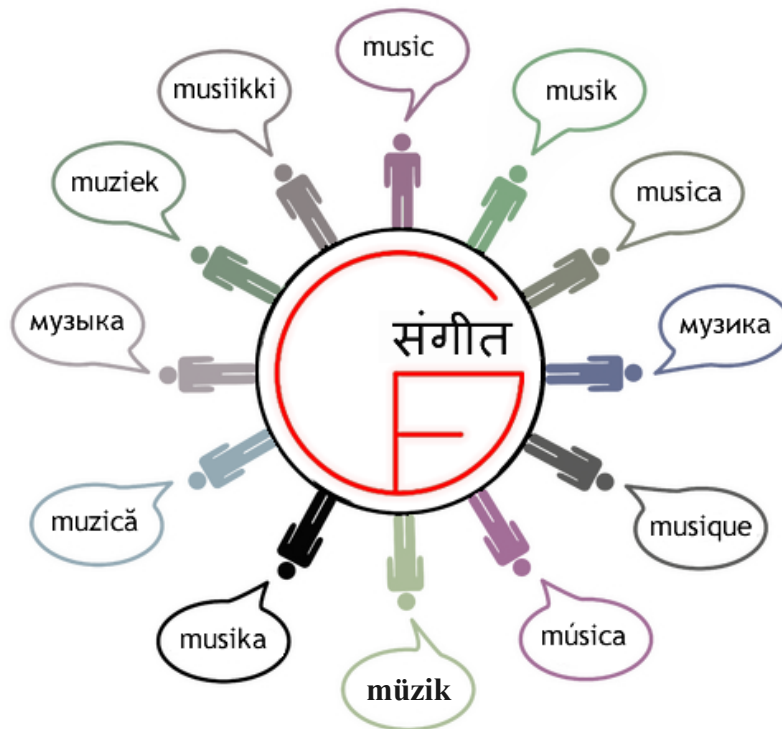


CHALMERS



Implementing the GF Resource Grammar for Nepali Language

Master of Science Thesis in Software Engineering and Technology

DINESH SIMKHADA

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
Göteborg, Sweden, June 2012

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

Implementing GF Resource Grammar for Nepali language

DINESH, SIMKHADA

© DINESH SIMKHADA, June 2012.

Examiner: AARNE RANTA, Prof.

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Cover: concept showing translation of Nepali word संगीत (music) to different languages that are available in Grammatical Framework. Inspired from GF summer school poster and stock images

Department of Computer Science and Engineering
Göteborg, Sweden June 2012

Abstract

The Resource Grammar Library is a very important part of Grammatical Framework. The resource grammar mainly contains morphological and syntactic modules and implements a common abstract syntax. This work focuses on writing a complete implementation of Nepali computational grammar in Grammatical Framework. It tries to cover in-depth details of morphological and syntactic structures of the Nepali language and the way we formalize these grammatical details in the abstract structure that the Grammatical Framework provides. The current implementation can be used for multilingual applications just like any other language that is present in Grammatical Framework. The grammar covers as much details as possible on both morphological and syntactic level, so claiming this implementation to be one of the most comprehensive computational grammar for Nepali language won't be wrong. The correctness of this work has been verified by feedback from native Nepali speakers about the correctness of the examples in the GF resource grammar API.

Table of Contents

1. Introduction.....	6
1.1 Motivation.....	7
1.2 Related work.....	7
1.3 Grammatical Framework.....	7
1.4 The GF Grammar Example.....	8
1.5 Nepali Language.....	10
1.6 Nepali Writing System.....	10
1.7 Lexical form of Nepali Language.....	12
2. Morphology.....	13
2.1 Nouns.....	13
2.2 Morphological inflection of Nepali nouns in GF.....	15
2.3 Verbs.....	17
2.3.1 Morphological inflection and Structure in GF.....	19
2.4 Adjective.....	26
2.4.1 Functional of adjectives.....	27
2.4.2 Some of the characteristics of Nepali adjectives.....	27
2.4.3 Implementation of adjective in GF.....	28
2.5 Adverbs.....	29
2.6 Pronouns.....	29
2.7 Conjunctions.....	30
2.7.1 Coordinating conjunctions.....	30
2.7.2 Subordinating conjunctions.....	30
2.8 Numerals.....	31
3. Syntax.....	33
3.1 Noun-Phrases(NP).....	33
3.1.1 Common-noun phrase(CNP).....	33
3.1.2 Common nouns as head.....	33
3.1.3 Determiners in the common-noun phrase.....	34
3.1.3.1 Demonstratives.....	35
3.1.3.2 Quantifiers.....	35
3.1.2 Limiters.....	35
3.1.4 Modifiers in the common-noun phrase(CNP).....	35
3.2 Proper-noun phrase.....	36
3.2.1 Person names.....	36
3.2.2 Place names.....	37
3.3 Pronoun Phrase.....	37

3.3.1	Pronouns as heads.....	37
3.3.2	Personal pronoun phrase modifiers.....	40
3.4	Adjectival Structures.....	40
3.4.2	Quantifiers in the Adjp.....	42
3.4.2.3	Superlative quantifier phrases.....	43
3.5	Modifiers within common noun phrases.....	43
3.6	Adverbial Structures.....	44
3.6.1	Simple adverbs.....	44
3.6.1.1	Adverbs ending in -ari/अरि 'doing'.....	44
3.6.1.2	Adverbs ending in -sath/ साथ 'with'.....	45
3.6.1.3	Adverbs ending in -purvaka/पूर्वक.....	45
3.6.2	Compound adverbs.....	46
3.7	Conjunctions.....	46
3.7.1	Coordinating conjunctions.....	47
3.7.2	Subordinating conjunctions.....	48
3.8	Verb Phrases.....	48
3.8.1	Verbs as heads.....	50
3.8.2	Auxiliary verbs in the verb phrase.....	50
3.8.3	The negative verb forms.....	50
3.8.3.1	The negative suffix na-/न-.....	51
3.8.4	Modifiers in the verb phrase.....	52
3.8.5	Transitive verbs and their complements.....	52
3.8.6	The equational verbs.....	53
3.8.6.1	Identificational hunu/हुनु case.....	53
3.8.7	Transitive verbs and complements.....	54
3.9	Clauses.....	54
3.9.1	Imperative clauses.....	55
3.9.2	Question clauses.....	55
3.10	Sentential Structure.....	56
3.10.1	Clauses as segmental constituents.....	57
4.	Evaluation and Results.....	58
5.	Further work.....	58
6.	Conclusion.....	59

7. Important Terms.....60

8. Appendix.....61

9. References.....62

Acknowledgement

I would like to thank everyone who helped me directly or indirectly to achieve this goal. I would like to express my special thanks to Prof. Aarne Ranta for giving me this opportunity to work on such an interesting project, and for all the help and support during the project. I would also like to thank Shafqat Virk for his support and assistance during the entire project. He helped me to understand many internal and implementation details of the Grammatical Framework. Also my appreciation goes to author Jayaraj Acharaya, his high quality of research on Nepali grammar has made it easy to understand and formalize Nepali language in Grammatical Framework, finally I would like to thank all my friends and family for their support to accomplish this goal.

List of Abbreviations

- AB - Ablative case
- AC - Accusative case
- Adjp - Adjective phrases
- CNP - Common-noun phrase
- DT - Dative case
- GN - Genitive case
- GF - Grammatical Framework
- HGH - High grade Honorific
- IN - Instrumental case
- LC - Locative case
- LGH - Low grade Honorific
- MGH - Mid grade Honorific
- Mod - modifier
- NP - Noun phrase
- NM - Nominative case
- SC - Subject complement
- TV1 - Transitive verb 1
- TV2 - Transitive verb 2
- TV3 - Transitive verb 3
- TV4 - Transitive verb 4
- VP - Verb Phrase

Chapter 1

1. Introduction

Natural Language Processing(NLP) has become one of the most practical and useful fields of computer science. When we look at the world wide web or off-line print media there are enormous amounts of useful information that is freely available, but often the language becomes barrier for them to be explored. Learning new language to access each piece of information written in another language is impossible task since we live in a world where we have more than 6000 spoken languages, of which 4500 have more than 1000 speakers[8]. A natural approach to solve this problem is to build a reliable multilingual system that can easily help to translate the content from one language to another. Grammatical Framework(GF, Ranta, 2004) is one of the efforts to build a reliable multilingual translation system, where translation works in a very controlled environment so that we can trust output of the system.

The main aim of this project is to implement Nepali language as a resource grammar for Grammatical Framework, to achieve this goal we need to study every grammatical constructs of the Nepali language and formalize the Nepali grammar in such a way that we can express it in computational form.

This chapter explains the motivation behind this thesis, and also covers a brief introduction to grammatical framework and Nepali language. Chapter 2 covers the morphology of Nepali language and its implementation in Grammatical Framework. Third chapter explains the syntax part of the language and implementation details, as expected implementation of syntax part is complex compared to the morphological part. Morphology part describes the word structure of the language on the other hand syntax part covers the principles and rules for constructing sentences. Later chapters covers the possible future work, conclusion of the thesis work is given in the final chapter which is followed by appendix and references.

1.1 Motivation

This work is motivated by building a reliable and most comprehensive computational grammar for Nepali language. Grammatical Framework is very suitable system to implement and test computational grammar for natural languages, so Nepali resource grammar has been build on top of the Grammatical Framework. With the Nepali resource grammar implemented on Grammatical Framework it is possible to translate Nepali texts to languages that is already implemented in Grammatical Framework and vice versa. Since the popular commercial product like Google translate doesn't have support for Nepali language it is good to invest time on the open-source tools like GF so that everyone can benefit from it. This is also one of the motivation behind this work.

Work for this thesis is divided in to two parts, first is the study of Nepali grammar and formalizing the grammar in computational form so that we can implement all the abstract syntax provided by the GF, of course this formalization is useful to implement Nepali computation grammar in any other system. The second part is the implementation of Nepali formalized grammar and standard lexicons into GF framework so that we will get the complete and reliable GF resource grammar for Nepali language.

1.2 Related work

There have been few previous efforts of writing a translation system for Nepali language. DoBhase¹ is one of the effort of wring translation system, it mainly focused on building Nepali to English translation system. The project has been conducted under the supervision of Kathmandu University and is running since 2005. Madan Puraskar Pustakalaya² had made some efforts of researching the translation system for Nepali language but no information about the implementation details has been made available in it's home page. We believe this thesis will serve as a valuable resource in future for similar related works.

1.3 Grammatical Framework

Grammatical Framework (GF) is a domain specific or a special propose language designed for defining the grammar of natural languages. GF is much inspired by functional and logical languages, it heavily depends on the notion of abstract and concrete syntax for the definition of grammar. GF grammars consists of an abstract syntax and at least one concrete syntax. Abstract syntax is declarative and it defines the scope of the grammar, abstract syntax gives an abstract over view of all the structures that can be build from it, on the other hand concrete

1 <http://nlp.ku.edu.np/lpdobhase/en/index.php>

2 <http://www.mpp.org.np/index.php>

syntax takes the definition of abstract syntax and implement them for particular natural language. The concrete implementation heavily uses inflection tables to get different forms of a word to maintain contract between different part of sentence. In the core level GF has a common abstract syntax and every language have to implement this abstract syntax to get full concrete syntax of that language. This helps to achieve the multilingual grammar, as translation from one language to another is trivial because of the common contract between all languages.

GF provides features like parsing (converting concrete to abstract syntax) and linearization (converting abstract syntax to concrete syntax), it has separate rules for abstract and concrete syntax[3]. It is a strongly typed system and supports multilingual grammars. GF provides resource grammar for different languages as library, all these resource grammar are based on the common abstract syntax provided with GF platform. GF currently contains resource grammar for 27 languages.

1.4 The GF Grammar Example

To get the better idea of GF grammar let's take a very small grammar that describes the quality of different types of books. The abstract syntax has the terms like category **cat** and function **fun**, a function can take different categories to produce a required result, for example in the example below Pred takes Item and Quality and produces Phrase, Quality can either be Interesting, Romantic, Boring or Educational similarly we have other functions to get the required result.

```
abstract Book = {
    flags starcat = Phrase;
    cat
        Phrase; Item, Kind, Quality ;
    fun
        Pred : Item -> Quality -> Phrase ;
        This, That : Kind -> Item ;
        Mod : Quality -> Kind -> Kind ;
        Novel, Course, Children : Kind ;
        Very : Quality -> Quality ;
        Interesting, Romantic, Boring, Educational : Quality ;
}
```

Figure 1.1: Abstract syntax of Book grammar.

The code below shows the English concrete syntax of Book grammar. *lincat* is the linearization type, for every category in abstract syntax we must have *lincat* type defined. Here every category has the String as linearization type, that means the final output of all categories will be string.

```
concrete BookEng of Book {
  lincat
    Phrase, Item, Kind, Quality = {s : Str};
  lin
    Pred item quality = {s = item.s ++ "is" ++ quality.s};
    This kind = {s = "this" ++ kind.s};
    That kind = {s = "that" ++ kind.s};
    Mod quality kind = {s = quality.s ++ kind.s};
    Novel = {s = "novel"};
    Course = {s = "course book"};
    Children = {s = "children's book"};
    Interesting = {s = "interesting"};
    Romantic = {s = "romantic"};
    Educational = {s = "educational"};
}
```

Figure 1.2: Concrete syntax for English language

While writing concrete syntax for a new language we need to follow the structure of abstract syntax and grammar of the implementing language. The contract between different parts of the sentence, gender, tense, number etc. are the basic things that comes into consideration while implementing concrete syntax of a natural language. The concrete syntax of Nepali for the abstract syntax above looks as in the code below, we can clearly see that there is some grammatical difference between the concrete syntax of English and Nepali.

```
concrete BookNep of Book {
  lincat
    Phrase, Item, Kind, Quality = {s : Str};
  lin
    Pred item quality = {s = item.s ++ quality.s ++ "छ्"};
```

```

This kind = {s = "यो" ++ kind.s} ;
That kind = {s = "त्यो" ++ kind.s}
Mod quality kind = {s = quality.s ++ kind.s} ;
Novel = {s = "उपन्यास"} ;
Course = {s = "पाठ्य पुस्तक"} ;
Children = {s = "वाल्‍य पुस्तक"} ;
Interesting = {s = "रोमान्‍चक"} ;
Romantic = {s = "रोम्यान्‍टिक"} ;
Educational = {s = "शीक्षामुलक"} ;
}

```

Figure 1.3: Concrete syntax for Nepali language.

1.5 Nepali Language

Nepali language belongs in the Indo-Aryan branch of the Indo-European family, mother tongue of around 60 percent of the Nepal's total population. It is the official language of Nepal and is also spoken in Bhutan, parts of India and Burma. Nepali is the medium of uniform communication in public administration, educational system, TV, newspaper etc. It was known with different names in different point of times, like Khas bhasa, Lokabhasa, Gorkhali, and later the name Nepali came from the name of the country itself. It is a Sanskrit derived language and some of the dialectal variation of Nepali follows the caste system of Hinduism.

1.6 Nepali Writing System

Nepali is written in Devanagari script, it is written from left to right and Devanagari alphabet has no Capital or Small letters[1]. Following vowels and consonant alphabets are found in classical Nepali grammar,

Vowels	अ, आ, इ, ई, उ, ऊ, ऋ, ए, ऐ, ओ, औ, अं, अः
Consonants	क, ख, ग, घ, ङ, च, छ, ज, झ, ञ, ट, ठ, ड, ढ, ण, त, थ, द, ध, न, प, फ, ब, भ, म, य, र, ल, व, स, ष, श, ह, क्ष, त्र, ज्ञ

Table 1.1: Devanagari alphabets

Vowels	अ H, आ A, इ i:, ई I:, उ f, ऊ F, ऋ Z, ए e:, ऐ E:, ओ O औ W, अं HM, अः Hh:
Consonants	क k, ख K, ग g, घ G, ङ n:, च C, छ c, ज j, झ J, ञ Y, ट q, ठ Q, ड x, ढ X, ण N, त t, थ T, द d, ध D, न n, प p, फ P, ब b, भ B, म M, य y, र r, ल L, व v, श S, ष z, स s, ह h, क्ष kx:z, त्र tx:r, ज्ञ jx:Y

Table 1.2: The Devanagari alphabets with Transliteration we used in GF program.

The Devanagari form of vowels have two forms 1) free form which are written the single vowel constitutes the syllabus, which are displayed in the table above 2) conjunct forms which are written when vowels are preceded by consonants[1]

Vowels	अ	आ	इ	ई	उ	ऊ	ऋ	ए	ऐ	ओ	औ
Conjunct Forms		ा	ि	ी	ु	ू	ृ	े	ै	ो	ौ

The vowel sound अ doesn't have corresponding conjunct form, which means it's presence is indicated by nothing but the shape of the bare consonant symbol(ः)[1]. When these conjunct form of vowels are joined with consonant symbols they form the following CV syllabus.

क् + अ	= क
क् + आ	= का
क् + इ	= कि
क् + ई	= की
क् + उ	= कु
क् + ऊ	= कू
क् + ऋ	= कृ
क् + ए	= के
क् + ऐ	= कै
क् + ओ	= को
क् + औ	= कौ

Additional symbols:

Apart from the things mentioned above following symbols are used in the Devanagari writing system

Bindu/Chandarabindu	ँ
Ansuvar/Cirbindu	ं

Visarga	:
Purna Viram (Full stop mark)	

Table 1.3: Additional Symbols

1.7 Lexical form of Nepali Language

Nepali grammar consists of both inflected and uninflected forms which are open and the closed form of classes, traditionally known as the parts of speech. The open classes are those classes whose membership is in principle indefinite or unlimited and closed form of classes are those whose membership is fixed or limited where the new items are not regularly added. Noun, adjective, verb and adverb belongs to the open class whereas the pronoun, coordinating conjunction, subordinating conjunction, postpositions, interjection, vocative and nuance particle belongs to the closed class.[1]

Chapter 2

2. Morphology

This chapter covers the morphology of Nepali grammar and their implementation.

2.1 Nouns

Nouns in Nepali inflects for singular or plural numbers and for seven different cases (nominative, accusative, instrumental, dative, ablative, genitive and locative)[1] as listed in the table below. It doesn't inflect for gender (masculine or feminine) but belong directly to a determined or undetermined gender class[1]. If we look at the traditional grammar it shows nouns has a inflected form for gender, number and seven cases eg. choro 'son' vs. chori 'daughter', but modern grammar prefers to treat such form as separate lexical items, independent of each other.

Nepali noun shows inflectional contrasts for singular vs. plural eg. manis/मानिस 'man' , manisharu/मानिसहरु 'men' and for seven different cases. The case-number suffixes are shown in the table below.

Cases	Singular	Plural
Nominative (Nm)	-	-haru / हरु
Accusative (Ac)	-lai / लाई	-haur-lai / हरुलाई
Instrumental (In)	-le / ले	-haur-le / हरुले
Dative (Dt)	-lai / लाई	-haur-lai / हरुलाई
Ablative (Ab)	-bata / बाट	-haru-bata / हरुबाट
Genitive (Gn)	-ko / को	-haru-ko / हरुको
Locative (Lc)	-ma / मा	-haru-ma / हरुमा

Table 2.1: The number and case suffixes of nouns

The inflection table of noun kitab/किताब 'book' looks like as shown below

Cases	Singular	Plural
Nominative (Nm)	किताब	किताबहरु
Accusative (Ac)	किताबलाई	किताबहरुलाई
Instrumental (In)	किताबले	किताबहरुले
Dative (Dt)	किताबलाई	किताबहरुलाई
Ablative (Ab)	किताबबाट	किताबहरुबाट
Genitive (Gn)	किताबको	किताबहरुको
Locative (Lc)	किताबमा	किताबहरुमा

Table 2.2: The number and case inflection of noun Book/किताब

Noun in GF defined as

```

N = { s : Number => Case => Str ;
      g : Gender ;
      t : NType ;
      h : NPerson
    };

```

where

Number = Sg | Pl ;

Case = Nom | Acc | Ins | Dat | Abl | Loc ;

Gender = Masc | Fem ;

NType = Living | NonLiving ;

NPerson = Pers1 | Pers2_L | Pers2_M | Pers2_H | Pers3_L | Pers3_M | Pers3_H ;s

It shows noun is a record of four fields 's', 'g', 't' and 'h', 's' which is defined as 's : Number => Case => Str' is read as 'a table from Number to Case to String'. It is an inflection table that stores different inflectional forms of a noun inflecting for number and case. In a table type structure the entity on the left hand side of the symbol '=' is a parameter and on the right hand side is a term assigning a value to the parameter on left. 'g' is the gender of noun, which holds the information whether the noun is Masculine or Feminine, 't' is used to define the type of noun, it stores information about the animacy of a noun like living or nonliving and also in some cases it is used to indicate professions and occupations. This information is useful at

syntax level in different grammatical constructions. One such example is the construction of a noun-phrase from a determiner and a common noun e.g.

charjana manis haru/चारजना मानिसहरु 'four men'

charwata lathiharu/ चारवटा लठीहरु 'four sticks'

here in the case of men (human case) jana/जना is added and in the case of sticks (non-human case) wata/वटा is added after the determiner.

'h' defines the different honorific levels for nouns. This information is useful in the subject-verb agreement at syntax level, to illustrate that we can take the following example.

Johan kitab padchha / जोन किताव पढछ 'Johan reads the book'

Buba kitab padhnu hunchha / बुवा किताव पढनुहुन्छ 'Father reads the book'

In these constructions, verb “reads” agrees with the honorific level of its subjects “Johan” and “Father” respectively. In Nepali the noun, the honorific level of father is higher (High grade honorific) and this information is very essential for correct grammatical constructions at syntax level.

2.2 Morphological inflection of Nepali nouns in GF

The section above discussed about the different parameters that noun takes, here we will discuss about the construction of noun, we use a special functions called lexical paradigms to construct different lexical categories. These paradigms take canonical form of a word and build the corresponding lexical category according to the lexical structure that has been already defined . To illustrate the point lets take an example of a noun 'man' as $man_N : N$; this is the abstract definition of noun man and every language implementing this function should provide its concrete definition, for Nepali we defined as follows,

$man_N = regN$ "manchey" human ;

where $regN$ is a overloaded function to satisfy construction of different types of nouns. For the construction of noun man it will call the function $regN$,

$regN : Str \rightarrow NType \rightarrow N = \backslash s, t \rightarrow mkNReg\ s\ t\ Pers3_L$;

To handle the different types of noun we have defined different functions of noun separately so that users don't have to provide all the details to the system. Femenine nouns are constructed using function $mkNF$, all the nouns defined using this function takes the feminine gender by default. To handle uncountable nouns we have defined a function $mkNUC$, this

nouns have same singular and plural forms, similarly *mkPN* is used to construct proper nouns, *mkCmpdNoun* is used to construct compound nouns and all the other regular nouns are handled by *regN*, these nouns takes masculine as default gender. In Nepali unless explicatively defined noun is treated in masculine form.

All these functions calls a common function *mkN*, this function builds a inflection table of different morphological forms of noun including different cases as we discussed above.

```
mkN : (x1,_,_,_,_,_,_,_,_,_,_,x12 : Str) -> Gender -> NType -> NPerson -> Noun =
  \sn, sa, si, sd, sab, sl, pn, pa, pi, pd, pab, pl, g, t, h -> {
    s = table {
      Sg => table {
        Nom => sn ; Acc => sa ; Ins => si ;
        Dat => sd ; Abl => sab ; Loc => sl
      };
      Pl => table {
        Nom => pn ; Acc => pa ; Ins => pi ;
        Dat => pd ; Abl => pab ; Loc => pl
      }
    };
    g = g ; t = t ; h = h
  };
```

Based on above definition the GF table of noun man, man_N = regN "मान्छे" looks like, as follows.

```
{s = table Number [table Case ["मान्छे"; "मान्छेलाई"; "मान्छेले";
                               "मान्छेलाई"; "मान्छेबाट"; "मान्छेमा"];
  table Case ["मान्छेहरु"; "मान्छेहरुलाई"; "मान्छेहरुले";
              "मान्छेहरुलाई"; "मान्छेहरुबाट"; "मान्छेहरुमा"]];
  g = Masc; h = Pers3_L; t = Living; lock_N : {} = <>
}
```

Figure 4: Inflection table representation in GF for noun Man/मान्छे.

As we can see, inflection table creates all the possible variations of word and retains other informations like gender, honorific form and its type (i.e. living or non living)

2.3 Verbs

Nepali verb is one of the complex structure in grammar, it inflects for person (first, second and third), number (singular and plural), gender of the subject in third person singular and tenses. Verbs also inflect to show contrasts of the gender of honorifics in second and third persons, similarly it also shows inflection for infinitive, perfective participle, imperfective participle, conjunctive participle and absolutive participle forms[1][2][7]. There are three levels of honorifics and the difference in gender is also marked in low grade honorific forms, all these measures makes the verb's inflectional system fairly complicated.

Verbs function as the head of the clause structure. As head of the clause structure, verbs either stand alone or in construction with various types of complements . We can classify verbs in to transitive, equational and intransitive types. Transitive are the type of verbs which take direct objects as complements, whereas equational verbs take subject complements as dependents and intransitive verbs are marked by the absence of either direct object or subject complements.

As verbs shows inflection for person, number, gender, tense, mood and aspect[1], because of all these parameters a single verb can show more than 400 forms. We have used the following definition to represent our verb including all required parameters.

VerbForm =

```
VF VTense Aspect Polarity NPerson Number Gender
| Root – Root form
| Inf – Infinitive form
| ProgRoot Aspect Number Gender
| PVForm -- Partial verb form 'khan' is the PVForm of 'khanu'
| Imp ;
```

where

VTense = NPresent | NPast PTypes | NFuture FTypes;

PTypes = Simpl | Hab ;

FTypes = Defin | NDefin ;

Aspect = Perf | Imperf ;

NPerson = Pers1 | Pers2_L | Pers2_M | Pers2_H | Pers3_L | Pers3_M | Pers3_H ;
Polarity = Pos | Neg ;
Gender = Masc | Fem ;

The structure of *VerbFrom* looks a bit complex due to the fact that inflection of verb depends on many parameters and some of the data-types like *Root*, *Inf*, *Imp* holds the information like root, infinitive form and imperative form of a verb which we require in different inflection cases. Nepali grammar has 3 main tenses, present, past and future. Past tense is further divided into unknown and known types where known types makes further two sub types Simple and Habitual, similarly future tense has definite and non-definite sub types. Unknown past is not very common in use so we have just considered Simple and Habitual subtype of past tense.

PTypes = Simpl | Hab ;

There are two aspects in Nepali grammar, perfective aspect and non-perfective aspect so the definition

Aspect = Perf | Imperf ;

One strange structure of Nepali grammar is that the negative verb form are formed at the morphological level. For the imperative, infinitive, conditional and participial forms morpheme *na/-न-* is prefixed and for the verbs stem elsewhere *-na/-न* suffixed, because of this rule the morphological inflection table of verb becomes very large. Parameter Polarity = Pos | Neg ; helps to determine whether its positive or negative form.

Nepali grammar has two modes, progressive and non progressive modes, as progressive form can be made easily from the root form of the verb by adding some prefix, instead of forming it in morphological level we introduce a data type '*ProgRoot*' along with parameters Aspect Number Gender which is sufficient to make the progressive form of verb and which helps to keep the inflection table small.

PVForm is a special form of a verb, which is used when another verb is used as complement of the main verb. An example of such a usage is 'want to eat'. In this grammatical construction 'eat' is used as complement of the auxiliary verb 'want'. In English infinitive form of the second verb (i.e. eat) is used as complement of the auxiliary verb (i.e. want). But in Nepali a special form of the second verb is used as complement of the first verb. We use constructor '*PVForm*' to store this form. It is used at syntax level for the constructions like 'want to eat'.

2.3.1 Morphological inflection and Structure in GF

Here we will see how the verbs are constructed in GF for Nepali, we will take the example of a v1 play/खेल्नु and see how this verb inflection makes morphological inflection table of different forms.

play_V : V defines the abstract definition of noun play and every language implementing this function should provide their own concrete definition, for Nepali we defined as follows

```
play_V = mkV "खेल्नु" ;
```

mkV is a lexical paradigm which takes infinite form of a verb and builds a inflection table based on the rules we have defined. *mkV* is a function used to handle all intransitive(v1) verbs, similarly we have *mkV2*, *mkV3*, *compoundV* to handle transitive, ditransitive and compound verbs. All these functions makes use of a function *mkVerb* which is defined as

```
mkVerb : (_: Str) -> Verb = \inf ->
  let root = (tk 2 inf) ;
  in {
    s = table {
      Root    => root ;
      Inf     => inf ;
      PVForm  => (tk 1 inf) ;
      Imp     => (mkImpForm root).s ;
      ProgRoot aspect number gender => (mkProgRoot root aspect number gender).s ;
      VF tense aspect polarity person number gender =>
        case aspect of {
          Imperf => (mkVImperf root tense polarity person number gender).s ;
          Perf   => (mkVPerf root tense polarity person number gender).s
        }
    }
  } ;
```

As defined in the function this function with the helper functions handles all the inflectional cases and builds the required inflection table. All these functions take the primary base case of a verb and to which certain suffix or prefix is added to form the inflection table. These primary base case of the verbs are formed by dropping certain suffixes from the infinitive verb, it is important to note that depending upon the base ending of verb the suffix we need to remove varies. Table bellow shows the different cases.

	Group Infinitive	Primary base
a.	गर्नु / to do	गर्
b.	खानु / to eat	खा
c.	धुनु / to wash	धु
d.	आउनु / to come	आउ

Table 2.2: Different types of root depending on word ending

Here we have four different cases:[6]

- Case of base ending in a consonant: बस्, गर्
- Case of base ending in the vowels आ, इ : खा, जा
- Case of base ending in vowel उ : धु, रु
- Case of base ending in vowels आउ, इउ: पिउ, पढाउ

To handle these cases we have a function called '*rootCheck*' which tells what type of root it is and depending upon its type of ending it applies the appropriate rule to get the correct root form. The code below shows how we achieve the different root forms for verbs.

```
rootCheck : Str -> {root1:Str; root2:Str; vcase: VCase} =
```

```
  \root -> {
    root1 = case root of {
      "ja"           => root + "nx:" ;
      "hu"           => root + "nx:" ;
      rot + "x:"     => root ;
      rot + ("h")    => root + "nx:" ; --cmnt
      rot + ("a"|"i"|"I") => root + "nx:" ;
      rot + ("e"|"u"|"U") => root + "nx:" ;
      rot + ("f"|"F")  => root + "V" ;
      -               => root
    };
```

```
  root2 = case root of {
    "ja"           => "g" ;
    "hu"           => "B" ; -- ङु -> भ
```



```

rot + "x:"          => rot ;
rot + ("a"|"i"|"I") => root ;
rot + ("h")         => root + "nx:" ; --cmnt
rot + ("e"|"u"|"U") => rot + "o" ;
rot + ("f"|"F")     => rot ;
-                   => root
};

```

-- it categories whether its regular or irregular form

vcase = case root of {

```

rot + "x:"          => VReg;
-                   => VReg
}
};

```

In the code above first case to match is the irregular inflection of जा, if the root matches to जा then its other roots will be root1 = जान् and root2 = ग, similarly if root case matches with हु then their roots will be root1 = हुन् and root2 = भ, similarly the definition of the root matching cases for पढ़, खा, धु and आर is also defined by the function.

Now we will see other inflection cases of verb. Table below is the verb case of non-perfective aspect, non-progressive mode, Simple present tense and positive polarity.

Person		Singular	Plural
First		छु / cu (पढ़छु)	छौं / cwV (पढ़छौं)
Second (male)	Low grade Honorific(Lgh)	छस् / csx: (पढ़छस्)	छौ / cw (पढ़छौ)
	Mid grade Honorific(Mgh)	छौ / cw (पढ़छौ)	छौ / cw (पढ़छौ)
	High grade Honorific(Hgh)	नुहुन्छ / nuhunx:c	नुहुन्छ / nuhunx:c

		(पढ्नुहुन्छ)	(पढ्नुहुन्छ)
Second (female)	Low grade Honorific(Lgh)	छेस् / cesx: (पढ्छेस्)	छौ / cw (पढ्छौ)
	Mid grade Honorific(Mgh)	छ्यौ / cx:yw (पढ्छ्यौ)	छौ / cw (पढ्छौ)
	High grade Honorific(Hgh)	नुहुन्छ / nuhunx:c (पढ्नुहुन्छ)	नुहुन्छ / nuhunx:c (पढ्नुहुन्छ)
Third (Male)	Low grade Honorific(Lgh)	छ / c (पढ्छ)	छन् / cnx: (पढ्छन्)
	Mid grade Honorific(Mgh)	छन् / cnx: (पढ्छन्)	छन् / cnx: (पढ्छन्)
	High grade Honorific(Hgh)	नुहुन्छ / nuhunx:c (पढ्नुहुन्छ)	नुहुन्छ / nuhunx:c (पढ्नुहुन्छ)
Third (Female)	Low grade Honorific(Lgh)	छे / ce (पढ्छे)	छन् / cnx: (पढ्छन्)
	Mid grade Honorific(Mgh)	छिन् / cinx: (पढ्छिन्)	छन् / cnx: (पढ्छन्)
	High grade Honorific(Hgh)	नुहुन्छ / nuhunx:c (पढ्नुहुन्छ)	नुहुन्छ / nuhunx:c (पढ्नुहुन्छ)

Table 2.3: Inflectional suffixes in Nepali verb for positive form, case of non-perfective aspect and non-progressive mode

(more information about the honorific form is available in pronoun phrases section)

Table below shows the inflectional suffixes of same case as above but negative polarity

Person		Singular	Plural
First		इँ / dinV (पढ्दिँ)	ऐँ / dEnEV (पढ्दैँ)
Second (male)	Low grade Honorific(Lgh)	ऐनस् / dEnsx: (पढ्दैनस्)	ऐने / dEnE (पढ्दैने)
	Mid grade Honorific(Mgh)	ऐने / dEnE (पढ्दैने)	ऐने / dEnE (पढ्दैने)
	High grade	नुहुन्छ / nuhunx:c	नुहुन्छ / nuhunx:c

	Honorific(Hgh)	(पढदिनुहुन्छ)	(पढदिनुहुन्छ)
Second (female)	Low grade Honorific(Lgh)	इनस् / dinsx: (पढदिनस्)	ऐनै / dEnE (पढ्दैनौ)
	Mid grade Honorific(Mgh)	इनै / dinE (पढदिनौ)	ऐनै / dEnE (पढ्दैनौ)
	High grade Honorific(Hgh)	नुहुन्छ / nuhunx:c (पढदिनुहुन्छ)	नुहुन्छ / nuhunx:c (पढदिनुहुन्छ)
Third (male)	Low grade Honorific(Lgh)	ऐन / dEn (पढ्दैन)	ऐनन् / dEnx:n (पढ्दैनन्)
	Mid grade Honorific(Mgh)	ऐनन् / dEnx:n (पढ्दैनन्)	ऐनन् / dEnx:n (पढ्दैनन्)
	High grade Honorific(Hgh)	नुहुन्छ / nuhunx:c (पढदिनुहुन्छ)	नुहुन्छ / nuhunx:c (पढदिनुहुन्छ)
Third (female)	Low grade Honorific(Lgh)	इन / din (पढदिन)	ऐनन् / dEnx:n (पढ्दैनन्)
	Mid grade Honorific(Mgh)	इनन् / dinx:n (पढदिनन्)	ऐनन् / dEnx:n (पढ्दैनन्)
	High grade Honorific(Hgh)	नुहुन्छ / nuhunx:c (पढदिनुहुन्छ)	नुहुन्छ / nuhunx:c (पढदिनुहुन्छ)

Table 2.4: Inflectional suffixes in Nepali verb for negative form, case of non-perfective aspect, non-progressive mode

These tables in GF is defined by the function *mkVPreNPReg*, which looks like as below

mkVPreNPReg : Str -> Polarity -> NPerson -> Number -> Gender -> {s:Str} =

\root, po, pn, n, g ->

{s = case <po, pn, n, g> of {

-- Positive case

<Pos, Pers1, Sg, _> => root + "छु" ;

<Pos, Pers1, Pl, _> => root + "छौँ" ;

<Pos, Pers2_L, Sg, Masc> => root + "छस्" ;

<Pos, Pers2_L, Sg, Fem> => root + "छेस्" ;

```

<Pos, Pers2_L, Pl, _>    => root + "छौ" ;
<Pos, Pers2_M, Sg, Fem> => root + "छयी" ;
<Pos, Pers2_M, _, _>    => root + "छौ" ;
<Pos, Pers3_L, Sg, Masc> => root + "छ" ;
<Pos, Pers3_L, Sg, Fem> => root + "छे" ;
<Pos, Pers3_L, Pl, _>    => root + "छन्" ;
<Pos, Pers3_M, Sg, Fem> => root + "छिन्" ;
<Pos, Pers3_M, _, _>    => root + "छन्" ;
<Pos, _, _, _>          => root + "नुहुन्छ" ;

```

-- Negative Case

```

<Neg, Pers1, Sg, _>      => root + "दिनँ" ;
<Neg, Pers1, Pl, _>      => root + "दैनँ" ;
<Neg, Pers2_L, Sg, Masc> => root + "दैनस्" ;
<Neg, Pers2_L, Sg, Fem> => root + "दिनस्" ;
<Neg, Pers2_L, Pl, _>    => root + "दैनै" ;
<Neg, Pers2_M, Sg, Fem> => root + "दिनै" ;
<Neg, Pers2_M, _, _>    => root + "दैनै" ;
<Neg, Pers3_L, Sg, Masc> => root + "दैन" ;
<Neg, Pers3_L, Sg, Fem> => root + "दिन" ;
<Neg, Pers3_L, Pl, _>    => root + "दैन्र" ;
<Neg, Pers3_M, Sg, Fem> => root + "दिन्न" ;
<Neg, Pers3_M, _, _>    => root + "दैन्र" ;
<Neg, _, _, _>          => root + "नुहुन्न"
}

```

};

We can clearly see that the function takes parameters like polarity, person, number, gender and creates an inflection table, also the suffixes added for the positive polarity and negative polarity is distinctly different so we need morphological inflection for the negative verbs as well.

Inflection table for the simple past tense looks like as shown in table below,

Person		Singular	Plural
First		एँ / eV (पढें)	यौँ / ywM (पढ्यौँ)
Second (male)	Low grade Honorific(Lgh)	इस् / isx: (पढिस्)	यौ / yw (पढ्यौ)
	Mid grade Honorific(Mgh)	यौ / yw (पढ्यौ)	यौ / yw (पढ्यौ)
	High grade Honorific(Hgh)	नुभयो / nuByo (पढनुभयो)	नुभयो / nuByo (पढनुभयो)
Second (female)	Low grade Honorific(Lgh)	इस् / isx: (पढिस्)	यौ / yw (पढ्यौ)
	Mid grade Honorific(Mgh)	यौ / yw (पढ्यौ)	यौ / yw (पढ्यौ)
	High grade Honorific(Hgh)	नुभयो / nuByo (पढनुभयो)	नुभयो / nuByo (पढनुभयो)
Third (male)	Low grade Honorific(Lgh)	यो/yo (पढ्यो)	ए/e (पढे)
	Mid grade Honorific(Mgh)	ए/e (पढे)	ए/e (पढे)
	High grade Honorific(Hgh)	नुभयो / nuByo (पढनुभयो)	नुभयो / nuByo (पढनुभयो)
Third (female)	Low grade Honorific(Lgh)	इ/I (पढी)	ए/e (पढे)
	Mid grade Honorific(Mgh)	इन्/inx: (पढिन)	ए/e (पढे)
	High grade Honorific(Hgh)	नुभयोस/nuByo (पढनुभयो)	नुभयो/nuByo (पढनुभयो)

Table 2.5: Inflectional suffixes for simple past positive form

Similarly the negative form have different inflection form.

Inflectional suffixes of definite future tense positive form

Person		Singular	Plural
First		नेछु/necu	नेछौं/necwM
Second (male)	Low grade Honorific (Lgh)	नेछस्/necsx: (पद्नेछस्)	नेछौं/necwM (पद्नेछौं)
	Mid grade Honorific (Mgh)	नेछौं/necwM (पद्नेछौं)	नेछौं/necwM (पद्नेछौं)
	High grade Honorific (Hgh)	नुहुनेछ/nuhunec (पद्नुहुनेछ)	नुहुनेछ/nuhunec (पद्नुहुनेछ)
Second (female)	Low grade Honorific (Lgh)	नेछस्/necsx: (पद्नेछस्)	नेछौं/necwM (पद्नेछौं)
	Mid grade Honorific (Mgh)	नेछौं/necwM (पद्नेछौं)	नेछौं/necwM (पद्नेछौं)
	High grade Honorific (Hgh)	नुहुनेछ/nuhunec (पद्नुहुनेछ)	नुहुनेछ/nuhunec (पद्नुहुनेछ)
Third (male)	Low grade Honorific (Lgh)	नेछ/nec (पद्नेछ)	नेछन्/necnx: (पद्नेछन्)
	Mid grade Honorific (Mgh)	नेछन्/necnx: (पद्नेछन्)	नेछन्/necnx: (पद्नेछन्)
	High grade Honorific (Hgh)	नुहुनेछ/nuhunec (पद्नुहुनेछ)	नुहुनेछ/nuhunec (पद्नुहुनेछ)
Third (female)	Low grade Honorific (Lgh)	नेछ/nec (पद्नेछ)	नेछन्/necnx: (पद्नेछन्)
	Mid grade Honorific (Mgh)	नेछिन्/necinx: (पद्नेछिन्)	नेछन्/necnx: (पद्नेछन्)
	High grade Honorific (Hgh)	नुहुनेछ/nuhunec (पद्नुहुनेछ)	नुहुनेछ/nuhunec (पद्नुहुनेछ)

Table 2.6: Inflectional suffixes of definite future tense positive form

Likewise we have many other functions like *mkVPstHNP*, *mkVFutNDNP*, *mkVPreP*, *mkVPstSP*, *mkVPstHP*, *mkVFutDefP* and *mkVFutNDefP* to cover all the different forms of positive and negative verbs.

2.4 Adjective

The main role of adjective is to qualify a noun or noun phrase. In Nepali adjective that ends in -o/ो inflects for gender, and number giving different form of adjective root.

Table below shows the inflections of adjectives.

Singular form		Plural form:
Masculine	Feminine	Masculine/Feminine
ramro/राम्रो 'handsome'	ramri/राम्री 'beautiful'	ramra/राम्रा 'handsome, beautiful'
batho/बाठो 'clever'	bathi/बाठी 'clever'	bathi/बाठा 'clever'
lato/लाटो 'dumb'	lati/लाटी 'dumb'	lata/लाटा 'dumb'
kalo/कालो 'black'	kali/काली 'black'	kala/काला 'black'

Table 2.7: Adjective inflection forms[1].

There are many borrowed words in Nepali specially from Sanskrit and Hindi sources. These words doesn't show inflection so they form a category of uninflected adjectival forms. They show the same distribution and functions as adjectives, e.g.

Aasal keto/असल केटो 'good boy'

Aasal ktaharu/असल केटाहरु 'good boys'

Aasal keti/असल केटी 'good girl'

Aasal ketiharu/असल केटीहरु 'good girls'

Here असल/good takes same form for both singular and plural masculine and feminine forms.

2.4.1 Functional of adjectives

Normally adjectives function as the head of a adjective phrase structure. The adjective phrase also function as pre-head modifiers in the noun phrases structures[1], e.g. birami manish/बिरामी मानिस 'a sick man'. The dependents of the adjectives are quantifiers which qualify the adjectives by showing the degree of intensity including the comparative and superlative forms.

2.4.2 Some of the characteristics of Nepali adjectives

- The demonstrative adjectives are यो/this and त्यो/that[8]
- the word 'good' has many different forms, one or other is more suitable depending on case[8].
 - ramro/राम्रो Strictly speaking means 'pleasing to eye', 'beautiful' is used in most sense of the English 'good'
 - aasal/असल means 'of good quality' or 'morally good', e.g. त्यो राम्रो मान्छे हो/He is a good man.

- besa/बेस and its emphatic form बेसै refers to quality, e.g. यो कलम् बेस छ/This pen is good.
- mitho/मीठो means 'good to taste' and is only used for food and drinks
- Repetitive of an adjective indicates plurality, e.g. thula yhula/ठूलठूला meaning big, त्यहाँ ठूलठूला घरहरू छन् । 'There are big houses'.
- Adjectives ending in -ओ and postpositions -को have a feminine singular form in -ई, eg, budhi/बूढी 'old women'
- There are few cases though ends in -o they doesn't inflect for feminine gender like others e.g. rato/रातो 'red'

2.4.3 Implementation of adjective in GF

Adjectives have the following structure in our construction.

$A = \{s : \text{Number} \Rightarrow \text{Gender} \Rightarrow \text{Str}\};$

now we will take an example of a adjective and see its construction in GF, lets see an example of a adjective ramro/राम्रो 'good', the abstract definition of good is given in gf as $\text{good_A} : A$; so in Nepali it is implemented as $\text{good_A} = \text{mkA "राम्रो"}$; mkA is an overloaded function which calls another function mkAdjnp , it is the main function which takes cares of all the inflection and exception cases

$\text{mkAdjnp} : \text{Str} \rightarrow \text{npAdjective} = \backslash \text{str} \rightarrow$

```

case str of {
  st + t@"त" + "ो" => mkAdj1 str str (st+t+"ी"); -- No Fem for red
  st + "ो"         => mkAdj1 str (st+"ी") (st+"ी");
  _                => mkAdj1 str str str
};

```

the helping function mkAdj1 makes the inflectional table out of different inflection forms as described in table: 2.7

$\text{mkAdj1} : (x1, _, x3 : \text{Str}) \rightarrow \text{npAdjective} =$

```

\sm, sf, smf -> {
  s = table {
    Sg => table { Masc => sm ; Fem => sf };

```



```
Pl => table { Masc => smf ; Fem => smf }  
};
```

2.5 Adverbs

Adverbs can modify verbs, adjectives, clauses and other adverbs, and they do not show inflections. Adverbs show the gradation of comparative and superlative degrees by syntactic means of their dependents quantifiers or adverbs of quantity[1]. Adverbs can either occur as independent of or as the head of an adverbial phrase structure, and function as dependents of the verb i.e. as complements or adjuncts, e.g. ramrari khau/राम्ररी खाउ 'eat well'. Adverbs can also function as quantifiers or intensifiers of adjectives, e.g. dharai ramro /साहै राम्रो 'very good'. The adverbs which function as quantifiers of adjectives or quantifiers of other adverbs are adverbs of quantity. We should note that the the comparative and superlative formations of the adverbs are syntactic, not morphological e.g.

Comparative: aali bistari/अलि बिस्तारी 'more slowly'

Superlative: jyadai bistari/ज्यादै बिस्तारी 'extremely slowly'

In GF adverb is simply defined as $Adv = \{s : Str\}$; as it doesn't have any inflectional forms, and is constructed using function *mkAdv* e.g. here $Adv = mkAdv$ "यहाँ" ; . *MkAdv* is defined as $mkAdv : Str \rightarrow Adv = \lambda str \rightarrow \{s = str\}$; as we can see from the definition there are no inflection cases, it simply takes a string and makes it *Adv* type.

2.6 Pronouns

A pronoun can replace a noun or another pronoun. Pronouns constitute a small closed class of forms and their inflectional behavior is similar to nouns i.e. they inflects for case (seven cases we defined previously) and number (singular and plural) . Pronouns belong indirectly to the gender of nouns. The gender of pronoun is taken from the gender of noun, it is shown syntactically in the third person by its cross reference tie to verb for which they function as subject. Pronouns occur as head of the pronoun phrase and function as subject or object complements and adjuncts of verbs[1]. Pronouns are marked by their simple forms that distinguish them from other form classes.

In GF pronoun 'I' is defined as `i_Pron = mkPron "म" "मेरो" Sg Masc Pers1` ; `mkPron` finally calls the function `makePron` which is defined as,

```
makePron : (x1,_,_,_,x6 : Str) -> {s : Case => Str} =
  \n,ac,i,d,ab,l-> {
    s = table { Nom => n ; Acc => ac ; Ins => i ;
              Dat => d ; Abl => ab ; Loc => l }
  };
```

2.7 Conjunctions

2.7.1 Coordinating conjunctions

Coordinating conjunctions are a closed class of uninflected forms[1][2].

ani/अनि 'and then'

ki/कि 'or'

kintu/किन्तु 'but'

naki/नकि 'but not'

ra/र 'and'

tara/तर 'but'

wa/वा 'or'

athava/अथवा 'or'

ki...ki/कि ... कि 'either ... or'

nn...na/न ... न 'neither ... nor'

parantu/परन्तु 'but'

taipani/तैपानि 'even then'

tatha/तथा 'and'

ya/या 'or and'

These conjunctions are used in Nepali.

2.7.2 Subordinating conjunctions

subordinating conjunctions are a closed un-inflected classes. The closed list is: की/that, भने/if, पछि/after, अघि/before, पनि/although, and the subordinating relative conjunctions.

The function of subordinating conjunctions is to mark dependent (adverbial or noun) clauses as subordinate to the principal clause in sentential structures[1].

ki/कि 'that' (makes noun clause)

pachadi/पछाडि 'after' (makes adverbial clause)

pachi/पछि 'after' (makes adverbial clause)

pachi/पछि 'if' (makes adverbial clause)
yadi/यदि 'if' (makes adverbial clause)
agadi/अगाडि 'before' (makes adverbial clause)
bhane/भने 'if' (makes adverbial clause)
aghi/अघि 'before'(makes adverbial clause)
pani/पनि 'although'(makes adverbial clause)
yadhapi/यध्यपि 'if' (makes adverbial clause)

2.8 Numerals

Numerals in Nepali can be divided in to:

- Cardinal adjectives or adjectivals, answering 'how many'.
- Ordinal adjectives answering 'which one of a series'.
- Distributive adjectives answering 'how many each' and
- Ordinal adverbials answering 'which time of a series'.

Table below shows the different cases for the numbers one, two and three

Cardinal adjectives	Ordinal adjectives	Distributive adjectives	Ordinal adverbials
On/एक	first/पहिलो	one-one/एक-एक	once or first time/ एक पल्ट
Two/दुइ	second/दोस्रो	two-two /दुइ-दुइ	twice or second time/ दुइ पल्ट
Three/तीन	third/तेस्रो	three-three/तीन-तीन	thrice or third time/ तीन पल्ट

Table 2.8: Different forms of numeral cases.

The numerals below 100 in Nepali is irregular, so we represent this construction in GF by the following algebraic type

$DSize = sg \mid r2 \mid r3 \mid r4 \mid r5 \mid r6 \mid r7 \mid r8 \mid r9 ;$

where sg defines all numerals ending in 1 e.g. 1, 11, 21 etc similarly r2 represent numerals ending in 2 e.g. 12, 22, same applies for remaining cases. Another feature of numerals we need to consider is the size, which is defined as $Size = singl \mid less100 \mid more100 ;$

In Nepali numbers multiply by ten are written in the following way

१ एक /1

१ 0 दश /10

१ 00 सय / 100

१,000 हजार / 1,000

१ 0,000 दश हजार / 10,000

१,00,000 एक लाख / 1,00,000

१ 0,00,000 दस लाख / 10,00,000

१,00,00,000 करोड / 1,00,00,000

१ 0,00,00,000 दश करोड / 10,00,00,000

Chapter 3

3. Syntax

This chapter describes the syntax of Nepali grammar and its implementation in Grammatical Framework.

3.1 Noun-Phrases(NP)

Noun-phrase for Nepali in GF is defined as

$$\text{NP} : \text{Type} = \{s : \text{NPCase} \Rightarrow \text{Str} ; a : \text{Agr} ; t : \text{NType} \};$$

where

$$\text{NPCase} = \text{NPC Case} \mid \text{NPObj} \mid \text{NPErg} ;$$
$$\text{Case} = \text{Nom} \mid \text{Acc} \mid \text{Ins} \mid \text{Dat} \mid \text{Abl} \mid \text{Loc} ;$$
$$\text{Agr} = \text{Ag Gender Number NPerson} ;$$
$$\text{NType} = \text{Living} \mid \text{NonLiving} ;$$

3.1.1 Common-noun phrase(CNP)

A common noun phrase in Nepali consists of three functional slots, an optional determiner, an optional modifier slot marked by the sign \pm and an obligatory head slot marked by the sign $+$. The structure of the CNP is represented by the following formula

$$\text{CNP} = \pm \text{Determiner} \pm \text{Modifier} + \text{Head}[1]$$

eg. tyo bishan bhaban/त्यो विशाल भवन् 'hat huge building'

3.1.2 Common nouns as head

When common nouns act as head they have following structure.

Head: +stem ± plural suffix + inflectional suffix[1]

The structure shows that the common noun can either be single or plural, and the plural suffix is optional. The inflectional suffix is required if it is necessitated semantic purpose, eg.

मानिस/man, मानिस(हरु)/men, मानिस(हरु)लाई/to men

The common nouns ending in -o/ो like boko/बोको, doko/डोको, etc. have their allomorphs ending in -a such as boka/बोका, doka/डोका, etc[1]. when they are followed by the optional plural marker, or by a case inflection they show the following pattern

Singular	Plural	Inflectional forms
doko/डोको 'a basket'	doka(haru)/डोका(हरु) 'baskets'	dokama/डोकामा 'in the basket'
boko/बोको 'male goat'	boka(haru)/बोका(हरु) 'goats'	bokale/बोकाले 'by the male goat'
choro/छोरो 'son'	chora(haru)/छोरा(हरु) 'sons'	choralai/छोरालाई 'to the son'

Gender of Nouns

Although the traditional Nepali grammar talks about masculine, feminine, neuter and undefined genders but in modern grammar every noun belongs to either masculine or feminine gender as reflected morphologically in the verbs[1]. Thus gender in Nepali is a syntactic property, in other words the gender of nouns is not decided by the form of noun instead it is indicated morphologically by the form of verbs e.g.

शारदा जान्छ/sharda jancha 'Sharada goes' शारदा जान्छे/sharda janche 'Sharada goes'

In the example above Sharada can be names of both man or woman, but the difference in the gender is reflected in and by the form of verbs.

3.1.3 Determiners in the common-noun phrase

Determiners in cnp can be summarized by the following formula.

±determiner: ± demonstrative ± limiter ± quantifier ± classifier[1][2]

The demonstrative, limiters, quantifiers and classifiers can be called 1st, 2nd, 3rd and 4th order determiners.

In GF, Nepali determiners are defined as

Determiner = {s : Number => Gender => Str ; n : Number};

so we can say that inflection of determiners depends on number and gender.

3.1.3.1 Demonstratives

Demonstratives determiners are a small closed subset of determiners which are inflected only for numbers[2]. e.g. यो/yo 'this', यी/yi 'these', and remote: त्यो/tyo 'that', ती/ti 'those' are the commonly used demonstrative determiners in Nepali.

3.1.3.2 Quantifiers

The quantifiers are cardinal numbers like ek/एक 'one', dui/दुई 'two' followed by one of the two classifiers i.e. either human classifier or non-human classifiers. The quantifiers followed by classifiers. The classifier जना/jana occurs with human [+human] count nouns; the classifier वटा/wata occurs with non-human [-human] count nouns. eg.

human [+ human] classifier: pach jana manis /पाँच जना मानिस 'Five men'[2]

non-human [- human] classifier: pach wata kalam /पाँच वटा कलम 'Five pens'[2]

Note that the form of एक/ek 'one' and wata classifier is euta. Other forms show two free variants each eg.

dui wata/दुई वटा 'two ones'

duita /दुईटा 'two ones'

tin wata/तीन वटा 'three ones'

tinota /तिन्टा 'three ones'

1.3.2 Limiters

They are the closed set of quantifiers which as determiners follow the demonstratives in the linear order of occurrence . The limiters are either definite or indefinite. e.g.

definites:

हरेक/harek 'each', प्रत्येक/pratyek 'each'

indefinites:

केही/kehi ", एकै/eaklai 'only', अन्तिम्/antim 'final',

The nouns and pronouns in possessive forms also function as limiters.

3.1.4 Modifiers in the common-noun phrase(CNP)

Nouns or noun phrase, adjective or adjective phrase and the clause that act as adjectives can function as a modifier in the common noun phrases. When noun or noun phrases co-occur with a common nouns the first noun is the modifier and the final noun is the head of the common noun phrase e.g.

nepali bhasa /नेपाली भाषा 'Neplai Language'

Similarly in case of adjectives and adjective phrases, they can act as modifiers after the determiners and before the head of the common noun phrases[1] e.g.

tyo sano sahar/त्यो सानो शहर 'That small city'

The clauses that function as adjectives also acts as a dependent modifier of the nouns e.g.

johan birami bhayako bkhat /जोन विरामि भएको बखत 'The time when Johan was sick'

here the clause जोन विरामि भएको modifiers the item बखत

functions like *AdjCN*, *ComplN2*, *AdvCN*, *RelCN* are written in grammatical framework to handle the cases that can modify common nouns.

3.2 Proper-noun phrase

The proper-noun phase in Nepali has an optional modifier, an obligatory proper noun head and is filled by the place and person names. Place and person names in Nepali do not co-occur with determiners like demonstrative, limiters or quantifiers, however the person names do occur with the modifiers like adjectives or adjectival modifiers[1].

3.2.1 Person names

Person names when functioning as heads consists of an obligatory person name with optional modifiers but without plural number[1]. As obvious persons name does not inflect for number as the person name is inherently singular as it refers to an individual is inherently singular. When the plural number suffix -haru/हरु occurs with a person name it does not mean plural number marker it stands for the others, the unspecified names semantically associated with the person name with which it occurs, for examples when we say जोनहरु/johanharu that implies 'Johan and others'

Like common nouns, personal nouns also have a syntactic property of gender which is in one-to-one correspondence with the sex of the individual that is reflected to by the name[1]. The gender of the person name is reflected in the gender concord they have with the form of verbs, e.g.

Sharada as male Sharada gayo/ शारदा गयो 'Sharada went'

Sharada as female Sharada gayi /शारदा गई 'Sharada went'

Devi as male Devi gayo/देवी गयो 'Devi went'

Devi as female Devi gayi/देवी गई 'Devi went'

When the person names of masculine gender such as शारदा प्रसाद/Sharada Prasad and देवी प्रसाद /Devi Prasad reduces to Sharada and Devi, their gender is ambiguous as they can refer to male as well as female, only way their gender is disambiguated is by the verb in the clausal

structure.

In GF implementation proper names to noun phrase has been defines as

```
UsePN pn = { s = \\c => toNP pn.s c ;
              a = toAgr Sg pn.h pn.g ;
              t = pn.t
            } ;
```

3.2.2 Place names

Place names when function as head consists of an obligatory place name, they do not inflect for number as they refer to one geographical place name. e.g. Kathmandu, Göteborg

3.3 Pronoun Phrase

The pronoun phrase consists of an optional modifier slot and the obligatory head shot, it can be formulated as

pronoun-phrase = ±modifier +head ±modifier[1]

where ±modifier is optional modifier and +head is obligatory pronoun head. In some cases the optional modifier occurs after head this applies to few personal pronouns and doesn't occur with other personal pronouns.

3.3.1 Pronouns as heads

Pronouns are a small closed class and takes indirectly the gender of the nouns which they substitute but are not inflected for gender. The gender of the pronouns is expressed morphologically by the verbs with which they stand in syntactic construction[1]. Compared to nouns pronouns inflect more irregularly for case and number. Table below shows the personal pronouns and their honorific forms.

Person		Singular	Plural
First		ma/म 'I'	hami-(haru)/हामी-(हरु) 'we'
Second (male)	Low grade Honorific(Lgh)	ta/तँ	timi-haru/तिमी-हरु
	Mid grade Honorific(Mgh)	timi/तिमी	timi-haru/तिमि-हरु

	High grade Honorific(Hgh)	tapai/तपाई	tapai-haru/तपाइ-हरु
Second (female)	Low grade Honorific(Lgh)	u, tyo/उ, त्यो	uni-haru, tini-haru/उनि-हरु, तिनी-हरु
	Mid grade Honorific(Mgh)	tini/तिनी	tini-haru/तिनी-हरु
	High grade Honorific(Hgh)	waha/वहाँ	waha-haru/वहाँ-हरु

Table 3.1: Personal pronouns and their honorific forms.

In formal conversations tapai and tapai- haru show variants yeha and yeha-haru

Cases	1 st p.	2 nd p. (LGH)	2 nd p. (MGH)	2 nd p. (HGH)	3 rd p. (LGH)	3 rd (MGH)	3 rd (HGH)
Nm	ma/ म	ta/ तँ	timi/ तिमी	tapai/ तपाई	u, tyo/ उ, त्यो	uni/ उनी	waha/ वहाँ
Ac	malai/ मलाई	talai/ तँलाई	timilai/ तिमीलाई	tapailai/ तपाईलाई	uslai/ उस्लाई	unalai/ उनीलाई	wahalai/ वहाँलाई
In	maile/ मैले	taile/ तँले	timile/ तिमीले	tapaile/ तपाईले	usle/ उस्ले	unale/ उनीले	wahale/ वहाँले
Dt	malai/ मलाई	talai/ तँलाई	timilai/ तिमीलाई	tapailai/ तपाईलाई	uslai/ उस्लाई	unalai/ उनीलाई	wahalai/ वहाँलाई
Ab	mabata/ मबाट	tabata/ तँबाट	timibata/ तिमीबाट	tapaibata/ तपाईबाट	usbata/ उस्बाट	unabata/ उनीबाट	wahabata/ वहाँबाट
Gn	mero/ मेरो	tero/ तेरो	timro/ तिम्रो	tapaiko/ तपाईको	usko/ उस्को	unako/ उनीको	wahako/ वहाँको
Lc	mama/ ममा	tama/ तँमा	timima/ तिमीमा	tapaima/ तपाईमा	usma/ उस्मा	unama/ उनीमा	wahama/ वहाँमा

Table 3.2: Inflection of the personal pronouns in singular number

The function *UsePron* converts pronouns to noun phrase, which is defined as

```
UsePron p = {s = \\c => np2pronCase p.s c p.a ;
             a = p.a ;
             t = Living
             } ;
```

functions like *makePron* and *makePronReg* builds the inflection table of pronouns.

makePronReg builds the inflection table for regular nouns where as *makePron* builds the inflection of other cases. The function *makePronReg* is defined as follows

```
makePronReg str=makePron str (str + "लाई") (str + "ले") (str + "लाई") (str + "बाट") (str + "मा") ;
```

The plural case of the pronouns are also handled by the same functions as we call these functions separately for single and plural cases but we pass a parameter to indicate whether its singular or plural case. The example below makes it clear

```
i_Pron = mkPron "म" "मेरो" Sg Masc Pers1 ;
```

```
we_Pron = mkPron "हामीहरु" "हामीहरुको" Pl Masc Pers1
```

These function creates the inflection table of I and we pronouns. The inflections of pronouns in singular and plural form are given in the table 3.2 and table 3.3

Case s	1 st p.	2 nd p. (LGH & MGH)	2 nd p. (HGH)	3 rd p. (LGH & MGH)	3 rd (HGH)
Nm	hami-haru/ हामि-हरु	hami-haru/ हामि-हरु	tapaiharu/ तपाईंहरु	tiniharu/ तिनीहरु	wahaharu/ वहाँहरु
Ac	hami-(haru)lai/ हामि-(हरु)लाई	hami-harulai/ हामि-(हरु)लाई	tapaiharulai/ तपाईंहरुलाई	tiniharulai/ तिनीहरुलाई	wahaharulai/ वहाँहरुलाई
In	hami-(haru)le/ हामि-(हरु)ले	hami-harule/ हामि-हरुले	tapaiharule/ तपाईंहरुले	tiniharule/ तिनीहरुले	wahaharule/ वहाँहरुले
Dt	hami-(haru)lai/ हामि-(हरु)लाई	hami-harulai/ हामि-हरुलाई	tapaiharulai/ तपाईंहरुलाई	tiniharulai/ तिनीहरुलाई	wahaharulai/ वहाँहरुलाई
Ab	hami-(haru)bata/ हामि-(हरु)बाट	hami-harubata/ हामि-हरुबाट	tapaiharubata/ तपाईंहरुबाट	tiniharubata/ तिनीहरुबाट	wahaharubata/ वहाँहरुबाट

	हामि-(हरु)बाट	हामि-(हरु)बाट	/ तपाईंहरुबाट	तिनीहरुबाट	वहाँहरुबाट
Gn	hami-haruko, hamro/ हामि-हरुको, हाम्रो	hami-haruko/ हामि-हरुको	tapaiharuko/ तपाईंहरुको	tiniharuko/ तिनीहरुको	wahaharuko/ वहाँहरुको
Lc	hami-(haru)ma/ हामि-(हरु)मा	hami-haruma/ हामि-हरुमा	tapaiharuma/ तपाईंहरुमा	tiniharuma/ तिनीहरुमा	wahaharuma/ वहाँहरुमा

Table 3.3: Inflection of the personal pronouns in plural number

3.3.2 Personal pronoun phrase modifiers

Pronouns do not occur with the demonstratives, numerals and classifiers determiners, but they do co-occur with certain modifiers. e.g.

keval/केवल 'only'

khali/खाली 'only'

The following modifiers follow the pronouns they modify[2]:

eklai/एकलै 'alone'

matrai/मात्रै 'only, alone'

dubai/दुबै 'both'

sabai/सबै'all'

aphai/आफै'oneself'

These modifiers precede the pronouns they stand in construction with.

3.4 Adjectival Structures

The abstract syntax of Grammatical Framework defines the principal ways of forming an adjectival phrase, they are positive, comparative, relational, reflexive-relational, and elliptic-relational. The following functions helps to build the required adjectival phrases.

PositA : A -> AP ;

ComparA : A -> NP -> AP ;

ComplA2 : A2 -> NP -> AP ;

ReflA2 : A2 -> AP ;

CAdvAP : CAdv -> AP -> NP -> AP ;

The internal structure of Nepali adjective consists of the obligatory head slot filled by adjective and an optional modifier slot filled by a qualifier or quantifier adverb.

An obligatory adjective occurs as the head within the internal structure of the adjective phrases[1].

ramro/राम्रो 'handsome'

asal/असल 'good'

aglo/अग्लो 'tall'

Adjectives ending in -o/ओ which occur in the head reflects the gender and number of the noun with which they stand in construction. e.g.

ramro keto/राम्रो केटो 'handsome boy'

ramri keti/राम्री केटी 'beautiful girl'

ramra ketaharu/राम्रा केटाहरु 'handsome boys'

ramra ketiharu/राम्रा केटीहरु 'good girls'

In the above examples ramro indicates the male gender where as ramri indicates the female gender where as ramra is used both for masculine and feminine gender. To show this relation in GF we have defined the function *mkAdjnp* as

```
mkAdjnp : Str -> npAdjective = \str ->
  case str of {
    st + "ो"   => mkAdj1 str (st+"ी") (st+"ा");
    _         => mkAdj1 str str str
  };
```

which will check string against the -o ending and necessary suffixes are added, then the function *mkAdj1* will build the inflection table which will build all the necessary inflection cases.

```
mkAdj1 : (x1,_,x3 : Str) -> npAdjective =
  \sm, sf, smf -> {
    s = table {
      Sg => table { Masc => sm ; Fem => sf };
      Pl => table { Masc => smf ; Fem => smf }
    }
  };
```

The function *mkAdj1* distributes the input parameter in required category of inflection table.

Adjectives which end in -o/ओ show inflected evaluative forms ending in -ai/ँ which show an evaluative degree of quality. Then an evaluative connotation lai/'fairly' or 'more or less' is added to the meaning of such adjectives[1][2]. e.g.

thulo/ठुलो 'big' → thulai/ठुलै 'fairly big'
sano/सानो 'small' → sanai/सानै 'fairly small'
aglo/अग्लो 'tall' → aglai/अग्लै 'fairly small'

The adjective sab/सब 'all', though not ending in -o/ओ, also shows an inflection for its evaluative form ending in -ai which adds to its meaning the concatenation 'fairly' or 'more or less'. For instance sab/सब 'all' vs. sabai/सबै 'more or less all'

3.4.2 Quantifiers in the Adjp

The adverbs of quantity are

ajha/अझ 'more' alik/अलीक 'somewhat'
alikasi/अलिकति 'a little' bahut/बहुत 'very'
kehi/केही 'some, somewhat' sarhai/साहै 'extremely'

these quantifiers function as quantifying determiners in the common noun phrase, e.g.

ajha ramro ghar/अझ राम्रो घर 'more beautiful house'

3.4.2.1 Comparative with bhanda/भन्दा 'than'

Comparative with bhanda follows the following construction order +nominal +comparative bhanda +nominal +head(adj)

e.g. bhagya bhanda purushrtha thulo [ho]/भाग्य भन्दा पुरुषार्थ ठुलो [हो][1]

'Hard work is greater than luck'

in GF the function ComparA implements this case

3.4.2.2 Comparative Adjp with jhan/झन 'the more'

comparative adjective phrase can be build by the rule +nominal + comparative (jhan) + head

e.g. taha jhan thulo rookha chha/त्यहाँ झन ठुलो रुख छ।

'There is more bigger tree'

3.4.2.3 Superlative quantifier phrases

Superlative quantifier phrase with sabbhanda/सबैभन्दा 'most-of-all' and an adjective head.

e.g. +sagarmatha + sabhanda +aglo [pahada +ho]/+सगरमाथा +सबभन्दा अग्लो [+पाहाड +हो]

'Mt. Everest is the highest mountain'

The order of the constituents of the superlative quantifier phrases indicates a statistical order. However the position of the subject is changeable. This is illustrated by the following example.

+Sababhanda +thulo [+santosa +yehi +ho]/+सबभन्दा +ठुलो [+सन्तोष +यही +हो]

'This is the greatest satisfaction'.

3.5 Modifiers within common noun phrases

The dependent adjectivals modifies the common noun phrases. The common noun phrase structures with dependent adjectivals have an optional modifier slot filled by one of the two types of the dependent adjectivals and an obligatory head slot filled by a noun.

common noun phrase = +modifier +head

Dependant adjectivals are derived verbal adjectives. The participles are divided into two subclasses

1. imperfect participle marked by the suffix -ne/ने
2. perfect participle marked by the suffix -eko/एको.

The imperfect participle marked by the definitional suffix -ne, functioning as a dependent adjectival, is not inflected for tense, person, number, gender and aspect. The perfect particles marked by the derivational suffix -eko/एको is not inflected for tense and person but inflected for number and gender, e.g.

Singular		Plural
Masculine	Feminine	Masculine, Femenine
-eko/'एको'	-eki/'एकी'	-eka/'एका'

In the case when the non-progressive vs progressive mode is marked, the imperfect participial suffixes -ne, and perfect participial suffix -eko follow the progressive mode marker -irah-.

The progressive mode is unmarked, e.g.

Non-progressive mode		Progressive mode	
Imperfect participial	Perfect participial	Imperfect participial	Perfect participial
-ne/ने	-eko/एको (m. sg.)	-iraha-ne/इरहने	-irah-eko/इरहेको (m. sg)
-ne/ने	-eki/एकी (f. sg.)	-iraha-ne/इरहने	-irah-eki/इरहेकी (f. sg)
-ne/ने	-eka/एका (m/f pl.)	-iraha-ne/इरहने	-irah-eka/इरहेका (m/f pl.)

Table 3.4: non-progressive and progressive mode[1].

3.6 Adverbial Structures

Adverbial phrase consists of an optional complement slot filled by an instrumental, dative, ablative or locative complement, an optional modifier slot filled by a quantifying adverb, and an obligatory head filled by simple adverbs, or adverbials, or compound adverbials[1]. It can be summed by the following formula:

Adverbial phrase = (\pm complement) \pm modifier +head

3.6.1 Simple adverbs

Simple adverbs act as the head in an adverb phrase and are divided into two categories derived adverbs and non-derived adverbs.

The derived adverbs are grouped into three subclasses

1. adverbs ending in -ari/अरि
2. adverbs ending in -sath/साथ and
3. adverbs ending in -purvaka/पुर्वक.

The derived adverbs consists of a stem (adjective, adverb or noun) and one of the following suffixes: -ari/अरि , -sath/साथ and -purvaka/पुर्वक . The suffix -ari/अरि occurs with the Nepali stems; the suffix -sath/साथ occurs with stems borrowed from Hindi and the suffix -purvaka/पुर्वक occurs with stems borrowed from Sanskrit.

3.6.1.1 Adverbs ending in -ari/अरि 'doing'

Adverbs ending in -ari/अरि 'in a manner' are derived from Nepali adjectives and adverbs. The underlying linear order consists of an adjective stem, or an adverb stem plus the adverb suffix -ari/अरि.

Nepali Adjective stem:	Derived adverbs in -ari
ramro/राम्रो 'good'	ramrati/राम्ररी 'in a good manner'
susta/सुस्त 'slow'	sustari/सुस्तरी 'slowly'
kasto/कस्तो 'how'	kastari/कस्तरी 'in what way'
jaso/जस्तो 'how(relative)'	jasari/जसरी 'in which way'

3.6.1.2 Adverbs ending in -sath/ साथ 'with'

They are the Hindi derived adjectives or nouns. The underlying linear order consists of a Hindi adjective or noun stem plus the adverb suffix -sath/साथ e.g.

Hindi noun, adj-stems	Derived adverbs in -sath
khusi/खुशी (adj.) 'happy'	khusisath/खुशीसाथ 'happily'
dikdari/दिक्दारी (n.) 'sadness'	dikdarisath/दिक्दारीसाथ 'sadly'

3.6.1.3 Adverbs ending in -purvaka/पूर्वक

The adverbs ending in -purvaka/पूर्वक are of Sanskrit origin and are derived from nouns. The underlying linear order consists of a Sanskrit noun root plus the adverb suffix -purvaka/पूर्वक, e.g.

Sanskrit noun stems:	Derived adverbs in -purvaka
utsaha/उत्साह 'enthusiasm'	utsahapurvaka/उत्साहपूर्वक 'enthusiastically'
ananda/आनन्द 'happiness'	anandapurvaka/आनन्दपूर्वक 'happily'

Different forms distinguishes the adverbials from the adverbs. In terms of distribution, ans they fill the same functional slots as the adverbs do. The list below presents a list of adverbials of most frequent occurrences[1].

aba/अब 'from now on',

ahile/अहिले 'now, at this time',

aghi/अधी 'before, previously',	ahile/अहिले 'now',
akhira/आखिर 'finally',	atyanta/अत्यन्त 'extremely',
barabar/बारबार 'frequently',	bhaepani/भनेपछी 'although',
bharkhar/भर्खर 'recently',	bitikai/वितीकै 'recently',
eklai/एकलै 'alone',	ekdam/एकदम् 'completely',
jahile/जहिले 'when',	jasari/जसरी 'in which way',
kaha/कहाँ 'where',	kasari/कसरी 'in which way?',
kahilekahi/कहिलेकाहीं 'sometimes',	kehi/केही 'somewhat',
paraspara/परस्पर 'mutually',	najikai/नाजीकै 'near',
pachiltira/पछिल्लिर 'behind',	pani/पनि 'also',
saberai/सबेरै 'early',	sadaiva/सदैब 'always',
talatira/तलतिर 'downward',	tyasari/त्यसरी 'in that way',
uhile/उहिले 'then',	vyarthai/व्यर्थै 'unnecessarily',
yastari/यस्तरी 'in such a way'.	

3.6.2 Compound adverbs

Compound adverbs are formed by the combinations of two adverbials. Hyphen indicates that they are they are compound adverbials, but in their transcribed from although there is no hyphen in their Devanagari orthography they are compound adverbs, like as shows in the examples below[1].

agadi-palti/अगाडी-पट्टि 'in front'	aghil-tira/अघिल्लिर 'in front'
amane-samne/आमने-सामने 'face to face'	bicabica-ma/विचविचमा 'intermittently'
mas-tira/मास्तिर 'upward'	mathi-tira/माथीतिर 'upward'
pari-patti/पारि-पट्टि 'on the other side'	tala-tira/तलतिर 'beneath'

3.7 Conjunctions

Conjunctions helps to conjoin two or more structures like word, phrases, or clauses. We can find two types of conjunctions, namely co-ordinating conjugations and subordinating conjugations. Any two equal structures are conjoined by the co-ordinating conjunction where as the subordinating conjunctions conjoins unequal structures like a clause dependant on a word, phrase or clause. The internal structure of the co-ordinating conjunctive structures is presented in the following formula[1]:

coordinating conjunctions = \pm head \pm head ... \pm head +C: +head

where +C is connecting slot and an obligatory head at the end. The optional heads may be

words, phrases, clauses or sentences.

3.7.1 Coordinating conjunctions

Any two equal structures, like phrases, words, clauses or sentences are conjoined by the coordinating conjunctions. Table below shows the Nepali coordinating conjunctions in alphabetical order.

Coordinating conjunctions	Words	Phrases	Clauses
ani/अनि 'and then'	-	-	+
athava/अथवा 'or'	+	+	+
ki/कि 'or'	+	+	+
ki...ki/कि कि 'either ... or'	+	+	+
kintu/किन्तु 'but not'	-	-	+
parantu/परन्तु 'but'	-	-	+
ra/र 'and'	+	+	+
taipani/तैपनि 'even then'	-	-	+
tara/तर 'but'	-	-	+
tatha/तथा 'and'	+	+	-
va/वा 'or'	+	+	-
ya/या 'or'	+	+	-

Table 3.5: Table showing Nepali coordinating conjunction and the elements they co-join[1]

From the table we can see that ani, naki, parantu, taipani and tara conjoin only clauses and sentences, and the coordinating conjunctions athava, ki and ra conjoin words, phrases and clauses. e.g.

ani/अनि 'and then':

ani ke bhayo/अनि के भयो ? 'And then what happened ?'.

ki...ki/कि ... कि 'either ... or':

ki tyo jancha, ki ma janchu/कि त्यो जान्छ, कि न जान्छु। 'either he goes, or I will go'.

3.7.2 Subordinating conjunctions

The subordinating conjunctions like aghi/अघि 'before', agadi/अगाडि 'before', bhane/भने 'if', pachi/पछि 'after' and pani/पनि 'although' occur at the end of the subordinate clause[2]. The subordinate clauses marked by these subordinating conjunctions occur before the principal clause. Subordinating conjunctions ki/कि 'that' and kinaki/किनकि 'because' occur at the beginning of the subordinating clause. The subordinating clause marked by these subordinating conjunctions occur after the principal clause. In Nepali the subordinate clauses are marked by subordinating conjunctions in two ways the formula for the subordinate clauses is written in the following two ways:

SubCl = +Axis: clause + Relator: aghi, agadi, bhane, pachi and pachi

SubCl = +Relator: ki and kinaki + Axis: clause

the following example illustrates the subordinating conjunctions in context:

daktar aunu aghi birami marisakeyko thiyo/डाक्टर आउनु अघि बिरामी मरिसकेको थियो[1]

'The patient had died before the doctors came'.

3.8 Verb Phrases

There are two forms of verb phrase, infinitive form and finite forms. The internal structure of the verb phrases is as below

verb-nonfinite = ±negative(na- ... -i kana) + root ± causative +voice +aspect

verb phrase-finite = ±prefix (±negative na-) + root ±causative +voice +mode +aspect +Aux:
suffixes (+person +number +gender +tense (±negative -na-))[1]

The non-finite forms are

1. infinitive marked by the infinitive suffix -na or -nu
2. participles marked by the suffixes -eko, -ne, -dai, -tai, -era, -i, -i kana
3. conditional marked by the suffix -e

1. infinite forms:

ja-na or ja-nu/जानु 'to go'

kha-na or kha-nu/खानु 'to eat'

gar-nu or kha-nu/ 'to do'

2. Participles forms:

gar-eko/गरेको '(perfect participle) done'

gar-ne/गर्ने '(imperfect participle) doing'

gar-dai/गर्दै '(conjunctive participle) doing'
done'

gar-era/गरेर '(absolutive participle) having

gar-i/गरि '(absolutive participle) having done'
having done'

gar-i kana/गरिकन '(absolutive participle)

3. Conditional forms

ga-e/गरे 'if-go'

kha-e/खाए 'if eat'

gar-e/गरे 'if do'

Nepali verb phrase in GF is defined as

```
VPH : Type = {  
  s      : VerbForm => {inf : Str};  
  obj    : {s : Str ; a : Agr};  
  subj   : VType ;  
  comp   : Agr => Str;  
  inf    : Str;  
  ad     : Str;  
  embComp : Str ;  
};
```

where

S is verb form, obj is object of the verb, subj is of VType and VType is defined as VType = VIntrans | VTrans | VTransPost ; comp, inf and ad are complement, infinitive form and adverb of the verb and finally embComp is the embedded component s.

3.8.1 Verbs as heads

The simple finite verb forms are the head of verb phrases. Thus we can formulate verb as:

Verb = ±Prefix +Stem ±Causative +Voice +Mode +Aspect +Suffix

3.8.2 Auxiliary verbs in the verb phrase

The auxiliary verbs in Nepali are: parnu/पर्नु 'should, must', hunu/हुनु 'be' and saknu/सक्नु 'can, may'. Auxiliary verb parnu/पर्नु 'should, must' is inflected for tense, but uninflected for aspect, person, number or gender, e.g.

parcha/पर्छ (present) 'should must'	paryo/पर्यो (simple past) 'had to'
parthyo/पर्थियो (habitual past) 'had to'	parecha/परेछ (unknown past) 'had to'
parla/पर्ला (future) 'will have to'	

The auxiliary verbs hunu/हुनु 'be' and saknu/सक्नु 'can, may' are inflected for aspect, person, number and gender. With the auxiliary verb hunu/हुनु 'be' the head of the VP carries the perfect participial suffix -eko for masculine singular, -eki for feminine singular and -eka for plural both cases. With the auxiliary saknu/सक्नु 'can, may' and parnu/पर्नु 'should, must' the head of the verb phrase is in the infinitive form[1].

3.8.3 The negative verb forms

Unlike most of the languages in Nepali negative verbs forms are formed at the morphological level[1]. To make the verb negative, morpheme na-/न- is prefixed to the imperative, infinite, conditional and participial forms or is suffixed to the verb stems elsewhere, depending upon the verbs contract with noun and tense it can take different inflectional case of negative. The negative prefix na-/न- : The negator na-/न- 'not' is prefixed to imperative, infinitive and participial forms, e.g.

Imperative:	
Positive	Negative
khanos/खानुस 'Please eat'	nakhanos/नखानुस 'Please don't eat'
jano/जानु 'Please go'	najanos/नजानुस 'Please do not go'
garnos/गर्नुस 'Please do it'	nagarnos/नर्नुस 'Please do not do it'

Table 3.6: Negative form of imperative verbs[1].

Infinitive:	
Positive	Negative
Jana/जान 'to go'	najana/नजान 'not to go'
khana/खान 'to eat'	nakhana/नखान 'not to eat'
garna/गर्न 'to do'	nagarna/नगर्न 'not to do'

Table 3.7: Negative form of inflective verbs[1].

Conditional forms:	
Positive	Negative
khane/खाने 'if eat'	nakhane/नखाने 'if not eat'
gare/गने 'if do'	nagarne/नगरे 'if not do'

Table 3.8: Negative form of conditional verbs[1].

3.8.3.1 The negative suffix na-/न-

The negative form in Nepali is not uniform, there are few cases which adds -na at the last of word to make it negative form e.g.

gardina/गर्दिन 'He does not do it'	khadina/खादिन 'He does not eat'
jadaina/जाँदिन 'He does not go'	khadaina/खादैनन् 'They do not eat it'
jadainan/जाँदैनन् 'They do not go'	

In the third person plural form, negative -na- is followed by the third person plural suffixes. The full range of the negative verb forms are exemplified by the conjugation of the verb garnu. In GF the negative form is morphologically inflected as shows in the code below, this is the negative form of present tense, non-perfective aspect, non-progressive mode.

```
mkVPreNPIReg : Str -> Str -> Polarity -> NPerson -> Number -> Gender -> {s:Str} =
  \root, root1, po, pn, n, g ->
  {s = case <po, pn, n, g> of {
    -- Negative Case
    <Neg, Pers1, Sg, _> => root + "VdinV" ; -- इँ (खाँदिँ)
```

<Neg, Pers1, Pl, _> => root + "VdEnEV" ; -- ऐनै (खाँदैनै)
 <Neg, Pers2_L, Sg, Masc> => root + "VdEnsx:" ; -- ऐनस् (आउँदैनस्)
 <Neg, Pers2_L, Sg, Fem> => root + "Vdinsx:" ; -- इनस् (खाँदिनस्)
 <Neg, Pers2_L, Pl, _> => root + "VdEnE" ; -- ऐनै (खाँदैनै)
 <Neg, Pers2_M, Sg, Fem> => root + "VdinE" ; -- इनै
 <Neg, Pers2_M, _, _> => root + "VdEnE" ; -- ऐनै
 <Neg, Pers3_L, Sg, Masc> => root + "VdEn" ; -- ऐन
 <Neg, Pers3_L, Sg, Fem> => root + "Vdin" ; -- इन (खाँदिन)
 <Neg, Pers3_L, Pl, _> => root + "VdEnx:n" ; -- ऐनन्
 <Neg, Pers3_M, Sg, Fem> => root + "Vdinx:n" ; -- इनन्
 <Neg, Pers3_M, _, _> => root + "VdEnx:n" ; -- ऐनन्
 <Neg, _, _, _> => root + "Vnuhunx:z=n" -- चुहुन्न
 }
 };

3.8.4 Modifiers in the verb phrase

Modifiers in verb phrase are either adverbs, adverbial or post-positional phrases, e.g.

Adverbs: sustari bhanin/सुस्तरी भनिन् 'she said faintly'

Adverbial phrases: eak eak gari here/एक एक गरि हेरे 'He examined one by one'

3.8.5 Transitive verbs and their complements

All transitive verbs(tv) occur with an obligatory direct object. The transitive verbs are sub-categorized as transitive verb-1, transitive verb-2, transitive verb-3 and transitive verb-4 on the basis of other obligatory complements they take besides the direct object.

Transitive verb-1(tv1) is a verb which occurs with an compulsory direct object, similarly transitive verbs-2(tv2) occurs which an obligatory dative complement, besides an obligatory direct object complement. Transitive verb-3(tv3) occurs with an obligatory direct object in accusative case and an obligatory object complement in the accusative case and tv4 occurs with compulsory direct object and an compulsory locative complement.

3.8.6 The equational verbs

The verbs which co-occur with subject complement(SC) are equational verbs[1]. Equational verbs in Nepali are hunu/हुनु 'to be', dekhinu/देखिनु 'appear', and lagnu/लाग्न 'feel'. The equational verb-1 hunu has two forms, the existential and identificational hunu.

3.8.6.1 Identificational hunu/हुनु case

The form of hunu/हुनु 'be' which identifies its subject shows the following inflections.

Pronouns	Past	Present	Future	
	Simple		Definitive	Indefinite
1st (sg.) ma	thie/थियँ	hu/हुँ	hunechu/हुनेछु	hula/हुँला
2nd (sg.) ta	thiyau/थियो	hos/होस	hunechas/हुनेछस्	holas/होलास्
3rd (sg) u	thiyo/थियो	ho/हो	hunecha/हुनेछ	hola/होला
1st (pl) hami	thiyau/थियो	hau/हौ	hunechau/हुनेछौ	houla/हौला
2nd (pl) timiharu	thiyau/थियो	hau/हौ	hunechau/हुनेछौ	houla/हौला
3rd (pl) uniharu	thie/थिए	hun/हुनु	hunechan/हुनेछन्	holan/होलान्

Table 3.9 : Inflection of identificationla 'hunu' case.

3.8.6.2 The existential hunu/हुनु case

The form of hunu/हुनु which indicates the more existence of its subject shows the following inflection:

Pronouns	Past	Present	Future	
	Simple		Definitive	Indefinite
1st (sg.) ma	thie/थियँ	chu/छु	hunechu/हुनेछु	hula/हुँला
2nd (sg.) ta	thiyau/थियो	chas/छस्	hunechas/हुनेछस्	holas/होलास्
3rd (sg) u	thiyo/थियो	cha/छ	hunecha/हुनेछ	hola/होला
1st (pl) hami	thiyau/थियो	chau/छौ	hunechau/हुनेछौ	houla/हौला

2nd (pl) timiharu	thiyau/थियौ	chau/छौ	hunechau/हुनेछौ	houla/हौला
3rd (pl) uniharu	thie/थिए	chan/छन्	hunechan/हुनेछन्	holan/होलान्

Table 3.10 : Inflection of existential 'hunu' case.

3.8.7 Transitive verbs and complements

Intransitive verbs are the type of verbs which do not stand in construction with direct object function or a subject complement function. Nepali has three types of intransitive verbs.

Intransitive verbs-1 (iv-1) are verbs which occur with no complements[2].

e.g. Johan laughed/जोन हाँस्यो

Intransitive verb-2 (iv-2) are verbs which occur with an obligatory dative complement[2]

e.g. Johan went home soon/ जोन चाँडै घर गयो।

Similarly intransitive verb-3 (iv-3) are verbs which occur with an obligatory locative complement[2] e.g. Johan lay in the bed/जोन खाटमा पल्टियो

3.9 Clauses

Clauses are the small grammatical unit that can express a complete proposition. Clauses of Nepali are defined in GF as shown below

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

where

VPHTense =

VPGenPres
 | VPSmplPast
 | VPFut
 | VPPerfPres
 | VPPerfPast
 | VPPerfFut
 | VPCondPres
 | VPCondPast;

Polarity is either positive or negative, and order is defined as Order = ODir | OQuest ;

The construction above shows that a clause takes parameters like *VPHTense*, polarity, order and forms a record of only one filled labeled 's'. One limitation in this implementation is that common API of GF only supports eight tenses, combination of four tenses (present, past, future and conditional) and two anteriorities Anter and Simul but Nepali tense system has some rich constructs like aspect and mode which makes tense cases slightly different than the GF's common resource API, so *VPHTense* has been defined and later it is mapped with the standard definition of GF's common resource API.

GF has lots of different abstract functions defined for the construction of clauses. One of the important function is *PredVP* : NP -> VP -> Cl ; which takes noun and verb phrase and constructs the required clause. The concrete syntax of *PredVP* for nepali looks like *PredVP* np vp = mkClause np vp ;. *PredVP* calls another function *mkClause*, this is a complex function which creates a clause with variable tense, order and polarity.

3.9.1 Imperative clauses

There are some special type of clauses in Nepali they are passive, imperative and question clauses. The GF resource grammar have a limited support for imperative sentences. Mainly the function *ImpVP* : VP -> Imp ; helps to form a imperative sentences it takes verb phrase and converts it in to imperative case. Basically, any verb phrases can be turned in to an imperative clause.

3.9.2 Question clauses

There are various ways we can form a question in GF, more than 14 abstract classes are available for the construction of question clauses. However, *QuestCl* and *QuestVP* are the main functions that comes frequently while constructing a question clause. *QuestCl* is responsible for creating yes/no question and *QuestVP* is responsible for creating wh-questions.

The concrete syntax of *QuestCl* for Nepali looks like shown below,

```
QuestCl cl = {
  s = \t,p,qf => case qf of {
    QDir   => cl.s ! t ! p ! OQuest;
    QIndir => "येदि" ++ cl.s ! t ! p ! ODir
  }
};
```

The function above create clauses which do not have fix tense and polarity. If it is indirect case then the term "येदि" is added at the beginning of the clause.

ho/hoina हो/होइन 'yea/no' question has the same syntactic or grammatical structure as the declarative sentence but is differentiated by the shift in intonation, e.g. Is Johan going to school/जोन स्कूल जाँदैहो ?

3.10 Sentential Structure

Sentence is the highest grammatical unit and as such it is not a constituent of any higher level grammatical structure. The internal structure of the sentence consists of a segmental constituent and a prosodic constituent[1] i.e.

+ Segmental constituent

S =

+ Prosodic constituent

The segmental constituent is a sentence is filled primarily by a clause or clauses. The prosodic constituent consists of one of the three intonation patterns. The intonation pattern of a statement and imperative statement, and the intonation pattern of ho/hoina questions 'yes/no questions' and the intonation pattern of K-question 'wh- question'.

There are many functions to construct sentence in GF, the most common one being the *UseCl* function. The definition of *UseCl* looks like

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

It takes tense, anteriority, polarity and a clause as input and generates a sentence as output.

The concrete syntax of *UseCl* for Nepali is implemented as shown below

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 ! VPSmplPast ! 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 ! VPCondPres ! p.p ! ODir;

```

    <Cond, Anter> => temp.s ++ p.s ++ cl.s ! VPCondPast ! p.p ! ODir
  };
};

```

Similarly we have functions like UseQCl : Temp -> Pol -> QCl -> QS ; and UseRCl : Temp -> Pol -> RCl -> RS ; to handle the case of question and relative sentences, these also belong to the category of frequently used functions for the construction of sentences.

3.10.1 Clauses as segmental constituents

We have covered the full clauses as segmental constituents in previous sections. A clause with finite verb or number of clauses with finite or non-finite verb embedded in the principal clause constitute a sentence.

4. Evaluation and Results

Nepali resource grammar implements all the abstract syntax provided by the Grammatical Framework. It has 194 standard syntax functions and few new functions specially defined to handle the special cases of Nepali language. This covers almost every aspect of the language but there are few corners which are not currently covered by resource grammar. These parts requires some theoretical foundation to implement them successfully in computational form, which is left as a possible future work.

The synopsis page³ of Grammatical Framework lists all the API that GF application programmers can use. It displays all the standard and language specific functions with examples. These examples are translated in every languages that are available in Grammatical Framework, one way to test correctness of resource grammar is to check thoroughly whether these examples are translated correctly or not. If all examples are correctly translated then we can be sure that the rules covered by the resource grammar are correct. Of course to verify the accuracy we need someone who understands the given language plus at least one stable language that has been implemented in GF, for our case English was taken as a reference language to test the accuracy of Nepali language.

For the evaluation of our resource grammar feedback from the native speakers of Nepali language was taken, on the first feedback we discovered errors mostly related to syntactic conjunction and typos. The link to synopsis page has been send to three native speakers of Nepali and in total roughly 40 errors were reported out of around 400 synopsis examples, some of them were just typo-graphical errors and few of them were serious issues like error in word formation as we translate from one language to another. All the reported errors were fixed in the second iterations, now except the few known issues resource grammar produces accurate result. There are still a few issues in indirect question clauses and few verb cases which will require bigger change in the resource grammar.

5. Further work

This implementation covers all the basic functions and covers almost all part of the Nepali language but still some research is needed to implement some of the complex part of the language like idioms, and also there are some expressions which can be expressed in many different ways in different contexts which need more theoretical foundation to implement these things in GF. Another possible work is to expand Nepali vocabulary in GF, currently we have common 450 words, these words has been chosen to cover wide variety of examples as

3 <http://www.grammaticalframework.org/lib/doc/synopsis.html>

possible but we can expand this to cover full vocabulary of Nepali which also helps to check the accuracy of Nepali resource grammar.

6. Conclusion

It has been a great experience to work with the Grammatical Framework and Nepali language. Effort has been made to cover a wide variety of structures in Nepali language, but the current resource grammar doesn't cover every aspect of the language. This work provides useful information about the Nepali computational grammar and the current implementation can be used in the multilingual applications where well defined text is available. Despite some of the limitations of the system, GF can be used successfully in the applications where you have full control of your text, like software localization, dialogue systems, translation of your web pages etc. As all the resource grammars share a common syntax, translation in other languages is automatic this makes very easy to build multilingual systems.

7. Important Terms

Conjunction: Conjunction is a word that joins two or more parts of a sentence.

Equational verb: Verb which occurs with subject complement.

Inflection: Modification of a word to express different category of grammar, eg cats is the inflection of cat, and its the inflection case for number

Inflection table: Table formed by containing all the inflected form of a term

Interjection: It is a word added to convey emotion, interjections are not grammatically related to any other part of the sentence.

Intonation: Intonation is variation of pitch while speaking which is not used to distinguish words

Linearization: Converting tree into string.

Morphology: Morphology is a branch of linguists that study the words, their formation and internal structures.

Multilingual: Being able to express in more than one language.

Non-perfective aspect: Non-perfective aspect is an aspect that expresses an event or state, with respect to its internal structure, instead of expressing it as a simple whole[11].

Parsing: Converting string into an abstract tree (finding a tree that produce given string).

Perfective aspect: Perfective aspect is an aspect that expresses a temporal view of an event or state as a simple whole[10].

Preposition: A preposition links nouns, pronouns or phrases to other words in a sentence.

Postpositions: The placing of word or a suffixed element after the given word.

8. Appendix

Here are examples of some of the inflection tables formed by grammatical framework. As verb in Nepali has complex structure and inflects for many cases and inflection table becomes very large so only the fragments of table has been included here.

Nouns:

```
{s = table Number [table Case ["आईमाई"; "आईमाईलाई"; "आईमाईले";  
                                "आईमाईलाई"; "आईमाईबाट"; "आईमाईमा"];  
table Case ["आईमाईहरु"; "आईमाईहरुलाई"; "आईमाईहरुले";  
            "आईमाईहरुलाई"; "आईमाईहरुबाट"; "आईमाईहरुमा"]];
```

g = Fem; h = Pers3_M; t = Living; lock_N : {} = <>}

Table 8.1: Inflection of Noun woman

Verbs:

```
{s = table VerbForm ["जान्छु"; "जान्छु"; "जान्छौं"; "जान्छौं"; "जान्छस्"; "जान्छेस्";  
                    "जान्छौ"; "जान्छौ"; "जान्छौ"; "जान्छौ"; "जान्छौ"; "जान्छ्यौ";  
                    "जानुहुन्छ"; "जानुहुन्छ"; "जानुहुन्छ"; "जानुहुन्छ"; "जान्छ";  
                    "जान्छे"; "जान्छन्"; "जान्छन्"; "जान्छन्"; "जान्छिन्"; "जान्छन्";  
                    "जान्छन्"; "जानुहुन्छ"; "जानुहुन्छ"; "जानुहुन्छ"; "जानुहुन्छ"; "जानुहुन्छ"]; }
```

Table 8.2: Inflection of verb 'go', for present tense, non-perfective aspect, non-progressive mode, positive form

```
{s = table VerbForm [ "जाँदिनँ"; "जाँदिनँ"; "जाँदैनेँ"; "जाँदैनेँ"; "जाँदैनेस्"; "जाँदिनेस्";  
                    "जाँदैने"; "जाँदैने"; "जाँदैने"; "जाँदिने"; "जाँदैने"; "जाँदैने";  
                    "जाँनुहुन्न"; "जाँनुहुन्न"; "जाँनुहुन्न"; "जाँनुहुन्न"; "जाँदैने";  
                    "जाँदिने"; "जाँदैने"; "जाँदैने"; "जाँदैने"; "जाँदिने"; "जाँदैने";  
                    "जाँदैने"; "जाँनुहुन्न"; "जाँनुहुन्न"; "जाँनुहुन्न"; "जाँनुहुन्न"; "जाँनुहुन्न"]; }
```

Table 8.3: Inflection of verb 'go', for present tense, non-perfective aspect, non-

progressive mode, negative form

9. References

- [1] Jayaraj Acharya, "*A descriptive grammar of Nepali And An Analyzed Corpus*", Georgetown University Press, 1991
- [2] Dr. Hemangaraj Adhikari, "*Sama Samayik Nepali Vyakaran*" fifth edition, Bidarthi pustak bhandar, Kathmandu, 2066 B.S.
- [3] Ranta A. , "*Grammatical Framework : A Type Theoretical Grammar Formalism*", The Journal of Functional Programming, 14(2), 145- 189 , 2004
- [4] Aarne Ranta, "*Grammatical Framework: Programming with Multilingual Grammars*", CSLI Publications, 2011
- [5] Shambhu Oja , Banu Oja , Mark Turin , Elisabeth Uphoff , "*Nepali-English & English-Nepali Glossary*", second edition, Madan Puraskâr Pustakâlaya library , 2004 URL: http://www.nepalresearch.org/dictionaries/nep_eng/nep_eng.pdf
- [6] Bal Krishna Bal , "*Structure of Nepali Grammar*", PAN Localization, Working Papers 2004-2007, pp. 332-96.
- [7] Christopher Jay Manders, "*A Foundation in Nepali Grammar*", AuthorHouse, 2007
- [8] Spoken languages in the world, <http://www.infoplease.com/askeds/many-spoken-languages.html>
- [9] Ranta A., "*The GF Resource Grammar Library. Linguistic Issues in Language Technology*", CSLI Publications, Volume 2, Issue 2, 2009
- [10] Glossary of linguistic terms
<http://www.sil.org/linguistics/GlossaryOfLinguisticTerms/WhatIsPerfectiveAspect.htm>
- [11] Glossary of linguistic terms
<http://www.sil.org/linguistics/GlossaryOfLinguisticTerms/WhatIsImperfectiveAspect.htm>