئى:n.Loudly،ئئى:ئئى:n.Slowly</td>
</tr>
</tbody>
</table>

Table 2.3: Categories of Adverbs

GF uses following simple structure to store adverbs.

```plaintext
cat Adv ;
lincat Adv= \{} S : Str \} ;
```

These are constructed by simple functions. For Example

```plaintext
fun today_Adv : Adv ;
today_Adv = mkAdv "ئئى:today" ;
mkAdv : Str -> Adv = \str -> \{} s = str \} ;
```

Here the function mkAdv gets input string and stores it in the field ‘s’.
2.1.6 Pronouns

Pronouns are substitute of the nouns. These inflect for case and number. Few pronouns mark for the gender. Pronoun for person III is gender distinction.

<table>
<thead>
<tr>
<th>Type</th>
<th>Case/Masc</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Sg</td>
<td>Pl</td>
<td>Sg</td>
<td>Pl</td>
<td>Sg</td>
<td>Pl</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dir</td>
<td>Obl</td>
<td>Dir</td>
<td>Obl</td>
<td>Dir</td>
<td>Obl</td>
</tr>
<tr>
<td>Personal Person I</td>
<td>(I)</td>
<td>mon</td>
<td>,</td>
<td>mon</td>
<td>,</td>
<td>a:s:n</td>
<td>,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>mhn+jo</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(We)</td>
<td>a:s:n</td>
<td></td>
<td>a:s:n</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Personal Person II</td>
<td>(You)</td>
<td>ثن + جو</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ton</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(You)</td>
<td>toha:n</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>tohi:n</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(You)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.4: Pronoun for 1st person and 2nd person

Types of pronoun are personal, reflexive, relative, interrogative and demonstrative.

<table>
<thead>
<tr>
<th>Type</th>
<th>Masc</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Sg</td>
<td>Pl</td>
<td>Sg</td>
<td>Pl</td>
<td>Sg</td>
<td>Pl</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dir</td>
<td>Obl</td>
<td>Dir</td>
<td>Obl</td>
<td>Dir</td>
<td>Obl</td>
</tr>
<tr>
<td>Demonstrative</td>
<td>(he)</td>
<td>ا:ha:</td>
<td></td>
<td>ا:ha:</td>
<td></td>
<td>ا:ha:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>a:o</td>
<td></td>
<td>a:n</td>
<td></td>
<td>a:n</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>a: (: )</td>
<td></td>
<td>a: (: )</td>
<td></td>
<td>a: (: )</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.5: Pronoun for 3rd person
Table 2.5: pronoun for 3rd Person

In GF pronoun inflects for case oblique and direct. The pronoun ‘we’ is defined in GF as
we_Pron = personalPN "اسان" , a:sa:n "سین" , a:si:n ;

mkPron : (x1,x2 :Str) -> {s:Case => Str} =
\( y1,y2 \rightarrow \{ s = \text{table } \{ \text{Dir} \Rightarrow y1; \text{Obl} \Rightarrow y2; \} \);

2.1.7 Conjunctions
The single phrase , words of sentence , two or more sentences stand to one another, to express their relation conjunctions are used. These belongs to an un-inflected classes. Conjunctions are divided at syntactic level in to two main categories, coordinate conjunctions and subordinate conjunctions.

2.1.7.1 Coordinate conjunctions
They join two mutually independent sentences. The core coordinate conjunctions are:

<table>
<thead>
<tr>
<th>Coordinate conjunctions</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copulative</td>
<td>Also ِپ (pr)</td>
</tr>
<tr>
<td>Alternative</td>
<td>Or یا (i:a:), Neither نکر (nko)</td>
</tr>
<tr>
<td>Adversative</td>
<td>But ِپ (pr), Either توڑی (tori:)</td>
</tr>
</tbody>
</table>

Table 2.6: Coordinate Conjunctions

2.1.7.2 Subordinate conjunctions

They are used to construct a complex sentence by joining subordinate clauses. Some of the core Subordinate conjunctions are time, direction, location, reason, condition, manner etc.

<table>
<thead>
<tr>
<th>Subordinate conjunctions</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>When، جن (dhn), Before، جن (phri:n), Till، تائین (ta:ei:n)</td>
</tr>
<tr>
<td>Location</td>
<td>Where جتي (ti:)</td>
</tr>
<tr>
<td>Direction</td>
<td>Where جن (ji:da:nhn)</td>
</tr>
<tr>
<td>Manner</td>
<td>As جن (ji:en)</td>
</tr>
<tr>
<td>Reason</td>
<td>Therefore انرک (a:nkri:), Because چو ت (cho t)</td>
</tr>
</tbody>
</table>

Table 2.7: Subordinate Conjunctions

2.1.8 Interjection

A word or a phrase that describes emotions or sudden feelings is an interjection. It is always used in the vocative form. It does not have any relation with any other word in a sentence grammatically. Basically it is a sentence itself. It expresses surprise, joy and grief etc. Some interjections are as below.

Joy واه واه، oa:h oa:h _ Wao  
Grief افسوس، a:fsos _ sad  
Surprise مار، ma:r _ oo  

Chapter 3. Syntax
The morphology section describes different Sindhi lexical words and their different inflectional forms. In this section words combination and making words will be discussed. Sindhi language syntax is divided into two categories, analytical and synthetic.

3.1 Analytical Syntax

Analytical part explains how different parts of the speech join together according to their exact meaning, essential value and particular purpose. It includes noun phrase, verb phrase, adjective phrase and another close categories. The word order in neutral sentences in Sindhi is subject-object-verb (SOV) but, word order of sentence can be changed like SOV, SVO, OSV, VSO, etc. For example as shown below:

\[
\text{ori: boy, li: cat } \text{di: seen} \quad \text{SOV}
\]

\[
\text{li: cat, ori: boy } \text{di: seen} \quad \text{SVO}
\]

\[
\text{li: cat, ori: boy } \text{di: seen} \quad \text{OSV}
\]

3.1.1 Noun Phrase (NP)

Sindhi noun phrase can be a word or it can be a bunch of words which are related to each other grammatically and work together as a noun phrase. It has a head noun and complements or modifiers. Noun phrases can be interpreted without determiner and its structure contains modifiers and complements as Determiner-Modifier-Noun (hi: This, nei: new, kta:b_book), Possesor-Quantifies-Noun (mhnja: my, ca:r four, kta:b_book) [10].

Some noun phrases have an exact meaning on their own, like the Sun “ستح sji”. All proper nouns are noun phrases [11].

Here a noun phrase structure is described in GF. The noun phrase and its all forms are recorded into two fields ‘s’ and ‘a’.

\[
\text{Lincat NP : Type = \{ S : NPCase => Str ; a : Agr ; \}}
\]

Here ‘s’ is the inflectional table storing the record from NPCase to string (NPCase => Str) and ‘a’ is a feature of the noun phrase and it is used to select the appropriate form of other categories which matches with the noun.

\[
\text{Param NPCase = NPC Case | NPNom | NPVoc | NPIns1 | NPIns2} \\
| NPGen | NPDat | NPAcc | NPLoc1 | NPLoc2 | NPAbl ;
\]

\[
\text{param Case = Dir | Obl | Voc | Abl ;}
\]
Param Agr = Ag Gender Number Pperson;
Param Gender = Masc | Fem;
Param Pperson = Pers1 | Pers2_Casual | Pers2_Respect
               | Pers3_Near | Pers3_Distant;
Param Number = Sg | Pl

Here the constructor 'NPC Case' stores the cases of noun phrase.

NPC Case: It returns the cases of noun as in lexicon.

NPNom: Nominate case with the case creator 'ji: جي'.
NPVoc: Vocative case is used to address an inferior person with case creator 'a:o : او'.
NPIns1: Instrumental case is used to define an action performed by noun with case creator 'ka:n : کان'.
NPIns2: Instrumental case is used to define noun accompanied with any thing with case creator 'sa:n : سان'.
NPGen: Genitive case with suffix case creator 'ji: جي'.
NPDat: Dative Case is used to denote the postposition case creator 'ki: کي'.
NPAcc: Accusative case is used as nominative case or dative case with postposition case creator 'ki: کي'.
NPLoc1: Locative case1 is used to define place and time with case creator 'ti: تي'.
NPLoc2: Locative case2 is used to define place with case creator 'mi: تان, ma:n مان'.

Postpositions are placed after nouns to show their relationship with other constituents of the sentence. The lexical category of noun is changed first into an intermediate category common noun. Then common noun is converted into a noun phrase. A common noun 'CN' deals with noun and its modifiers. For example:

fun AdjCN : AP -> CN -> CN;
Lin AdjCN ap cn = {
    s = \n,c => ap.s ! n ! cn.g ! c ++ cn.s ! n ! c ;
    g = cn.g
} ;

The AdjCN linearization results a common noun for example 'grm_ئت ca:nh _تئا garam chai '. Here ' ca:nh چان' is common noun which is modified by the Adjective Phrase (AP) 'گرم گرم'.

17
As Sindhi adjective inflects for number, gender, case and degree so that the appropriate form of the adjective concatenate which agrees with the common noun. It is ensured by choosing a particular form of noun and an adjective from inflecting table, by using the selection operator (!). The Noun phrase can be obtained by converting common noun by using different functions.

Some functions which can be used to construct the noun phrase are as bellow:

- fun DetCN : Det -> CN -> NP ; -- the Woman
- fun UsePN : PN -> NP ; -- John
- fun UsePron : Pron -> NP ; -- She
- fun MassNP : CN -> NP ; -- (water)

The noun phrase can be built in different ways that are so common in other languages, as in the abstract syntax resource grammar but the linearization is defined in the concrete syntax because it is the language dependent.

### 3.1.2 Verb Phrase (VP)

Sindhi verb phrase is a syntactic form. It has a complex structure. The main verb is complex followed by an auxiliary verb which marks mood and tense (as object of the verb, adverb etc), these are used in development of other categories and clauses. A verb phrase structure is described in GF as:

```
VPH : Type = {
    s : VPHForm => {fin, inf : Str} ;
    subj : VType ;
    obj : {s : Str ; a : Agr} ;
    comp : Agr => Str;
    ad : Str;
    inf : Str;
    embComp : Str ;
    prog : Bool ;
} ;
```

The Verb phrase is recorded in the inflectional table ‘s’ having two strings `{fin, inf : Str}` storing all the values of Parameter VPHForm that is described as follows:
The VPHForm has six cases. The first constructor VPTense shows all the inflectional forms which are inflecting for the number, person, gender and tense (VPPTense) with the second constructor agreement feature (Agr). The third constructor (VPReq) shows a request form of the verb phrase.

The forth (VPImp) shows an imperative form of the VP. The fifth constructor (VPInf) contains the infinitive form. The sixth constructor VPStem contains a root form of the verb. These all constructors do not occur at the same time in Sindhi language.

The ‘inf’ stores an appropriate required form of the verb which agrees with the tense and the ‘fin’ stores the copula or auxiliary verb. The ‘obj’ stores an object of the verb and its information. The ‘subj’ stores transitivity of the verb by means of VType (Intransitive, Transitive or transitivePost):

Param VType = VIntrans | VTrans | VTransPost;

The ‘Comp’ stores the complement of verb which also inflects for the gender, number and person with Agr. The ‘ad’ stores the adverb.

The ‘embCom’ deals with exceptions of the word order in Sindhi language during the development of clauses and stores embedded complements. For example, if a sentence is a complement of a verb then it takes a different place in a clause; that is it places at the end of the clause.

<table>
<thead>
<tr>
<th>Subject</th>
<th>Auxiliary</th>
<th>Verb</th>
<th>Inflection</th>
</tr>
</thead>
</table>
| हन | हिंदी त मा न रान्द किया तो | श्रे | संक्य  त 
| हन | आही त मा न रान्द किया तो | श्रे | संक्य  त |

He says that I play game.

If an adverb is used as a complement of the verb then it is placed before the main verb.

<table>
<thead>
<tr>
<th>Subject</th>
<th>Auxiliary</th>
<th>Verb</th>
<th>Inflection</th>
</tr>
</thead>
<tbody>
<tr>
<td>हन</td>
<td>आही त मा न रान्द किया तो</td>
<td>श्रे</td>
<td>संक्य  त</td>
</tr>
</tbody>
</table>

She says that she runs fast.

Through the development of different functions, the lexical form of the verb is changed to the syntactic form (VP).

fun UseV : V → VP ;
lin UseV v= PredV v ;
PredV is a function which converts the lexical form (V) of verb to the syntactic form of verb phrase.

\[
\text{predV} : \text{Verb} \rightarrow \text{VPH} = \lambda v \rightarrow \{
\begin{align*}
\text{s} &= \lambda vh \rightarrow \text{case vh of } \{
\quad \text{VPTense VPPres (Ag g n p)} &\Rightarrow \{ \text{fin} = \text{copula CPresent n p g} ; \text{inf} = \text{verb.s} ! \text{VF Subj p n g} \} ; \\
\quad \text{VPTense VPPast (Ag g n p)} &\Rightarrow \{ \text{fin} = \text{copula CPast n p g} ; \text{inf} = \text{verb.s} ! \text{VF Perf p n g} \} ; \\
\quad \text{VPTense VPPerf (Ag g n p)} &\Rightarrow \{ \text{fin} = \text{copula CFuture n p g} ; \text{inf} = \text{verb.s} ! \text{VF Imperf p n g} \} ; \\
\quad \text{VPTense VFutr (Ag g n p)} &\Rightarrow \{ \text{fin} = \text{copula CFuture n p g} ; \text{inf} = \text{verb.s} ! \text{VF Imperf p n g} \} ; \\
\quad \text{VPStem} &\Rightarrow \{ \text{fin} = \{ \} ; \text{inf} = \text{verb.s} ! \text{Root} ; \text{cka g n} \} ; \\
\quad \text{VPImp} &\Rightarrow \{ \text{fin} = \text{verb.s}!\text{VF Subj Pers3_Near Pl Masc} ; \text{inf} = \text{verb.s} ! \text{Root} \} ; \\
\quad \text{VPReq} &\Rightarrow \{ \text{fin} = \{ \} ; \text{inf} = \text{verb.s}!\text{VF Subj Pers1 Pl Masc} \} ; \\
\_ &\Rightarrow \{ \text{fin} = \{ \} ; \text{inf} = \text{verb.s} ! \text{Root} \} ; \\
\text{obj} &= \{ \text{s} = \{ \} ; \text{a} = \text{defaultAgr} \} ; \\
\text{subj} &= \text{VIntrans} ; \\
\text{inf} &= \text{verb.s} ! \text{Inf} ; \\
\text{ad} &= \{ \} ; \\
\text{embComp} &= \{ \} ; \\
\text{prog} &= \text{False} ; \\
\text{comp} &= \lambda \_ \Rightarrow \{ \} \\
\}
\end{align*}
\]

The verb has three lexical forms. These forms are used to make tenses (VPPres, VCPPast, VPPerf, VFutr, VPPer). All these forms cover different combinations of the tenses, mood and aspect of Sindhi.

The above code builds an inflectional table ‘s’ for the present tense at verb phrase level ‘VPPres’ of all possible forms of the number, gender and person(Ag p g n). In the above code, the lexical imperfective form (VF Imperf n g p) of verb (V) is used to make present tense. As described above ‘inf’ stores the main verb form and ‘fin’ stores the corresponding auxiliary verb. In the above code all other parts initialize with empty or default values. These parts improve the verb phrase with another component e.g. an adverb complement etc.
Want to eat, khaen: \textit{\textunderscore eat} chahen: \textit{\textunderscore want}

\textbf{Compl VV : } VV \rightarrow VP \rightarrow VP ;

Sit here, hity\textit{\textunderscore here} Weh: \textit{\textunderscore sit}

\textbf{AdvVP : } VP \rightarrow Adv \rightarrow VP

### 3.1.3 Adjectival Phrase (AP)

The Sindhi adjectives inflect for the number case and gender at morphological level. At the syntax level adjectival phrase agree with the noun. In GF adjectival phrases are described as bellow.

\textbf{fun} PositA : A -> AP ; -- (cold, thado)

The linearization of the adjectival phrase is a very simple and resembles with an adjective.

\textbf{lin} positA a = a ;

The other way to develop ‘AP’ is as following:

\textbf{fun} ComparA : A -> NP -> AP ;

A comparative adjectival phrase is developed from a noun phrase and an adjective. The linearization of the above function in GF is as follows:

\textbf{ComparA a np = } \{ 
  s = \langle n,g,c \rangle \Rightarrow np.s \! NPC Obl ++ "k'an" ++ a.s \! n \! g \! c ;
  s = \langle n,g,c \rangle \Rightarrow np.s \! NPC Abl ++ "k'an" ++ a.s \! n \! g \! c ;
\} ;

Other examples of adjectival phrase functions are as shown below:

\textbf{As sweet as I} , \textit{\textunderscore etro\_ As mitho\_ sweet jetro\_ As man\_ 1}

\textbf{fun} CAdvAP : CAdv -> AP -> NP -> AP ; -- as cool as John

Older then I , \textit{\textunderscore mun\_ 1 kan\_ then porho\_ old}

\textbf{fun} ComparA : A -> NP -> AP ;

### 3.1.4 Adverbs

Sindhi adverbs never inflect so their construction is very simple. At syntactical and lexical level there are different close classes and function constructions. For example, the construction of an adverb is as bellow:

\textbf{PositAdvAdj : A -> Adv ;}

\textit{\textunderscore Warmly, garam:}
3.1.5 Copula (Auxiliary verb)

A copula modifies the main verb in a clause to form a particular tense. It defines a link between the subject of a sentence and a predicate (an adverbial or a subject complement).

Sindhi copulas inflect for the tense, number, gender and person.

In GF auxiliary verb is implemented for subjective, perfective, imperfective and continuous tense.

```haskell
param  CTense = CPresent | CPast | CFuture | CContinuous;
oper
copula : CTense -> Number -> PPerson -> Gender -> Str = \t,n,p,g ->
case <t,n,p,g> of {
  <CPresent,Sg,Pers1,Masc > => "ت"; -- t
  <CPresent,Sg,Pers1,Fem > => "ت"; -- t
  <CPresent,Sg,Pers2_Casual,Masc > => "ت"; -- t
  <CPast,Pl,Pers2_Casual, _ > => "ئا "; -- hea: 
  <CPast,Pl,Pers2_Respect, _ > => "ئا "; -- hi:o
  ........
  <CPast,Pl,Pers3_Near,Masc > => "ئا "; -- hea: 
  <CPast,Pl,Pers3_Near,Fem > => "ئا "; -- hei:on
  <CPast,Pl,Pers3_Distant,Masc > => "ئا "; -- hea: 
  <CPast,Pl,Pers3_Distant,Fem > => "ئا "; -- hi:on
}
```

Here \( (t,n,p,g) \) are the variables which store different inflectional forms of the copula.

Here \( 't' \) describes the tense, \( 'n' \) defines the number which is either singular or plural, \( PPerson \) defines the 1st person or 2nd person or 3rd person, and \( 'g' \) is used to define gender of the copula.

In GF, eighty inflectional forms of the copulas are implemented. Those copulas are very useful to make meaningful sentences.
3.2 Synthetical Syntax

The synthetical syntax describes how parts of the speech join together to form a single clause or sentence, two clauses or sentences or more than two. [10]

A Sindhi phrase is head-final. Generally, the verb, post-position, and noun with complements and modifiers precede their heads.

I do not reject the word of mother and father.

3.2.1 Clauses

Clauses are included in the synthetical part of the syntax. As Sindhi phrase is a lexical word or combination of the lexical words and these words are related to each other grammatically. A clause is a single phrase or combination of the phrases which are linked together. A clause can be formed by joining the different types of the phrases including noun phrase, verb phrase, adjective phrase, etc. A clause may not have any subject sometimes or it may have a predicate and a subject of its own. These clauses are used to form a sentence. There is a difference between a sentence and a clause in the Grammatical Framework resource grammar API tense system. That is a clause has inconsistent tense whereas a sentence has fixed tense.

For the construction of a sentence in GF, clauses are developed first and then clause tenses which are used to make sentences.

```
linecat Clause : Type = {s : VPHTense => Polarity => Order => Str} ;
```

Where:

```
Param VPHTense = VPGenPres | VPImpPast | VPFut
                 | VPContPres | VPContPast | VPContFut
                 | VPPperfPres | VPPperfPast | VPPperfFut
                 | VPPperfPresCont | VPPperfPastCont | VPPperfFutCont
                 | VPSsubj ;
```

```
Param Order = ODir | OQuest ;
Param Polarity = POS | Neg ;
```
fun PredVP : NP -> VP -> Cl ; -- Rana walks

Its linearization is:

lin predVP np vp = mlClause np vp ;

The GF resource library does not cover the whole tense system of the Sindhi, however that is structured to cover more or less the aspect and mood of the tense system. It covers only eight combinations along with present tense, past tense, future tense and conditional.

To achieve more and more coverage of the Sindhi tense system, twelve tense (VPHTense) are developed based on sentences at clause level. The parameter Order is used to develop questions and direct clauses whereas parameter polarity is used to develop negative and positive clauses. The SOV agreement of the analytical syntax is accessible by using predVP.

The SOV agreement is a feature of the Sindhi and its behaviour is ergative which is covered in noun phrase (NP).

where

oper mkClause : NP -> VPH -> Clause = \np,vp -> { s = \vt,b,ord =>

Subj agr : NPCase * Agr = case vt of {

   VPImpPast => case vp.subj of {
      VTrans => <NPErg, vp.obj.a> ;
   VTransPost => <NPErg, defaultAgr> ;
   _ => <NPC Dir, np.a>
   } ;
   _ => <NPC Dir, np.a>
}

quest =

   case ord of { ODir => []; OQuest => "c'a" };

in

quest ++ np.s ! subj ++ vp.obj.s ++ vp.ad ++ vp.comp ! np.a ++ vps.inf ++ n++ vps.fin ++
vp.embComp
}

The above code takes a noun and verb phrase and then construct a clause having a polarity, order and tense and that is fixed in sentences by using different functions. The noun phrase selects an appropriate form of the verb phrase.
fun UseCl : Temp → pol → Cl → S ;

Here, the syntactic category is Temp, it is used to store a record of tense and anteriority. In Temp category, the tense refers to the abstract level tense and it is used for Sindhi tense by selecting the particular clause.

### 3.2.2 Sentences

Sentences cover an important and main part of synthetical syntax. Sindhi sentences are divided in two major categories: simple sentence and compound sentence.

Simple sentences must contain a subject and a predicate. The Subject may be defined by adjective or subjective or numeral or pronoun. The predicate may be defined by an adjective or verb or subjective or numeral with the copula.

The compound sentence is formed when two or more than two sentences are joined together. It is done either by the way of subordination or coordination.

Coordination sentences are formed by joining two or more sentences and each sentence remains an independent of the other. It is formed without or by the copulative, disjunction and by the negative adverb (Neither-Nor -نہ-).

The Subordinate sentences are formed when two or more sentences may be joined in such a way that one of the sentence depends on an other sentence where as the other sentence should be an independent.

As a clause have an inconsistent tense, order and polarity so through different functions the parameters like order and polarity can be fixed which results into declarative sentence.

In GF, generation of sentence construction is as follow:

```plaintext
UseCl  temp p cl =
    { s = case <temp.t,temp.a> of {
        <Pres,Simul> => temp.s ++ p.s ++ cl.s ! VPGenPres ! p.p ! ODir;
        <Pres,Anter> => temp.s ++ p.s ++ cl.s ! VPPerfPres ! p.p ! ODir;
        <Past,Simul> => temp.s ++ p.s ++ cl.s ! VPImpPast ! p.p ! ODir;
        <Past,Anter> => temp.s ++ p.s ++ cl.s ! VPPerfPast ! p.p ! ODir;
        <Fut,Simul> => temp.s ++ p.s ++ cl.s ! VPFut ! p.p ! ODir;
        <Fut,Anter> => temp.s ++ p.s ++ cl.s ! VPPerfFut ! p.p ! ODir;
        <Cond,Simul> => temp.s ++ p.s ++ cl.s ! VPSubj ! p.p ! ODir;
        <Cond,Anter> => temp.s ++ p.s ++ cl.s ! VPSubj ! p.p ! ODir;
    };
    };
```
The following function is one of the most important functions:

```
fun UseCl : Temp -> Pol -> Cl -> S ;
```

Here ‘Temp’ parameter is combination of the parameters for tense and for interiority. The function ‘UseCl’ accepts input as polarity, anteriority, tense and a clause and it gives output as a sentence. The different types of the sentences those are relative sentences and interrogative sentences are developed through the function as follows:

```
fun UseRCl : Temp -> Pol -> RCl -> RS ;
fun UseQCl : Temp -> Pol -> QCl -> QS ;
```

Chapter 4. Applications

The reported resource grammar can be used as a resource for language processing applications as natural language interfaces, translation system, multilingual generations and spoken dialogue systems.

The minibar online is web application that allows to translate text into implemented all languages. There are menu for selecting the output and the input language to translate. The abstract shows the sentence parse tree and its linearization in Sindhi. Some snapshots of Sindhi to English and English to Sindhi translations are shown bellow.
Abstract: ADEI UseCl (TAnz TPres ASimply) PPos (PredVP (PredCN this Det (UseN beer_N))) (CompAP (PostA good_A)))

Send

deny this beer is good
Chapter 5. Evaluations

Testing a resource grammar is different from testing natural language processing applications in general, where testing is done against some text corpus. Testing resource grammars is much like testing software libraries in general (Ranata 2009b). For checking the accuracy, the Sindhi resource grammar used to observe and translate. The accuracy of Sindhi resource grammar 97% achieved, because implemented resource grammar does not cover all the aspects of the language.

The noun phrase cases formed by noun and ergative cases are not yet implemented fully. It might be possible that reported grammar could not translate correctly.

Chapter 6. Conclusion

Grammatical Framework resource grammar for Sindhi is the only translation platform which provides the Grammatical translation results. Before this implementation, there was not any particular tool that could provide the facility for text translation into Sindhi language. Sindhi grammar library present in Grammatical Framework has used different categories and functions to manage the morphology and syntax implementation. The library has 44 categories and 190 functions. This grammar implementation covers a major part of the Sindhi language. As GF has an abstract syntax that is supporting a group of different languages, so it is very difficult to cover each and every aspect of the language in the library of resource grammar. The grammar of Sindhi language does not cover all aspects of language because limited lexicon word list is not enough to cover the whole language. There is a need to implement full Sindhi grammar and its all aspects that can translate the text more accurately. There is also a need to extend the resource grammar library boundaries so that full grammar of any language can be implemented. It is possible to increase the efficiency and effectiveness of the Sindhi grammar by implementations of extra features of language.
Chapter 7. Future Work

As mentioned already this resource grammar does not cover all the features of the Sindhi. In the future these uncovered features and aspects can be implemented and developed. As lexicon is not a full vocabulary of Sindhi language, one can increase lexicon vocabulary list as well.

The grammar resource can be extended by implementing a specific module for specific features of Sindhi language. It can not be accessed through the common API of GF, but it can be accessed in a particular language application grammar.

In general, the assessment of the resource grammar is an assessment of the software library. It is very different from the natural language processing assessment, where the assessment is generally done on some text croups. It is possible to develop a user friendly interface that can be easily accessed by native speakers and they can take benefit from this development.
References


2. http://linguistics.ucsc.edu/about/what-is-linguistics.html


Appendix

Lang: PositA warm_A
LangEng: warm
LangSnd: گرم

Lang: ComparA warm_A (UsePron i_Pron)
LangEng: warmer than I
LangSnd: مان کان گرم

Lang: SentAP (PositA good_A) (EmbedS (UseCl (TTAnt TPres ASimul) PPos (PredVP (UsePron she_Pron) (UseComp (CompAdv here_Adv))))))
LangEng: good that she is here
LangSnd: سنوت ہو ئی ہہی اہی

Lang: AdAP very_AdA (PositA warm_A)
LangEng: very warm
LangSnd: تمام گرم

Lang: PositAdvAdj warm_A
LangEng: warmly
LangSnd: گرم

Lang: PrepNP in_Prep (DetCN (DetQuant DefArt NumSg) (UseN house_N))
LangEng: in the house
LangSnd: گھر

Lang: ComparAdvAdj more_CAdv warm_A (UsePN john_PN)
LangEng: more warmly than John
LangSnd: جان گھنتا و ذیکا گرم

Lang: ComparAdvAdjS more_CAdv warm_A (UseCl (TTAnt TPres ASimul) PPos (PredVP (UsePron he_Pron) (UseV run_V))))
LangEng: more warmly than he runs
LangSnd: گھنتا و ذیکا گرم ہو دوزی تو
Lang: SubjS when_Subj (UseCl (TTAnt TPres ASimul) PPos (PredVP (UsePron she_Pron) (UseV sleep_V)))
LangEng: when she sleeps
LangSnd: جَذَّهِنَّ هِیٰ ۳ سَهِیٰ تُیٰ

Lang: AdNum (AdnCAdv more_CAdv) (NumNumeral (num (pot2as3 (pot1as2 (pot0as1 (pot0 n5)))))
LangEng: more than five
LangSnd: سَان گَر ۳۳ ۶ی پنج

Lang: ConjS and_Conj (BaseS (UseCl (TTAnt TPres ASimul) PPos (PredVP (UsePron he_Pron) (UseV walk_V))) (UseCl (TTAnt TPres ASimul) PPos (PredVP (UsePron she_Pron) (UseV run_V))))
LangEng: he walks and she runs
LangSnd: هُر هِنِیٰہُ ۳ُ دَرْزِیٰ تُیٰ

Lang: ConjAP and_Conj (BaseAP (PositA cold_A) (PositA warm_A))
LangEng: cold and warm
LangSnd: گِرم،گرم

Lang: ConjNP or_Conj (BaseNP (UsePron she_Pron) (UsePron we_Pron))
LangEng: she or we
LangSnd: ۶ی ۶ یا اسَان

Lang: ConjAdv or_Conj (BaseAdv here_Adv there_Adv)
LangEng: here or there
LangSnd: ۶ی ۶ یا اتِیٰ

Lang: ConjS either7or_DConj (BaseS (UseCl (TTAnt TPres ASimul) PPos (PredVP (UsePron he_Pron) (UseV walk_V))) (UseCl (TTAnt TPres ASimul) PPos (PredVP (UsePron she_Pron) (UseV run_V))))
LangEng: either he walks or she runs
LangSnd: کُکَرُ ۶ی بُدُورُ هُیٰ ۶ی لُوا ۶ی ۶ دَرْزِیٰ لَیٰ

Lang: ConjAP both7and_DConj (BaseAP (PositA warm_A) (PositA cold_A))
LangEng: both warm and cold
LangSnd: ب گرم ہ و

Lang: ConjNP either7or_DConj (BaseNP (UsePron he_Pron) (UsePron she_Pron))
LangEng: either he or she
LangSnd: گر ہ یا ہر ہ یاو و

Lang: ConjAdv both7and_DConj (BaseAdv here_Adv there_Adv)
LangEng: both here and there
LangSnd: ب ہ اتی

Lang: ImpersCl (UseComp (CompAP (PositA hot_A)))
LangEng: it is hot
LangSnd: کوسو اہی

Lang: GenericCl (UseV sleep_V)
LangEng: one sleeps
LangSnd: گرو سیمہ یو

Lang: CleftNP (UsePron i_Pron) (UseRCl (TTAnt TPast ASimul) PPos (RelVP IdRP (ComplSlash (SlashV2a do_V2) (UsePron it_Pron))))
LangEng: it is I who did it
LangSnd: مان اہیان جہیں اہ سکروں هم

Lang: CleftAdv here_Adv (UseCl (TTAnt TPast ASimul) PPos (PredVP (UsePron she_Pron) (UseV sleep_V)))
LangEng: it is here that she slept
LangSnd: یو سیمہ یو یا

Lang: ExistNP (DetCN (DetQuant IndefArt NumSg) (UseN house_N))
LangEng: there is a house
LangSnd: گر ای

Lang: ExistIP (IdetCN (IdetQuant which_IQuant NumPl) (UseN house_N))
LangEng: which houses are there
LangSnd: یہاں گھروں ہیں

Lang: PredVP (UsePron i_Pron) (ProgrVP (UseV sleep_V))
LangEng: I am sleeping
LangSnd: مان سمنان بیا تو

Lang: DetCN (DetQuant DefArt NumSg) (UseN man_N)
LangEng: the man
LangSnd: مان

Lang: UsePN john_PN
LangEng: John
LangSnd: جان

Lang: UsePron he_Pron
LangEng: he
LangSnd: وہ

Lang: PredetNP only_Predet (DetCN (DetQuant DefArt NumSg) (UseN man_N))
LangEng: only the man
LangSnd: صرف مان

Lang: PPartNP (DetCN (DetQuant DefArt NumSg) (UseN man_N)) see_V2
LangEng: the man seen
LangSnd: زسیل مان

Lang: DetNP (DetQuant this_Quant (NumCard (NumNumerals (num (pot2as3 (pot1as2 (pot0as1 (pot0 n5))))))))
LangEng: these five
LangSnd: ہی پنج

Lang: DetCN (DetQuantOrd this_Quant (NumCard (NumNumerals (num (pot2as3 (pot1as2 (pot0as1 (pot0 n5))))))) (OrdSuperl good_A)) (UseN man_N)
LangEng: these five best men
LangSnd: هی پنج سئز مائیور

Lang: DetCN (DetQuant this_Quant (NumCard (NumNumeral (num (pot2as3 (pot1as2 (pot0as1 (pot0 n5))))))) (UseN man_N)
LangEng: these five men
LangSnd: هی پنج مائیور

Lang: DetCN (DetQuant this_Quant NumPl) (UseN man_N)
LangEng: these men
LangSnd: مائیور

Lang: DetCN (DetQuant this_Quant NumSg) (UseN man_N)
LangEng: this man
LangSnd: مائیور

Lang: NumCard (NumNumeral (num (pot2as3 (pot1as2 (pot0as1 (pot0 n5)))))
LangEng: five
LangSnd: پنج

Lang: NumCard (NumDigits (IIDig D_5 (IDig D_1)))
LangEng: 5 1
LangSnd: ۱ & + ۵

Lang: NumCard (NumNumeral (num (pot2as3 (pot1as2 (pot1plus n5 pot01)))))
LangEng: fifty - one
LangSnd: اچکری و

Lang: NumCard (AdNum almost_AdN (NumDigits (IIDig D_5 (IDig D_1))))
LangEng: almost 5 1
LangSnd: ۱ & + ۵
ghecht_kari

Lang: OrdDigits (IIDig D_5 (IDig D_1))
LangEng: 5 1st
LangSnd: ۱ & + ۵
Lang: OrdNumeral (num (pot2as3 (pot1as2 (pot1plus n5 pot01))))
LangEng: fifty - first
LangSnd: ایک گنگاہ &+ ون

Lang: OrdSuperl warm_A
LangEng: warmest
LangSnd: گرم

Lang: DetCN (DetQuantOrd DefArt (NumCard (NumNumeral (num (pot2as3 (pot1as2 (pot0as1 (pot0 n5))))))) (OrdSuperl good_A)) (UseN man_N)
LangEng: the five best men
LangSnd: پنج سہیر مائہر

Lang: DetCN (DetQuant DefArt (NumCard (NumNumeral (num (pot2as3 (pot1as2 (pot0as1 (pot0 n5)))))))) (UseN man_N)
LangEng: the five men
LangSnd: پنج مائہر

Lang: DetCN (DetQuant IndefArt (NumCard (NumNumeral (num (pot2as3 (pot1as2 (pot0as1 pot01))))))) (UseN man_N)
LangEng: one man
LangSnd: ہیک مائہر

Lang: DetCN (DetQuant DefArt (NumCard (NumNumeral (num (pot2as3 (pot1as2 (pot0as1 pot01))))))) (UseN man_N)
LangEng: the one man
LangSnd: ہیک مائہر

Lang: DetCN (DetQuant DefArt NumSg) (UseN man_N)
LangEng: the man
LangSnd: مائہر

Lang: DetCN (DetQuant DefArt NumPl) (UseN man_N)
LangEng: the men
beer

my house

house

big house

house which John bought

house on the hill

question where she sleeps
Lang: DetCN (DetQuant (PossPron i_Pron) NumSg) (ApposCN (UseN friend_N) (UsePN john_PN)))
LangEng: my friend John
LangSnd: دوست جان

Lang: num (pot2as3 (pot1as2 (pot0as1 (pot0 n6)))))
LangEng: six
LangSnd: چھ

Lang: num (pot2as3 (pot1as2 (pot0as1 pot01)))
LangEng: one
LangSnd: ہک

Lang: num (pot2as3 (pot1as2 (pot1 n6)))
LangEng: sixty
LangSnd: سنھ

Lang: num (pot2as3 (pot1as2 pot110))
LangEng: ten
LangSnd: ڈھ

Lang: num (pot2as3 (pot1as2 pot111))
LangEng: eleven
LangSnd: بارھنھ

Lang: num (pot2as3 (pot1as2 (pot1to19 n6)))
LangEng: sixteen
LangSnd: سورھنھ

Lang: num (pot2as3 (pot1as2 (pot1 n6)))
LangEng: sixty
LangSnd: سنھ

Lang: num (pot2as3 (pot1as2 (pot1plus n6 (pot0 n5))))
LangEng: sixty-five
LangSnd: پنج ھٺ

Lang: num (pot2as3 (pot2 (pot0 n4)))
LangEng: four hundred
LangSnd: چار سو

Lang: num (pot2as3 (pot2plus (pot0 n4) (pot1plus n6 (pot0 n7))))
LangEng: four hundred and sixty-seven
LangSnd: چار سو ست ھٺ

Lang: num (pot3 (pot2plus (pot0 n4) (pot1plus n6 (pot0 n7))))
LangEng: four hundred and sixty-seven thousand
LangSnd: ک ستمہت هزار ا چار

Lang: IDig D_8
LangEng: 8
LangSnd: ۸

Lang: PhrUtt NoPConj (UttS (UseCl (TTAnt TPres ASimul) PPos (PredVP (UsePN john_PN) (UseV walk_V)))) NoVoc
LangEng: John walks
LangSnd: جان لی و

Lang: UttQS (UseQCl (TTAnt TPres ASimul) PPos (QuestCl (PredVP (UsePron it_Pron) (UseComp (CompAP (PositA good_A))))))
LangEng: is it good
LangSnd: ایا اها سنو اهمی

Lang: UttImpPol PNeg (ImpVP (UseV sleep_V))
LangEng: don't sleep
LangSnd: ن اسمه

Lang: UttIP whoPl_IP
who

why

this man

here

to sleep

, my friend

does John walk

who walks
why does John walk

where is John

which five songs

which five

with whom

where is it

who is it

Lang: QuestIAdv why_IAdv (PredVP (UsePN john_PN) (UseV walk_V))
LangEng: why does John walk
LangSnd: چو جان هی تو

Lang: QuestIComp (CompIAdv where_IAdv) (UsePN john_PN)
LangEng: where is John
LangSnd: جان کیا ایں

Lang: IdetCN (IdetQuant which_IQuant (NumCard (NumNumeral (num (pot2as3 (pot1as2 (pot0as1 (pot0 n5)))))))) (UseN song_N)
LangEng: which five songs
LangSnd: جینگ کنگ گانا

Lang: IdetIP (IdetQuant which_IQuant NumSg)
LangEng: which
LangSnd: جینگ

Lang: PrepIP with_Prep whoSg_IP
LangEng: with whom
LangSnd: سان گنگ

Lang: QuestIComp (CompIAdv where_IAdv) (UsePron it_Pron)
LangEng: where is it
LangSnd: ایا کنگ ایئر

Lang: QuestIComp (CompIP whoSg_IP) (UsePron it_Pron)
LangEng: who is it
LangSnd: ایا کنج ایئر
Lang: ExistNP (DetCN (DetQuant IndefArt NumSg) (RelCN (UseN woman_N) (UseRCl (TTAnt TPres ASimul) PPos (RelCl (PredVP (UsePN john_PN) (ComplSlash (SlashV2a love_V2) (UsePron she_Pron)))))))
LangEng: there is a woman such that John loves her
LangSnd: هناك امرأة سان عشق يوج أن

Lang: ExistNP (DetCN (DetQuant IndefArt NumSg) (RelCN (UseN woman_N) (UseRCl (TTAnt TPres ASimul) PPos (RelVP IdRP (ComplSlash (SlashV2a love_V2) (UsePN john_PN))))))
LangEng: there is a woman who loves John
LangSnd: هناك امرأة سان عشق يوج

Lang: ExistNP (DetCN (DetQuant IndefArt NumSg) (RelCN (UseN woman_N) (UseRCl (TTAnt TPres ASimul) PPos (RelSlash IdRP (SlashVP (UsePN john_PN) (SlashV2a love_V2))))))
LangEng: there is a woman whom John loves
LangSnd: هناك امرأة جان سان عشق يوج

Lang: PredVP (UsePN john_PN) (UseV walk_V)
LangEng: John walks
LangSnd: يوج تا

Lang: RelCN (UseN girl_N) (UseRCl (TTAnt TPres ASimul) PPos (RelSlash IdRP (SlashPrep (PredVP (UsePron he_Pron) (UseV walk_V)) with_Prep)))
LangEng: girl with whom he walks
LangSnd: کر یو جوین سان هی تا

Lang: ImpVP (ReflVP (SlashV2a love_V2))
LangEng: love yourself
LangSnd: عشق کر سان

Lang: UseCl (TTAnt TCond AAnter) PNeg (PredVP (UsePN john_PN) (UseV walk_V))
John wouldn't have walked

girl who wouldn't have walked

girl with whom I wouldn't have walked

she sleeps, which is good

John walks.

are they here?

LangEng: John wouldn't have walked
LangSnd: 

Lang: UseQCl (TTAnt TCond AAnter) PNeg (QuestCl (PredVP (UsePN john_PN) (UseV walk_V)))
LangEng: wouldn't John have walked
LangSnd: 

Lang: RelCN (UseN girl_N) (UseRCl (TTAnt TCond AAnter) PNeg (RelVP IdRP (UseV walk_V)))
LangEng: girl who wouldn't have walked
LangSnd: 

Lang: RelCN (UseN girl_N) (UseRCl (TTAnt TCond AAnter) PNeg (RelSlash IdRP (SlashPrep (PredVP (UsePron i_Pron) (UseV walk_V)) with_Prep)))
LangEng: girl with whom I wouldn't have walked
LangSnd: 

Lang: RelS (UseCl (TTAnt TPres ASimul) PPos (PredVP (UsePron she_Pron) (UseV sleep_V)) (UseRCl (TTAnt TPres ASimul) PPos (RelVP IdRP (UseComp (CompAP (PositA good_A))))))
LangEng: she sleeps, which is good
LangSnd: 

Lang: TFullStop (PhrUtt NoPConj (UttS (UseCl (TTAnt TPres ASimul) PPos (PredVP (UsePN john_PN) (UseV walk_V)))) NoVoc) TEmpty
LangEng: John walks.
LangSnd: 

Lang: TQuestMark (PhrUtt NoPConj (UttQS (UseQCl (TTAnt TPres ASimul) PPos (QuestCl (PredVP (UsePron they_Pron) (UseComp (CompAdv here_Adv)))))) NoVoc) TEmpty
LangEng: are they here?
LangSnd: 

46
Lang: PredVP (UsePron i_Pron) (UseV sleep_V)
LangEng: I sleep
LangSnd: ہاں ہے ہے

Lang: PredVP (UsePron i_Pron) (ComplVV want_VV (UseV run_V))
LangEng: I want to run
LangSnd: ہاں ہے ہے

Lang: PredVP (UsePron i_Pron) (ComplVS say_VS (UseCl (TTAnt TPres ASimul) PPos (PredVP (UsePron she_Pron) (UseV run_V))))
LangEng: I say that she runs
LangSnd: ہاں ہے ہے

Lang: PredVP (UsePron i_Pron) (ComplVQ wonder_VQ (UseQCl (TTAnt TPres ASimul) PPos (QuestVP whoSg_IP (UseV run_V))))
LangEng: I wonder who runs
LangSnd: ہاں ہے ہے

Lang: PredVP (UsePron they_Pron) (ComplVA become_VA (PositA red_A))
LangEng: they become red
LangSnd: اہی ہے ہے

Lang: PredVP (UsePron i_Pron) (ComplSlash (SlashV2S answer_V2S (UseCl (TTAnt TPres ASimul) PPos (PredVP (UsePron it_Pron) (UseComp (CompAP (PositA good_A)))))) (UsePron he_Pron))
LangEng: I answer to him that it is good
LangSnd: ہاں ہے ہے

Lang: PredVP (UsePron i_Pron) (ComplSlash (SlashV2A paint_V2A (PositA red_A)) (UsePron it_Pron))
LangEng: I paint it red
LangSnd: ہاں ہے ہے
car which I want to buy

he loves himself

this is warm

we are loved

we sleep here

we always sleep

we are small

I am a man
I am here

Paint the earth both smaller than the sun and bigger than the moon

every baby is either a boy or a girl

either from here, there or everywhere

bird which I want to paint red

want to buy it
In the Sindhi Resource Grammar test some sentences which do not give fully correct translation.

Lang: UttImpSg PNeg (ImpVP (ReflVP (SlashV2a love_V2)))
LangEng: don't love yourself
LangSnd: ن عشق ر پا سان

Lang: UttImpPl PNeg (ImpVP (ReflVP (SlashV2a love_V2)))
LangEng: don't love yourselves
LangSnd: ن عشق ر پا سان

Lang: QuestSlash whoSg_IP (SlashVP (UsePN john_PN) (SlashV2a love_V2))
LangEng: whom does John love
LangSnd: یک چراغ عشق کری تو
Lang: PredVP (UsePron i_Pron) (ComplSlash (Slash3V3 give_V3 (UsePron he_Pron))
(UsePron it_Pron))
LangEng: I give it to him
LangSnd: مان ہنہ ہو ڈیان تو

Lang: RelCN (UseN girl_N) (UseRCl (TTAnt TPres ASimul) PPos (RelSlash IdRP (SlashVP
(UsePron he_Pron) (SlashV2a see_V2)))))
LangEng: girl whom he sees
LangSnd: چہو چرگی جیھن ہو ڈیسی تو

Lang: RelCN (UseN girl_N) (UseRCl (TTAnt TPres ASimul) PPos (RelSlash IdRP
(AdvSlash (SlashVP (UsePron he_Pron) (SlashV2a see_V2)) today_Adv))
LangEng: girl whom he sees today
LangSnd: چہو چرگی جیھن اج ہو ڈیسی تو

Lang: RelCN (UseN girl_N) (UseRCl (TTAnt TPres ASimul) PPos (RelSlash IdRP (SlashVS
(UsePron she_Pron) say_VS (UseSlash (TTAnt TPres ASimul) PPos (SlashVP (UsePron
he_Pron) (SlashV2a love_V2))))))
LangEng: girl whom she says that he loves
LangSnd: چہو چرگی جیھن سنہ ہو چہو چرگی تو

Lang: RelCN (UseN girl_N) (UseRCl (TTAnt TPres ASimul) PPos (RelSlash IdRP (SlashVS
(UsePron she_Pron) say_VS (UseSlash (TTAnt TPres ASimul) PPos (SlashVP (UsePron
he_Pron) (SlashV2a love_V2))))))
LangEng: girl whom she says that he loves
LangSnd: چہو چرگی جیھن سنہ ہو چہو چرگی تو