Swahili Inflectional Morphology for the Grammatical Framework

Wanjiku Ng'ang'a
School of Computing & Informatics, University of Nairobi, Kenya
wanjiku.nganga@uonbi.ac.ke

Abstract

Grammatical Framework is a grammar formalism based on type theory and implemented in Haskell, that utilizes the interlingua approach to multilingual translation. Multilingualism is achieved by defining resource grammar libraries for each individual language within the framework. Here, we present the definition of the inflectional morphology grammars for Swahili, as part of the Swahili resource grammar library within the Grammatical Framework. In this framework, morphological constructs are implemented using the functional morphology approach which requires definition of linguistic data types and morphological functions.

1 Introduction

Grammatical Framework (GF) (Ranta, 2004) is a grammar formalism based on type theory. It has been designed to handle several languages in parallel. Its main feature is the separation of abstract and concrete syntax, which makes it very suitable for writing multilingual grammars. The abstract part of a grammar defines a set of abstract syntactic structures, called abstract terms or trees, while the concrete part defines a relation between abstract structures and concrete structures. Multilingual grammars formalize the notion that different languages share the same grammatical categories (e.g. noun phrases, verb phrases etc) and syntax rules (e.g. nominalization and predication). An abstract syntax in GF deals with language-independent (pure) tree structures while the language-dependent concrete syntax specifies how these trees are mapped into different languages. A multilingual GF grammar is therefore realised as a combination of an abstract syntax which can be mapped to a number of language-dependent concrete syntaxes, thereby achieving multilingualism.

To achieve multilingualism, GF uses a Resource Grammar (RG) library (Ranta, 2009), which is essentially a set of parallel grammars for different languages. The grammar defines, for each language, a complete set of morphological paradigms and a syntax fragment. Currently, the library covers sixteen languages: Bulgarian, Catalan, Danish, Dutch, English, Finnish, French, German, Italian, Norwegian (bokmål), Polish, Romanian, Russian, Spanish, Swedish and Urdu. Grammars for other languages, including Swahili, are under development. The main purpose of this grammar library is to allow application developers (non-linguist programmers) to write domain-specific grammars to support varied, linguistically correct multilingual applications. This is in line with modern software engineering practice where software libraries,comprising of a number of specially written routines are used as 'helper' code that provide services to some other independent programs. GF's multilingual grammar library provides a reusable interface to the thousands of details involved in morphology, lexicon, inflection and syntax, facilitating the easy development of multilingual natural language engineering applications such as machine translation, multilingual generation, natural language interfaces, spoken dialogue systems etc. To date, GF has been used to build a wide range of applications such as an authoring and documentation system for the KeY software specification system (Burke and Johannisson, 2005), WebALT's mathematical exercise translator (Caprotti, 2006), and the TALK project on spoken dialogue systems (Ljunglöf et al., 2008), to list a few.

A GF Resource Grammar comprises of two parts: morphological paradigms and a syntax fragment. This paper describes the development of Swahili morphological paradigms as a first step in creating a Swahili GF Resource Grammar. This work represents the first attempt to extend GF with a Bantu language. Section 2 gives a brief summary of the Swahili language with emphasis on morphology, while the definition of the language-dependent modules for Swahili morphology in GF is covered in Section 3.

2 Swahili

Swahili is a Bantu language belonging to the Niger-Congo family. It is a highly inflecting language where both prefixed and suffixed morphemes play an important grammatical role. The functions of prefixes are particularly important in both nominal and verbal morphology. In the case of nouns, as is typical with Bantu languages, each noun belongs to a noun class which is signaled by a pair of prefixes attached to the nominal stem, denoting singular and plural forms. In addition, some nouns take an optional locative suffix e.g. *nyumba-ni* which means 'in the house', is obtained by adding the *-ni* locational suffix to the root *nyumba*. Verbs have an agglutinating structure where a system of affixes is used to mark various grammatical relations, such as subject, object, tense, aspect, and mood. There is a system of concordial agreement in which nouns must agree with the main verb of the sentence in class (and animacy) and number. Adjectives, possessive pronouns and demonstratives also agree in class and number with the noun they modify.

Swahili has a fairly fixed word order (SVO) at the sentence level, where the subject precedes the verb and the object, while within constituent phrases, modifiers succeed the head. Therefore adjectives, pronouns, determiners etc., follow the nouns they modify while adverbs come after the verb. Concrete grammars defined for Swahili should therefore capture these rich morphological and grammatical features of the language, to ensure generation of grammatically correct Swahili sentences.

3 Inflectional Morphology Paradigms for Swahili

Hurskainen (1992) has defined Swahili morphology using the two-level formalism, within the Swahili Language Manager (SALAMA) computational suite (Hurskainen, 1999). In contrast to finite-state approaches to morphology, GF uses a functional morphology approach to defining inflectional morphological paradigms. Rather than working with untyped regular expressions which is the state of the art of morphology in computational linguistics, functional morphology defines finite functions over hereditarily finite algebraic data types (Forsberg and Ranta, 2005). The definitions of these data types and functions form the language-dependent part of the morphology, while the language independent part consists of an untyped dictionary format which is used for synthesis of word forms. These language-dependent data types and functions are organized into four grammar files: Res, Morpho, Cat and Paradigms. Construction of lexical entries relies on the language-dependent definitions, and the language-independent Lexicon file. In this paper, we describe the definition of Res, Morpho, Cat, Paradigms and Lexicon, for Swahili.

3.1 ResSwa

Res is a language-specific resource grammar that defines parameter types specific to a given language. These types denote linguistic features that cut across the language. For nouns, the following parameters are defined:

```
Gender = g1_2 | g3_4 | g5_6 | g5a_6 | g6 |
         g7_8 | g9_10 | g11 | g11_6 | g11_10 ;

Case = Nom | Loc ;

Animacy = AN | IN ;
```

The parameter Gender defines Swahili's noun classes where g1_2 refers to noun class *m_wa* where the singular form begins with prefix *m-* and the plural with *wa-* e.g. *m-sichana* (girl) and *wa-sichana*. (girls). The parameter Case defines two options – nominative and locative to handle cases where some nouns optionally take a locational suffix as explained in section 2. The parameter Animacy, with values animate and inanimate, is defined to ensure correct subject-verb agreement for animate nouns that do not fall in the typical noun class *m-wa* for animates, but whose agreement features must match those of animate nouns. Examples of such nouns that do not fall in the typical animate class *m-wa,* include names of most animals as well as family relations e.g. *mama* 'mother', *baba* 'father', *dada* 'sister', and *ndugu* 'brother'. Another important parameter is Number (singular and plural) which is common to other languages and is therefore pre-defined elsewhere in the resource grammar library. In this work, we adopt the Swahili noun classes as motivated by Moxley (1998).

For verbs, the following parameters are defined:

```
VForm =
  Vinf |
  Vimper Number Person |
  VPres Number Gender Animacy Person |
  VPast Number Gender Animacy Person |
  VFut Number Gender Animacy Person;
```

The parameter Vform defines five different forms that a verb can take. The first form is the Infinitive form. Next, is the Imperative form which is dependent on Number and Person since this form is only applicable to second person singular (2SG) or plural (2PL) e.g. for the verb *cheza* 'to play', the imperative form for 2SG is *cheza* while that for 2PL is *chezeni*. The other three forms define Present, Past and Future tense, which are all dependent on Number, Gender, Animacy and Person, as shown by the examples in Table 1.

Gen der	**An-imacy**	**Num-ber**	**Per-son**	**Pre-fix (Pres ent)**	**Pre-fix (Past)**	**Pre-fix (Fu-ture)**
1_2	AN	SG	3	ana	ali	ata
1_2	AN	PL	3	wana	wali	wata
9_10	AN	SG	3	ana	ali	ata
9_10	IN	SG	3	ina	ili	ita

Table 1: Verbs

The parameter AForm has been defined for Adjectives:

AForm = AF Number Gender Animacy | AA ;

AForm is defined using two operations, AF and AA. AF is applicable for adjectives that agree with the number, gender and animacy of the modified noun, while AA is a defunct operation that can be used to handle exceptions. The Spatial parameter has been defined to distinguish between the proximal, distal and referential demonstratives e.g. *hii* 'this', *hiyo* 'that' and *ile* 'that' (distal), and has been defined as follows:

Spatial = SpHrObj | SpHr | HrObj ;

3.2 MorphoSwa

Morpho contains language-specific resource grammar constructs that define exactly how various inflectional morphological paradigms are realized. Morpho uses the parameters defined in Res to realize the categories defined in Cat, as detailed in section 3.3. The definitions contained in Morpho are typically functional (constructionist) which define how to inflect stems to obtain different word forms e.g Nouns, Verbs, Adjectives, demonstratives and pronouns, and are accessed via function calls from the Paradigms grammar. Following is a GF code fragment that defines how to construct adjectives.

```
MkAdjective : Str -> Adj = \zuri ->   {
 s = table {
    AF n g anim => case Predef.take 1 zuri of {
          "a"|"e"|"i"|"o"|"u"  => VowelAdjprefix n g
           anim + zuri;
    _ => ConsonantAdjprefix n g anim + zuri
            };
     AA => zuri
      }
 } ;
```

The above code defines a function mkAdjective which takes an adjective root e.g. *zuri* 'good' or *eupe* 'white' which is of type string (Str) and creates an Adjective of type Adj. If the adjectival root begins with a vowel, a semi-vowel or a consonant must be inserted at the end of the prefix. To enforce this requirement, the code defines two helper functions VowelAdjprefix and ConsonantAdjprefix to handle the two cases separately. Since the adjective prefix is dependent on the AForm parameter as described in section 3.1, each of the helper functions take the number (n), gender (g) and animacy (anim) of the modified noun as input to determine the adjective prefix and prepends the prefix to the adjectival root, thus forming the adjective. Table 1 shows how mkAdjective creates adjectives that adhere to Swahili morphological rules as described.

Gen der	**Anim-acy**	**Num-ber**	**Vowel root = *zuri***	**Vowel root = *eupe***
1_2	Anim-ate	SG	m-zuri	M-eupe => mw-eupe
9_10	Inan-imate	SG	n-zuri	n-eupe => ny-eupe

Table 2: Adjectives

The following code fragment shows the definition of mkPronoun that takes Number and Person as input and generates a Pronoun:

```
   mkPronoun : Number -> Person-> Str = \n,p ->
 case <n,p> of {
          <Sg,P1> => "mimi" ;
          <Sg,P2> => "wewe" ;
          <Sg,P3> => "yeye" ;
          <Pl,P1> => "sisi" ;
          <Pl,P2> => "nyinyi" ;
          <Pl,P3> => "wao"   };
```

3.3 CatSwa

The Cat grammar defines all the lexical categories (closed, open and phrasal) that occur in language, and most of these are common to all languages. The Swahili concrete syntax file, CatSwa, currently defines the type specifications for the common base categories as they present in Swahili, as shown in Table 3.

Cat	Type	Example
N	Common Noun	*Msichana* 'Girl'
N2	Relational Noun	*Ndugu ya ..* 'Brother of ..'
Pron	Personal Pronoun	*Mimi* 'I'
V	One-place Verb	*Nitaimba* 'I will sing'
A	One-place Adjective	*Mzuri* 'Good'
Quant	Quantifier (Nucleus of Determiner)	*Hii/Hizi* 'This/These'
Prep	Preposition	*Ya* 'Of'
Num	Number	*Nne* 'Four'

Table 3: Lexical Categories

3.4 ParadigmsSwa

Paradigms defines the top level morphological functions that are used to construct lexical entries in the lexicon file, LexiconSwa. The functions correspond to the base lexical categories defined in CatSwa as shown in Table 3. Swahili Nouns are regular in the sense that given a noun root, gender, number and animacy, it is possible to use rules to generate a correct noun form. Hence, the abstract function regN is used to define common nouns of type N. In this definition, Str refers to the noun root which is passed to the regN function together with the values for Gender and Animacy:

regN : Str -> Gender -> Animacy -> N ;

The concrete form of regN calls the helper function mkNomReg that is defined in MorphoSwa to generate the final noun form. mkNomReg abstracts over the pre-defined Number parameter to generate both singular and plural forms of the noun root, and hence Number need not be passed directly to regN. Nouns of type N2 which take a Noun of type N followed by a preposition, are constructed by the function mkN2:

mkN2 : N -> Prep -> N2 ;

Pronouns are defined by the function mkPron:

mkPron : Number -> Person -> Pron ;

The category Verb (V) is a simple form and is defined by the function regV which makes reference to the parameter VForm as described in section 3.1, to generate the infinitive, imperative, present, past and future tense forms for any regular verb. :

regV : Str -> V ;

Adjectives are constructed by the abstract function regA which also uses the type AForm to generate adjectives that conform to concordial agreement with the noun they modify:

regA : Str -> A ;

Prepositions are constructed by the abstract function mkPrep whose definitions is:

mkPrep : Str -> Prep ;

Quantifiers are constructed via the helper function mkQuant defined in MorphoSwa.

mkQuant : Spatial -> Number -> Gender ->
Animacy -> Case -> Person -> Str;

mkQuant takes as input the Spatial parameter that specifies whether to construct a proximal, distal or referential demonstrative. In addition, the number, gender and animacy values have to be specified since Swahili quantifiers must agree in number and gender with the modified noun, as shown by the examples in Table 4.

Gender	Animacy	Number	Proximal	Distal	Referential
1_2	AN	SG	Huyu	Huyo	Yule
1_2	AN	PL	Hawa	Hao	Wale
3_4	IN	SG	Huu	Huo	Ule
3_4	IN	PL	Hii	Hiyo	Ile
7_8	IN	PL	Hivi	Hivyo	Vile
7_8	AN	PL	Hawa	Hao	Wale

Table 4: Swahili Quantifiers

3.5 LexiconSwa

Lexicon is part of the top-level grammar in GF. This grammar defines and gives access to content words that can then be used by the syntax component of the resource grammar library. LexiconSwa uses the functions defined in ParadigmsSwa to define lexical entries (words) that conform to Swahili morphology. The content words defined here must be of the types defined in CatSwa. Currently, LexiconSwa contains 83 content words out of a total 300 words defined in the corresponding language-independent lexicon (abstract) file, Lexicon. Table 5 shows example lexical definitions from LexiconSwa, while Table 6 shows the corresponding definitions in the English lexicon, LexiconEng. The abstract function e.g. country_N is defined in the abstract grammar Lexicon while LexiconSwa defines the corresponding concrete syntax. For example, *country_N* is the abstract function defined in Lexicon, while the definition: *country_N = regN “nchi” e_e inanimate*; is its corresponding concrete linearization in Swahili. This definition states that to form the Swahili form for the noun 'Country', the function regN defined in ParadigmsSwa is called by LexiconSwa with the Swahili string for country, *nchi*, followed by the gender (e_e in this case) and animacy value (inanimate). The function regN then inflects the root appropriately to construct the singular and plural forms for country_N in Swahili.

Cat	Definition in LexiconSwa (Swahili)
N	country_N = regN "nchi" e_e inanimate ;
N	cousin_N = regN "binamu" e_ma animate;
N	man_N = regN "mwanaume" m_wa animate ;
N	tree_N = regN "mti" m_mi inanimate ;
N	water_N = regN "maji" ma_ma inanimate ;
V	swim_V = regV "ogelea";
A	dirty_A = regA "chafu" ;
N2	father_N2 = mkN2 (regN "baba" e_e animate) (mkPrep "ya") ;
Quant	this_Quant = {s = \\n,g,anim,c => mkQuant SpHrObj n g anim Nom P3} ;

Table 5: Swahili Lexicon Entries

Cat	Definition in LexiconEng (English)
N	country_N = regN "country" ;
N	cousin_N = mkN human (regN "cousin") ;
N	man_N = mkN masculine (mk2N "man" "men") ;
N	tree_N = regN "tree" ;
N	water_N = regN "water" ;
V	swim_V = IrregEng.swim_V ;
A	dirty_A = regADeg "dirty" ;
N2	father_N2 = mkN2 (mkN masculine (mkN "father")) (mkPrep "of") ;
Quant	this_Quant = mkQuant "this" "these" ;

Table 6: English Lexicon Entries

LexiconSwa and LexiconEng clearly demonstrate how GF achieves multilingualism by allowing languages to share an abstract syntax, but define language-dependent features in the concrete grammars. Table 7 shows example translations at the lexical level that have been generated automatically within GF.

Swahili	Abstract Function	English
hawa	this_Quant	these
ndugu	brother_N2	brother/brothers
kiti	chair_N	chair
rafiki	rafiki_N	friend
marafiki	rafiki_N	friends
mwanaume	man_N	man
wanaume	man_N	men
mti	tree_N	tree
miti	tree_N	trees
vinalala	sleep_V	sleeping
wanawaza	think_V	thinking
nilitembea	walk_V	walked
watatembea	walk_V	walk
tulitembea	walk_V	walked
mrembo	beatiful_A	beautiful
warembo	beautiful_A	beautiful
kirembo	beautiful_A	beautiful
virembo	beautiful_A	beautiful

Table 7: Parsing and Generation Examples

4 Conclusion

In this paper, we have described the first attempt to extend the morphological component of the Grammatical Framework with a Bantu language, Swahili. We have described the data types and corresponding functions that implement Swahili inflectional morphology following the functional morphology methodology. For the 83 content words that have been defined in the lexicon, it is possible to generate translations to the other 16 languages currently implemented in GF, and vice-versa. Subsequent development work will focus on defining all possible lexical categories in CatSwa, following the abstract grammar, Cat. We will then define more inflectional morphology functions to support the new category additions. Once the morphology paradigms are completed, we will define the core syntax grammars for Swahili, thereby facilitating the translation of full sentences and utterances from Swahili into all the other GF languages. This work will contribute greatly to the development of domain-specific HLTD applications that require localization and customization for a Swahili-speaking audience as posited by Ng'ang'a (2006). We also envisage generalizing the Swahili grammars to cater for a wide range of Bantu languages, by adopting the language family modular structure within GF that allows Romance languages to share a whole lot of data types and functions, and define only language-dependent exceptions separately.

Acknowledgments

We acknowledge the input of Prof. Aarne Ranta who created GF and Krasimir Angelov, both of Chalmers University, Sweden, Juliet Mutahi and Kimani Njogu.

References

D. A. Burke and K. Johannisson. 2005. Translating Formal Software Specifications to Natural Language. A Grammar-Based Approach. In P. Blache, E. Stabler, J. Busquets and R. Moot (eds), Logical Aspects of Computational Linguistics (LACL 2005), Springer LNAI 3402, 51-66.

Olga Caprotti. 2006. WebALT! Deliver Mathematics Everywhere. In C. Crawford et al. (Eds.), Proceedings of Society for Information Technology & Teacher Education International Conference. 2164-2168).

Jeri L. Moxely. Semantic Structure of Swahili Noun classes. In I. Maddieson and T. Hinnebusch (eds), Language History and Linguistic Description in Africa: Trends in African Linguistics (2), Africa World Press.

M. Forsberg and A. Ranta. 2005. Functional Morphology: Tool Demonstration. *FSMNLP 2005*, Springer LNCS 4002, 304-305.

Arvi Hurskainen. 1992. A two-level computer formalism for the analysis of Bantu morphology: An application to Swahili. *Nordic Journal of African Studies,* 1(1):87-122.

Arvi Hurskainen. 1999. SALAMA: Swahili Language Manager. *Nordic Journal of African Studies,* 8(2):139-157.

Peter Ljunglöf and Staffan Larsson. 2008. A grammar formalism for specifying ISU-based dialogue systems. In B. Nordström and A. Ranta (eds), Advances in Natural Language Processing (GoTAL 2008), LNCS/LNAI 5221, Springer.

Aarne Ranta. 2004. A Type-Theoretical Grammar Formalism. *The Journal of Functional Programming,* 14(2):145–189.

Aarne Ranta. 2009. The GF Resource Grammar Library. Linguistic Issues in Language Technology, 2(2).

Wanjiku Ng'ang'a. 2006. Multilingual content development for eLearning in Africa. eLearning Africa: 1st Pan-African Conference on ICT for Development, Education and Training. Addis Ababa, Ethiopia.