A Finite-State Morphological Analyzer for Ge’ez Verbs

By

ELLENI ASCHALEW ZELEKE

Submitted in accordance with the requirements for the degree of

MASTER OF SCIENCE

In the subject of

COMPUTING

At the

UNIVERSITY OF SOUTH AFRICA

SUPERVISOR:	Prof. Ernest Mnkandla
CO-SUPERVISOR:	Prof. Sirgiw Gelaw Eggigu

JANUARY 2023

DECLARATION

Name: ELLENI ASCHALEW ZELEKE

Student number: 43698247

Degree: Master of Science (Computing)

Exact wording of the title of the dissertation as appearing on the electronic copy submitted for examination:

A Finite-State Morphological Analyzer for Ge’ez Verbs

I declare that the above dissertation is my own work and that all the sources that I have used or quoted have been indicated and acknowledged by means of complete references.

I further declare that I submitted the dissertation to originality checking software and that it falls within the accepted requirements for originality.

I further declare that I have not previously submitted this work, or part of it, for examination at Unisa for another qualification or at any other higher education institution.

February 3, 2023

SIGNATURE DATE

SUMMARY

A morphological analyzer is a valuable and necessary component in many natural language processing applications such as machine translation, automatic dictionaries, spell checking, speech recognition etc. Morphological analyzer is specifically important for Semitic languages with high inflection and productive verbs. Ge'ez is one of the ancient Semitic languages in the world. It is currently the liturgical language of the Ethiopian Orthodox Tewahido Church.

In this study, a finite-state morphological analyzer for the Ge'ez verbs was developed. The morphological analyzer was developed using the freely available finite-state tool Foma. The Ge'ez morphological analyzer was developed to analyze Ge'ez verbs into their root and feature tags. The analyzer also performs the generation of Ge'ez verbs from a given root and feature tags.

The Ge'ez morphological analyzer was tested using manually annotated verbs by Ge'ez experts from the Ethiopic New Testaments Ge'ez Bible (books of Matthew, Luke, Mark and John) and Ge'ez prayer book - ውዳሴ ማርያም -'*wudase maryam*'. The output of the Ge'ez morphological analyzer was compared with the manually annotated test data for accuracy. The result showed that the Ge'ez morphological analyzer analyzed the Ge'ez verbs with an accuracy of 97.29% and a precision of 80.24%. All in all, this research work achieved its objective by designing and implementing a Ge'ez verb morphological analyzer that performs both analysis and generation of Ge'ez verbs. The developed Ge'ez morphological analyzer will benefit the Ethiopian Orhodox Tewahido Church, interested Ge'ez language learners and the academic community that conduct researches in Ge'ez language.

KEYWORDS

The Ge'ez Language; Ge'ez verbs; Ge'ez morphology; morphology; computational morphology; morphological analyzer; morphological analysis and generation; finite-state morphology; finite-state transducer; gold standard

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

CV	Consonant vowel
FSA	Finite-state automata
FST	Finite-state transducer
IPA	International Phonetic Alphabet
NLP	Natural language processing
POS	Parts of speech
1PSGs	First person singular subject
1PPLs	First person plural subject
2PSGMs	Second person singular male subject
2PSGFs	Second person singular female subject
2PPLMs	Second person plural male subject
2PPLFs	Second person plural female subject
3PSGMs	Third person singular male subject
3PSGFs	Third person singular female subject
3PPLMs	Third person plural male subject
3PPLFs	Third person plural female subject
1PSGo	First person singular object
1PPLo	First person plural object
2PSGMo	Second person singular male object
2PSGFo	Second person singular female object
2PPLMo	Second person plural male object
2PPLFo	Second person plural female object
3PSGMo	Third person singular male object
3PSGFo	Third person singular female object

3PPLMo	Third person plural male object
3PPLFo	Third person plural female object
BASE	Base
CAUS	Causative
CAUSRECP	Causative reciprocal
NEG	Negation
RECIP	Reciprocal
REFLX	Reflexive
SERA	System for Ethiopia Representation in ASCII
VGRD	Gerundive verb
VIND	Indicative verb
VINF	Infinitive verb
VJUS	Jussive verb
VPER	Perfective verb
VSUBJ	Subjunctive verb

Chapter 1 – Introduction

1.1 Introduction

This document describes a research study conducted on modeling Ge’ez verbal morphology using finite-state methods. Morphology deals with the study of words and word structure and describes how words are created from morphemes – the smallest unit of a language that carries meaning or grammatical explanation.

Natural language processing (NLP) or human language processing is a field that aims to use the computer to perform important tasks involving human language such as human-machine communication, human-human communication or processing of text and speech (Trost, 2003). One of the major challenges in NLP is understanding natural language so that computers can derive meaning from human language input and generate a natural language. A morphological analyzer is one of the core components of NLP applications as it provides valuable information about the word’s internal structure – the root and its grammatical properties.

Ge’ez is one of the ancient Semitic languages in the world that is highly inflectional. A highly inflectional language like Ge’ez generates hundreds of words from a single root. The Ge’ez language is one of the lesser-studied languages and hence developing a morphological analyzer for the language is a valuable first step. Many NLP applications such as machine translation, information extraction and text to speech extraction require a base form of the word together with their syntactical information in order to perform further processing.

In this research work, a finite-state based morphological analyzer for Ge’ez verbs is proposed. While attempts have been made to develop a Ge’ez morphological analyzer (Abate, 2014; Desta, 2010), a morphological model for the Ge’ez language is still in the development stages. The purpose of this study is thus to exploit the finite-state approach in developing a morphological analyzer for Ge’ez verbs. One of the major differences in our Ge’ez morphological analyzer is that it not only performs analysis of words but also performs generation of words from a given root and its structural information.

The next section discusses the problem statement and the notion for conducting the research. Then I provide an overview of the different approaches to computational morphology. Then, I discuss the Ge’ez language and an approach to Ge’ez verbal morphology. Then follows the notion that led to this research project. Finally, I discuss the research methodology used in conducting this research work.

1.2 Problem Statement

The Ge’ez language is currently the liturgical language of the Ethiopian Orthodox Tewahido Church. The Church used to be the only school in Ethiopia for centuries, a place where people learned to read and write, learn the spiritual teachings, the literature ‘Qene’ and spiritual songs ‘Zema’ (Challiot, 2009). In fact, the Ge’ez language is still taught in the traditional schools of the Church to date. However, only people who dedicate their life to the service of God and the Church study the language as it is only taught in the Church. It could take years to become an expert in the Ge’ez language which includes the biblical study, the literature ‘Qene’ and the spiritual songs ‘Zema’. The ancient history, culture, spiritual, philosophical and medicinal knowledge (to name a few) of the country is written in the Ge’ez language (Sewasew, 1993). There are a large number of ancient books written in the Ge’ez language that document the identity of the people of Ethiopia. For instance, the ancient book of Enoch (Book of Enoch. 2019) was written in the Ge’ez language.

The information and knowledge encoded in the Ge’ez language is largely inaccessible to the current generation because the language does not have native speakers and the language is not taught in modern Ethiopian elementary or secondary schools. There are a few modern private schools that teach the Ge’ez language, in Ethiopia, in elementary and secondary school. አቡነ ጎርጎሪዮስ - *äbunä gorgoriyos* School, ምስካሄ ህዙናን - *məskahe həzunan* School and ራጉኤል -*raguʾel* school in Addis Ababa and ከሳቴብርሀን - *käsate bərhän* School in Mekele are some modern schools that teach Ge’ez language in Ethiopia. Besides, some universities in Ethiopia offer degree programs such as Addis Ababa University in Ge’ez Philology; Bahir Dar University MA in Ge’ez Literature; Wollo University BA in the Ge’ez language; Axum University BA in the Ge’ez language; Gonder University BA in the Ge’ez language. More than 30 universities in the world, for instance, universities in Europe (University of Hamburg) and America (University of Washington, University of Toronto) give some courses in the Ge’ez language. This shows that there is an interest in the Ge’ez language in order to access the rich knowledge inscribed in it. Therefore, for the language to be easily available for future generations and interested individuals, one way is to have NLP applications such as machine translation, spell checking and so forth. Being one of the integral components of natural language application, developing a morphological analyzer would be a valuable and necessary step.

Ge’ez verbs are highly inflectional and productive. Moreover, there are no native speakers of the Ge’ez language. Hence, this study aims at developing a morphological analyzer for the Ge’ez verbs.

1.2.1 The Purpose of the Research

Ge’ez word formation is basically characterized by a non-concatenative morphology but also uses affixes to create other word forms. Ge’ez verbs are the most inflectional and productive POS in the Ge’ez language. Most of the Ge’ez words are derived and formed from the verbs. A single Ge’ez verb may be inflected to more than hundreds of word forms. Moreover, a single inflected form of a verb can provide a complete sentence and / or meaning. For instance, the verb *አሐውር-äḥwr* means I will go.

Andualem (2007) describes how Ge’ez verbs are classified according to prominent Ge’ez schools and states that the Ge’ez verbs are categorized from six to eight main/head verbs. Desta (2010) designed a morphological analyzer for one of the head verbs (categories) *ቀተለ* - *qetele* of the Ge’ez language. In the same vein, Abate (2014) used a data-driven approach in developing a morphological analyzer for all Ge’ez verbs. In this study, a finite-state based morphological analyzer and generator is developed for all categories/head verbs of the Ge’ez language. The Ge’ez morphological analyzer would input a surface form of the Ge’ez verb and output the morphemes together with the feature tags or structural information about the word and vice versa. Hence, the main purpose of this research is to develop a finite-state based morphological analyzer for Ge’ez verbs. The two main processes of morphology are morphotactics – the sequencing of morphemes – and morphophonological alternations – the sound changes that occur at morpheme boundaries. In computational morphology, these two processes are modeled. The main objective of this research is to develop a morphological analyzer and generator for Ge’ez verbs using bidirectional finite-state technology. Finite-state technology is widely used in morphological analysis of different languages including Semitic languages such as (Beesley, 1998), Hebrew (Yona & Wintner, 2008) and Amharic (Amsalu & Gibbon, 2005). Finite-state technology offers the ability to handle concatenative as well as non-concatenative morphology and offers high speed and compact way of handling lexicon and morphological rules. In addition, the bidirectional feature of finite-state technology enables the use of the morphological analyzer as morphological generator in reverse. In this dissertation, we use a finite-state approach, as previously indicated.

1.2.2 Research Questions

1. Which Ge’ez verb classification is appropriate for Ge’ez verb computational morphology?
2. How can the non-concatenative morphology of Ge’ez verbs be efficiently represented using finite-state?
3. How to create FSTs that represent the morphotactics and the orthographic rules of the Ge’ez verbs?

4. How to create the lexicon for the Ge’ez verbs?
5. How to use the finite-state methods in developing a morphological analyzer for the Ge’ez verbs?
6. How to create gold-standard test data for evaluating the morphological analyzer?

1.2.3 Objective

Following the pioneering work by Desta (2010) in developing a morphological analyzer for one of the Ge’ez verb categories and Abate (2014) by developing a morphological analyzer for Ge’ez verbs using data-driven approaches, the main objective of this research is to extend the work by developing a finite-state based morphological analyzer and generator for all Ge’ez verb categories. To achieve the above objective, the researcher identified the following specific objectives:

- To study the morphotactics and the orthographic rules of the Ge’ez verb inflections for all the Ge’ez verb categories.
- To organize the Ge’ez verb lexicon (list of roots and affixes) that includes all the Ge’ez verb categories.
- To develop the morphotactics and alternation rules for all categories of the Ge’ez verbs using finite-state methods.
- To model and implement these morphotactics and alternation rules to create a finite-state morphological analyzer with Foma.
- To create gold-standard test data in consultation with the Ge’ez language experts.
- To test the morphological analyzer using the gold-standard test data.

1.2.4 Deliverables / Research Outcome

The main research outcome is a finite-state transducer that will analyze Ge’ez verbs. The Ge’ez morphological analyzer would be used as an important input for other Ge’ez language NLP application such as machine translation, automatic dictionaries, speech recognition etc. The secondary research outcome will be gold-standard data which can be used to evaluate other Ge’ez verb morphological analyzers in the future. Hence, the research output will be used:

- as an input for natural language applications of the Ge’ez language.
- for the Ge’ez language learners who are required to study the head verbs, their inflections and to identify the verbs that belong to each category. Using the Ge’ez morphological analyzer, the students will be able to identify the verb groups and their inflections.

- to evaluate other Ge’ez morphological analyzers.

1.3 Computational Morphology

Language is an important means of communication between human beings. Any natural language consists of a large number of words. However, these words are created from much smaller units called morphemes. Morphemes are the smallest unit of a language that carries meaning or grammatical explanation. Morphology deals with the study of words and word structure and describes how words are created from morphemes. Computational morphology can be defined as the use of a computer to perform the computational analysis and synthesis of word forms in the context of NLP (Jurafsky & Martin, 2008). Morphological analysis provides a morpheme together with the structural information such as the root, tense, mood, person etc. about a given word. Hence, a morphological analyzer breaks down a given word into smaller units such as roots, suffixes and prefixes. On the other hand, morphological generation provides a surface form of a word from a given morpheme (root) and its structural information.

NLP or human language processing is a field that aims to use the computer to perform important tasks involving human language such as human-machine communication, human-human communication or processing of text and speech (Trost, 2003). One of the major components for many NLP applications, especially for systems that involve parsing and / or generation of natural languages in written and spoken form (Jurafsky & Martin, 2008), is a morphological analyzer.

Computational morphology may be classified into two approaches, namely, rule-based and data-driven. Kazakov (2001) states that word segmentation (word morphology) methods may be based on clearly defined morphological rules (rule-based) or may be based on learning from text data (data-driven). The rule-based (symbolic) approach is based on linguistic theory and uses linguistically motivated rules for the analysis of the words whereas data-driven (statistical) approaches use the text data (corpus) to learn how to analyze or segment the words with little or no consideration for the knowledge of the language (Liddy, 2001). Machine learning and statistical methods make use of the data-driven approach while finite-state methods are rule-based.

Machine learning is the study of computational systems that gives computers the capability to learn from a given sample data and build an algorithm that enables the prediction of an output when receiving a new input (Alpaydin, 2010) “Machine learning is programming computers to optimize a performance criterion using example data or past experience” (Alpaydin, 2010, p. 3). Based on the type of input data for a machine-learning task, there are two types of learning, namely, supervised and unsupervised learning (Clark & Lappin, 2010).

In supervised learning, the machine is trained using sample data with their desired output. The goal is to develop an algorithm that correctly maps the input to the output so that when there are new input data, a corresponding output can be predicted (Alpaydin, 2010). On the other hand, in unsupervised learning, the machine is trained using sample data without its corresponding output. The goal is to develop an algorithm in order to learn more about the data structure. Applied to morphology, supervised learning makes use of a labeled corpus, that is, a lexicon with annotated text, whereas unsupervised learning uses a word list or corpus without annotated text (Alpaydin, 2010).

Statistical approaches in computational morphology make use of mathematical techniques to develop a model of morphological rules from natural language data. Statistical methods use statistical estimation on the language training corpus to predict useful information for new unknown input. Statistical methods often use large text corpora for developing the morphological model (Liddy, 2001).

The finite-state approach to computation morphology is based on representing a relationship between a set of strings, one representing the surface form of a word and the other representing its lexical form together with the morphological information about the word (Jurafsky & Martin, 2008). This relationship can be described using the metalanguage of regular expressions. Using a finite-state compiler, the regular expressions can be compiled into a finite-state transducer. Hence, a finite-state transducer serves a machine that reads one string – a word form – and generates another string – analysis of the word (Jurafsky & Martin, 2008). Finite-state approaches have been successfully used in developing morphological analyzers for a wide range of languages, including Semitic languages (Beesley, 2004).

The unavailability of electronic corpus data for Ge’ez languages makes it difficult to implement a data-driven approach to morphological analysis. For Ge’ez, as a morphologically complex language and a resource-scarce language, a rule-based approach to building a morphological analyzer was particularly suitable. The goal of this research work is to develop a Ge’ez verb morphological analyzer that performs both analysis and generation of verbs. Therefore, in this research work, a rule-based approach that uses finite-state tools and techniques was selected for the development of the morphological analyzer for Ge’ez verbs.

1.3.1 Ge’ez Language

Ge’ez is one of the ancient Semitic languages in the world. The Ge’ez language used to be the official language of Ethiopia until the 12th century when it was slowly replaced by Amharic and other local Ethiopian languages (Sewasew, 1993). Currently, Amharic language transliteration is available in Google searches and Microsoft applications to name a few. The Ge'ez language is one of the lesser-studied languages and hence developing a morphological analyzer for the language is a valuable first step. Many NLP applications such as machine translation, information extraction and text to speech extraction require base form of the word together with their syntactical information in order to perform further processing. Hence, a morphological analyzer is an important component in natural language applications.

The Ge'ez language has its own alphabet called ፊደል - *fidäl*. SERA (System for Ethiopic Representation in ASCII) is commonly used for transliteration between the Ethiopic alphabet and ASCII (Yacob, 1997). In this research, SERA transliteration is used to represent the Ge'ez language. There are 26 basic letters in the Ge'ez alphabet and each letter has seven forms with a total of 182 letters. The seven forms of the Ge'ez basic letters are represented using vowel sounds (ä, u, I, a, e, ə, o). The Ge'ez language is explained in detailed in Chapter 2.

As other Semitic languages, Ge'ez is characterized by non-concatenative morphology and is highly inflectional with a single verb being inflected as many as hundreds of word forms of the same or different parts of speech (POS). Ge'ez word formation can be considered as a root-pattern where the roots are a sequence of three or more consonants which are interdigitated with a vocalic pattern, a sequence of vowels with consonants into which the roots are being inserted. In addition, prefixes and suffixes may be added to indicate person, number, gender, and tense-mood. For instance, Table 1.1 shows some of the word forms obtained from the root constant ቅትል - *qtl*:

Table 1.1: Surface forms of the root qtl - to kill

Verb	**Word Forms**	**Tense**	**Meaning**
ቅትል - *qtl*	ቀተለ - *qätälä*	Perfective	He killed
	ይቀትል - *yəqätl*	Indicative	He will kill / He kills
	ይቅትል - *yəqtl*	Subjunctive/ Jussive	Kill

1.3.2 Finite-State Tools and Techniques

Finite-state technology uses regular expressions to represent morphological rules of a language. Finite-state automata or finite-state machine is a system that has a start state and one or more final states. The transition between states is triggered by an input and the transition between states is allowed only if the input is recognized by the system. A finite-state transducer (FST) is a type of finite-state automata with pairs rather than a single symbol which makes it possible to map one pair to another. It follows then, an FST can implement the relationship between the lexical and surface form of the word in morphological analysis. Some of the appealing features in using finite-state techniques are its simplicity in representing morphological rules, fast and compact in size when storing data and using morphological rules and its bidirectional feature, which works for both analysis and generation. Finite-state morphological tools and techniques are widely used in morphological analysis of different languages including Semitic languages such as Arabic (Beesley, 1998), Hebrew (Yona & Wintner, 2008) and Amharic (Amsalu & Gibbon, 2005). Hence, this approach can also be applied for the development of the Ge’ez verb morphological analyzer.

Some of the available finite-state tools include the Xerox finite-state tool (XFST), Helsinki Finite-State Technology (HFST) and Foma, which can be used for developing morphological analyzers. Foma is a free open-source finite-state tool for constructing finite-state automata and transducers (Hulden, 2009). Moreover, Foma is compatible with other finite-state tools such as the XFST. In this research, the open-source finite-state tool, Foma, was used for the development of the Ge’ez verb morphological analyzer.

1.4 Research Methodology

In this dissertation, research methodology is a framework that clearly explains the paradigm, the strategies and tools used in conducting the research.

1.4.1 Design and Creation

In this research, the design and creation research methodology was used for designing and developing the IT artifact - the finite-state based morphological analyzer for the Ge’ez verbs. The design and creation research methodology focuses on developing an IT artifact as a solution to a research problem and in doing so contributes to the body of knowledge - NLP. The steps to follow when using the design and creation research are as follows (Oates, 2005):

- Awareness – awareness of the research problem under study
- Suggestions – suggesting a solution to the research problem
- Development – design and implementation of the suggested solution using formal development methods
- Evaluation – testing the artifact or product using the evaluation criteria set
- Conclusion – reporting the findings of the research

The research process as illustrated in Figure 1.1

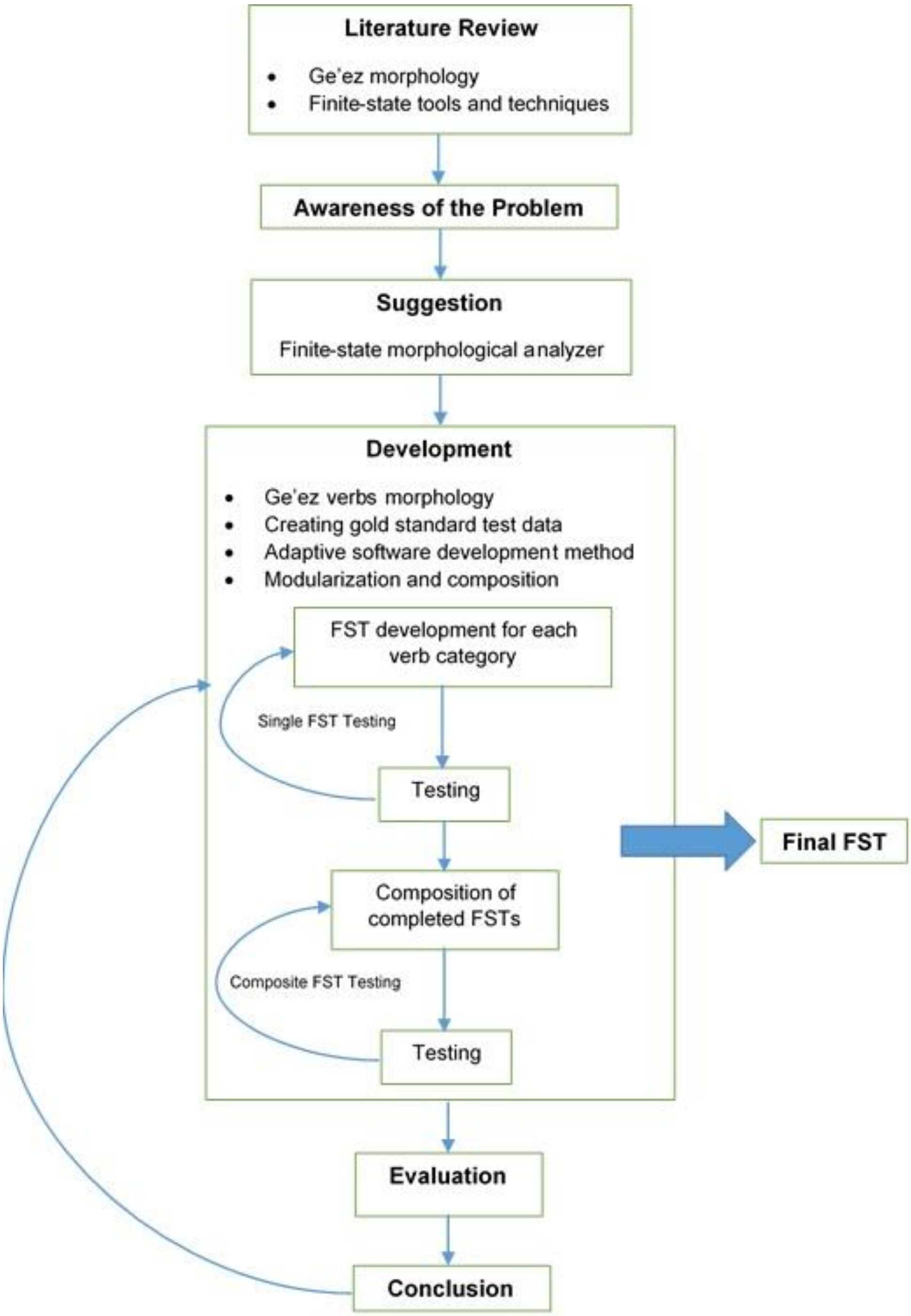

Literature Review
Ge'ez morphology
Finite-state tools and techniques
Awareness of the Problem
Suggestion
Finite-state morphological analyzer
Development
Ge'ez verbs morphology
Creating gold standard test data
Adaptive software development method
Modularization and composition
FST development for each verb category
Single FST Testing
Testing
Composition of completed FSTs
Composite FST Testing
Testing
Final FST
Evaluation
Conclusion

Figure 1.1: The research process

The first step in this approach was to perform a literature survey of the Ge’ez language in general, of the Ge’ez verb classification and the application of computational morphology for the Ge’ez language. Following the literature survey, the problem statement was determined and solutions to the problem suggested, namely, the development of the finite-state morphological analyzer for the Ge’ez verbs. In the development stage, the adaptive software development methodology was used focusing on modularization (component-based), testing, learning, iteration and composition. In developing the Ge’ez morphological analyzer, each verb-type morphology was developed and tested. Alternative paths were taken where necessary, and after successful testing of one component, the next component was developed. In the testing stage of a component, care was taken to ensure that mistakes that were observed would not be repeated in subsequent component developments. This approach of correcting problems at the initial stage of the development reduces rework in later stages of development. Composition of tested components was the next stage followed by testing the composite. This modularization, testing, composition, and testing was iterated until the final IT artifact (the final FST) met its objective. To evaluate the accuracy of the Ge’ez morphological analyzer (final FST), Ge’ez verbs were hand-annotated with the correct analysis by Ge’ez language experts to create a gold standard. The evaluation assumption was that for each Ge’ez verb in the gold-standard data, the Ge’ez morphological analyzer would produce the correct analysis.

1.4.2 Evaluation

In order to evaluate the accuracy of the Ge’ez morphological analyzer, we need a list of word forms annotated with their correct morphological analysis. A test data set was manually collected from the four chapters of Ethiopic Ge’ez New Testament Bible (chapter of Matthew, Markus, Lukas and John) and Ge’ez prayer book - *ውዳሴ ማርያም*- *wudase maryam*. A total of 1 519 verbs were collected from the Bible of which 1 365 verbs were selected for the test data set (non-repeat words). Ge’ez experts organized this test data set by providing the necessary structural and lexical information of each word, creating a gold standard. This gold standard was used as a reference in evaluating the Ge’ez morphological analyzer.

Each word (surface form) in the gold standard was input to the Ge’ez morphological analyzer and the analysis output produced by the system was compared with the gold standard analysis. The analysis output by the Ge’ez morphological analyzer (by comparing it against the gold standard) was measured in terms of precision (correct versus incorrect analysis) and recall (existing versus missing analysis) (Faaß, Heid & Schmid, 2010).

1.5 Organization of the Dissertation

This dissertation is organized as follows:

Chapter 1 describes the background of the Ge’ez language, the problem statement, why the study was conducted and its objectives. It also presents the deliverables or research outcomes, the research methodology and organization of the dissertation.

Chapter 2 describes in detail Ge’ez language and Ge’ez verb morphology. This chapter discusses the Ge’ez language alphabet, interdigitation of vowels into consonants and the prefixes and suffixes in the formation of verb forms.

Chapter 3 is a literature review on finite-state based morphological analysis by emphasizing Semitic language morphological analysis. In addition, it discusses NLP and computational morphology in general.

Chapter 4 introduces the research methodology used in conducting this research.

Chapter 5 discusses finite-state tools and techniques. It also describes the application of finite-state methods in computational morphology and how it is applied to this research.

Chapter 6 covers the design and development of the finite-state morphological analyzer of the Ge’ez verbs. All the steps involved in the design and development of the Ge’ez verb morphological analyzer are described in this chapter.

Chapter 7 presents the evaluation process, discusses the result and presents the findings of the evaluation.

Chapter 8 presents the conclusion and recommendations for future work.

Chapter 2 – Ge’ez Language

2.1 Introduction

This chapter serves as an introduction to the Ge'ez language, its unique alphabet, transliteration methods, and describes Ge’ez verb morphology.

The chapter is structured as follows; Section 2.2 provides and overview of the Ge’ez language. Section 2.3 explores the Ge’ez alphabet and details the transliterations employed in this dissertation. Section 2.4 offers a detailed description of Ge’ez verb morphology. It covers the formation of Ge’ez verbs from their root consonants using root-pattern morphology, affixation for affixes, and the phonological alternations that occur due to affixation. Additionally, this section discusses the head and troop classification of Ge’ez verbs. This chapter serves the crucial purpose of addressing the research sub-question of determining the appropriate verb classification for Ge’ez verb morphology computations.

2.2 Ge’ez

The Ge’ez language is a member of the southeast Semitic family (Lambdin, 1978) and is one of the ancient languages in the world. Currently, it serves as the liturgical language of the Ethiopian Orthodox Tewahido Church, the Eritrean Orthodox Tewahido Church, and the Ethiopian Catholic Church. For centuries, the Ethiopian Orthodox Church was the primary educational institution in Ethiopia, where people not only learned to read and write but also delved into spiritual teachings, literature ቅኔ - *qəne*, and spiritual songs ዜማ - *zema* (Challiot, 2009). Traditional Church schools still teach the Ge’ez language today, but it is predominantly studied by those who dedicate their lives to serving God and the Church. Becoming proficient in the Ge’ez language, which includes biblical studies and the study of ቅኔ - *qəne* and ዜማ - *zema* can take several years (Elleni, 1992). Despite the absence of native speakers, Ge’ez remains alive and in use within the Ethiopian Orthodox Church.

Ge’ez is not only a language but also a repository of Ethiopia's ancient history, culture, spirituality, philosophy, and medical knowledge (Sewasew, 1993). Numerous ancient books written in Ge’ez provide valuable insights into the identity of the Ethiopian people, including the notable example of the ancient book of Enoch (Sergew & Pawlos, 1997).

Similar to other Semitic languages, Ge’ez exhibits root-pattern morphology (Dillman et al., 2003). In addition, prefixes and suffixes are added to the root to create inflectional and derivational word forms. The next sections describe the Ge’ez alphabet and Ge’ez verbs morphology.

2.3 The Ge’ez alphabet / ፊደል - *Fedel*

The Ge’ez language consists of 26 basic letters referred to as ፊደል (*fidäl*) in Ge’ez. Each of these basic letters has seven forms, each representing a different sound. These seven forms are represented using the vowel sounds አ (ä), ኡ (u), ኢ (i), ኣ (a), ኤ (e), እ (ə), and ኦ (o), including the basic consonant letter itself. In total, there are 182 letters in the Ge’ez alphabet - ፊደል - *fidäl*. The vowels denote various sounds of the letter and follow this order: ግእዝ (*gə’zə*) - first-order, ካእብ (*ka’b*) - second-order, ሳልስ (*sals*) - third-order, ራብዕ (*rabə*) - fourth-order, ሓምስ (*hams*) - fifth-order, ሳድስ (*sads*) - sixth-order, and ሳብእ (*sabə*) - seventh-order. For example, combining the base letter በ -bä with the six vowel sounds ኡ (u), ኢ (i), ኣ (a), ኤ (e), እ (ə), and ኦ (o) results in six forms of the base letter: ቡ - bu, ቢ -bi, ባ -ba, ቤ - be, ብ - b, and ቦ - bo.

Apart from the basic 26 letters, the Ge’ez language includes four complex-sound letters (Adihana, 2015). Unlike the base letters, these four complex-sound letters have only five order forms. They are an extension of the base letters ከ - kä, ገ - gä, ቀ - qä and ኀ - ḫä formed by adding the letter ወ - wä. These four complex-sound base letters are ኰ - kwä, ጐ - gwä, ቈ - qwä, and ኈ - ḫwä.

In this study, we employ SERA (System for Ethiopic Representation in ASCII) (Yacob, 1997) for transliterating Ge’ez letters in the development of the Ge’ez morphological analyzer. It's important to note that SERA not only represents the Ge’ez language but also other Semitic languages of Ethiopia, such as Amharic and Tigrigna. Consequently, SERA includes letters that are not found in the Ge’ez language. However, for the purposes of this study, we utilize only those transliterations that are applicable to the Ge’ez language.

Table 2.1 displays the Ge’ez alphabet transliteration in SERA, while Table 2.2 focuses solely on the Ge’ez letters of the alphabet along with their corresponding transliterations used in the development of the Ge’ez analyzer for this research work. Furthermore, Table 2.3 provides IPA (International Phonetic Alphabet) transliteration for Ge'ez, which is used in the writing of this document.

Table 2.1: The Ethiopic Script in ASCII (adopted from SERA)

	1 ግዕዝ	**2** ካዕብ	**3** ሳልስ	**4** ራብዕ	**5** ሃምስ	**6** ሳድስ	**7** ሳብዕ	**8** ዲቃላ →	**12**	**9**	**10**	**11**	**12**
ሀ	he	hu	hi	ha	hE	h	ho						
ለ	le	lu	li	la	lE	l	lo				lWa		
ሐ	He	Hu	Hi	Ha	HE	H	Ho				HWa		
መ	me	mu	mi	ma	mE	m	mo	mWe	(mWu)	mWi	mWa	mWE	mW
ሠ	‘se	‘su	‘si	‘sa	‘sE	‘s	‘so						
ረ	re	ru	ri	ra	rE	r	ro						
ሰ	se	su	si	sa	sE	s	so						
ሸ	xe	xu	xi	xa	xE	x	xo						
ቀ	qe	qu	qi	qa	qE	q	qo	qWe	(qWu)	qWi	qWa	qWE	qW
ⷀ	‘qe	‘qu	‘qi	‘qa	‘qE	‘q	‘qo						
ቐ	Qe	Qu	Qi	Qa	QE	Q	Qo	QWe	(QWu)	QWi	QWa	QWE	QW
በ	be	bu	bi	ba	bE	b	bo						
ቨ	ve	vu	vi	va	vE	v	vo						
ተ	te	tu	ti	ta	tE	t	to						
ቸ	ce	cu	ci	ca	cE	c	co						
ኀ	‘he	‘hu	‘hi	‘ha	‘hE	‘h	‘ho	‘hWe	(‘hWu)	‘hWi	‘hWa	‘hWE	‘hW
ነ	ne	nu	ni	na	nE	n	no						
ኘ	Ne	Nu	Ni	Na	NE	N	No						
አ	e/a*	u/U	i	A/a	E	I	o/O						
ከ	ke	ku	ki	ka	kE	k	ko	kWe	(kWu)	kWi	kWa	kWE	kW
ⷈ	‘ke	‘ku	‘ki	‘ka	‘kE	‘k	‘ko						
ⷐ	Ke	Ku	Ki	Ka	KE	K	Ko	KWe	(KWu)	KWi	KWa	KWE	KW
ኸ	Xe	Xu	Xi	Xa	XE	X	Xo						
ወ	we	wu	wi	wa	wE	w	wo						
ዐ	‘e	‘u/’U	‘i	‘A/’a	‘E	‘I	‘o/’O						
ዘ	ze	zu	zi	za	zE	z	zo				zWa		
ዠ	Ze	Zu	Zi	Za	ZE	Z	Zo				ZWa		
የ	ye	yu	yi	ya	yE	y	yo				yWa		
ደ	de	du	di	da	dE	d	do				dWa		
ዸ	De	Du	Di	Da	DE	D	Do				DWa		
ጀ	je	ju	ji	ja	jE	j	jo				jWa		
ገ	ge	gu	gi	ga	gE	g	go	gWe	(gWu)	gWi	gWa	gWE	gW
ጘ	‘ge	‘gu	‘gi	‘ga	‘gE	‘g	‘go						

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**12**	**9**	**10**	**11**	**12**
	ግዕዝ	ካዕብ	ሳልስ	ራብዕ	ሃምስ	ሳድስ	ሳብዕ	ዲቃላ →					
ኘ	Ge	Gu	Gi	Ga	GE	G	Go	GWe	(GWu)	GWi	GWa	GWE	GW
ጠ	Te	Tu	Ti	Ta	TE	T	To				TWa		
ጨ	Ce	Cu	Ci	Ca	CE	C	Co				CWa		
ጰ	Pe	Pu	Pi	Pa	PE	P	Po				PWa		
ጸ	Se	Su	Si	Sa	SE	S	So				SWa		
ፀ	‘Se	‘Su	‘Si	‘Sa	‘SE	‘S	‘So						
ፈ	fe	fu	fi	fa	fE	f	fo	fWe	(fWu)	fWi	fWa	fWE	fW
ፐ	pe	pu	pi	pa	pE	p	po	pWe	(pWu)	pWi	pWa	pWE	pW

Table 2.2: Ge’ez Alphabet

	ግዕዝ 1st	ካዕብ 2nd	ሣልስ 3rd	ራብዕ 4th	ሐምስ 5th	ሳድስ 6th	ሳብዕ 7th		ግዕዝ 1st	ካዕብ 2nd	ሣልስ 3rd	ራብዕ 4th	ሐምስ 5th	ሳድስ 6th	ሳብዕ 7th
1	ሀ he	ሁ hu	ሂ hi	ሃ ha	ሄ hE	ህ h	ሆ ho	16	ዐ ‘A	ዑ ‘U	ዒ ‘Ai	ዓ ‘Aa	ዔ ‘AE	ዕ ‘I	ዖ ‘O
2	ለ le	ሉ lu	ሊ li	ላ la	ሌ lE	ል l	ሎ lo	17	ዘ ze	ዙ zu	ዚ zi	ዛ za	ዜ zE	ዝ z	ዞ zo
3	ሐ He	ሑ Hu	ሒ Hi	ሓ Ha	ሔ HE	ሕ H	ሖ Ho	18	የ ye	ዩ yu	ዪ yi	ያ ya	ዬ yE	ይ y	ዮ yo
4	መ me	ሙ mu	ሚ mi	ማ ma	ሜ mE	ም m	ሞ mo	19	ደ de	ዱ du	ዲ di	ዳ da	ዴ dE	ድ d	ዶ do
5	ሠ ‘se	ሡ ‘su	ሢ ‘si	ሣ ‘sa	ሢ ‘sE	ሥ ‘s	ሦ ‘so	20	ገ ge	ጉ gu	ጊ gi	ጋ ga	ጌ gE	ግ g	ጎ go
6	ረ re	ሩ ru	ሪ ri	ራ ra	ሬ rE	ር r	ሮ ro	21	ጠ Te	ጡ Tu	ጢ Ti	ጣ Ta	ጤ TE	ጥ T	ጦ To
7	ሰ se	ሱ su	ሲ si	ሳ sa	ሴ sE	ስ s	ሶ so	22	ጰ Pe	ጱ Pu	ጲ Pi	ጳ Pa	ጴ PE	ጵ P	ጶ Po
8	ቀ qe	ቁ qu	ቂ qi	ቃ qa	ቄ qE	ቅ q	ቆ qo	23	ጸ Se	ጹ Su	ጺ Si	ጻ Sa	ጼ SE	ጽ S	ጾ So
9	በ be	ቡ bu	ቢ bi	ባ ba	ቤ bE	ብ b	ቦ bo	24	ፀ ‘Se	ፁ ‘Su	ፂ ‘Si	ፃ ‘Sa	ፄ ‘SE	ፅ ‘S	ፆ ‘So
10	ተ te	ቱ tu	ቲ ti	ታ ta	ቴ tE	ት t	ቶ to	25	ፈ fe	ፉ fu	ፊ fi	ፋ fa	ፌ fE	ፍ f	ፎ fo
11	ኀ ‘he	ኁ ‘hu	ኂ ‘hi	ኃ ‘ha	ኄ ‘hE	ኅ ‘h	ኆ ‘ho	26	ፐ pe	ፑ pu	ፒ pi	ፓ pa	ፔ pE	ፕ p	ፖ po
12	ነ ne	ኑ nu	ኒ ni	ና na	ኔ nE	ን n	ኖ no	27	ቈ qWe		ቊ qWi	ቋ qWa	ቌ qWE	ቍ qW	
13	አ A	ኡ U	ኢ Ai	ኣ Aa	ኤ AE	እ I	ኦ O	28	ኰ kWe		ኲ kWi	ኳ kWa	ኴ kWE	ኵ kW	
14	ከ ke	ኩ ku	ኪ ki	ካ ka	ኬ kE	ክ k	ኮ ko	29	ጐ gWe		ጒ gWi	ጓ gWa	ጔ gWE	ጕ gW	
15	ወ we	ዉ wu	ዊ wi	ዋ wa	ዌ wE	ው w	ዎ wo	30	ዀ ‘hWe		ዂ ‘hWi	ዃ ‘hWa	ዄ ‘hWE	ዅ ‘hW	

Table 2.3: Ge’ez Alphabet (IPA)

	ግዕዝ 1st	ካዕብ 2nd	ሣልስ 3rd	ራብዕ 4th	ሐምስ 5th	ሳድስ 6th	ሳብዕ 7th		ግዕዝ 1st	ካዕብ 2nd	ሣልስ 3rd	ራብዕ 4th	ሐምስ 5th	ሳድስ 6th	ሳብዕ 7th
1	ሀ hä	ሁ hu	ሂ hi	ሃ ha	ሄ he	ህ h	ሆ ho	16	ዐ ʿ	ዑ ʿu	ዒ ʿi	ዓ ʿa	ዔ ʿe	ዕ ʿə	ዖ ʿo
2	ለ lä	ሉ lu	ሊ li	ላ la	ሌ le	ል l	ሎ lo	17	ዘ zä	ዙ zu	ዚ zi	ዛ za	ዜ ze	ዝ z	ዞ zo
3	ሐ ḥä	ሑ ḥu	ሒ ḥi	ሓ ḥa	ሔ ḥe	ሕ ḥ	ሖ ḥo	18	የ yä	ዩ yu	ዪ yi	ያ ya	ዬ ye	ይ y	ዮ yo
4	መ mä	ሙ mu	ሚ mi	ማ ma	ሜ me	ም m	ሞ mo	19	ደ Dä	ዱ du	ዲ di	ዳ da	ዴ de	ድ d	ዶ do
5	ሠ sä́	ሡ sú	ሢ sí	ሣ sá	ሤ sé	ሥ s´	ሦ só	20	ገ gä	ጉ gu	ጊ gi	ጋ ga	ጌ ge	ግ g	ጎ go
6	ረ rä	ሩ ru	ሪ ri	ራ ra	ሬ re	ር r	ሮ ro	21	ጠ ṭä	ጡ ṭu	ጢ ṭi	ጣ ṭa	ጤ ṭe	ጥ ṭ	ጦ ṭo
7	ሰ sä	ሱ su	ሲ si	ሳ sa	ሴ se	ስ s	ሶ so	22	ጰ p̣ä	ጱ p̣u	ጲ p̣i	ጳ p̣a	ጴ p̣e	ጵ p̣	ጶ p̣o
8	ቀ qä	ቁ qu	ቂ qi	ቃ qa	ቄ qe	ቅ q	ቆ qo	23	ጸ ṣä	ጹ ṣu	ጺ ṣi	ጻ ṣa	ጼ ṣe	ጽ ṣ	ጾ ṣo
9	በ bä	ቡ bu	ቢ bi	ባ ba	ቤ be	ብ b	ቦ bo	24	ፀ ḍä	ፁ ḍu	ፂ ḍi	ፃ ḍa	ፄ ḍe	ፅ ḍ	ፆ ḍo
10	ተ tä	ቱ tu	ቲ ti	ታ ta	ቴ te	ት t	ቶ to	25	ፈ fä	ፉ fu	ፊ fi	ፋ fa	ፌ fe	ፍ f	ፎ fo
11	ኀ ḫä	ኁ ḫu	ኂ ḫi	ኃ ḫa	ኄ ḫe	ኅ ḫ	ኆ ḫo	26	ፐ pä	ፑ pu	ፒ pi	ፓ pa	ፔ pe	ፕ p	ፖ po
12	ነ nä	ኑ nu	ኒ ni	ና na	ኔ ne	ን n	ኖ no	27	ቈ qʷä		ቊ qʷi	ቋ qʷa	ቌ qʷe	ቍ qʷ	
13	አ ʾ	ኡ u	ኢ i	ኣʾ a	ኤ e	እ ə	ኦ o	28	ኰ kʷä		ኲ kʷi	ኳ kʷa	ኴ kʷe	ኵ kʷ	
14	ከ kä	ኩ ku	ኪ ki	ካ ka	ኬ ke	ክ k	ኮ ko	29	ጐ gʷä		ጒ gʷi	ጓ gʷa	ጔ gʷe	ጕ gʷ	
15	ወ wä	ዉ wu	ዊ wi	ዋ wa	ዌ we	ው w	ዎ wo	30	ዀ ḫʷä		ዂ ḫʷi	ዃ ḫʷa	ዄ ḫʷe	ዅ ḫʷ	

There is a distinction between the SERA transliteration and the Ge’ez transliteration, as shown in Table 2.2, for the letters እ - *I* and ዕ - *'I*. The researcher chose to modify the transliteration for these two letters due to potential ambiguities when they are used in conjunction with other letters. This issue arises because the five forms of the letter እ - *I* are also used as vowels to create the six forms of other alphabets. In the SERA transliteration, the seven forms of the alphabet እ - *I* are represented as አ, ኡ, ኢ, ኣ, ኤ, እ, and ኦ, corresponding to e/a*, u/U, i, A/a, o/O. The vowels e, u, i, a, E, and o, which are used to create other forms of each letter, also represent the forms of the letter እ - *I* This can lead to ambiguities in word formation.

For example, the word ወጽኢ, meaning 'she won,' would be transliterated as "*weSi*" using SERA, which could also represent ወጺ. However, using the modified transliteration, ወጽኢ and ወጺ would be represented as "*weSAi*" and "*weSi*," respectively. This modified representation effectively resolves any ambiguity associated with the letter እ - *I* The modification for the letter ዕ - *'I* is done to maintain consistency with the similar-sounding letter እ - *I*.

In this study, the sixth order of the Ge’ez letter is considered as the root consonant and is referred to as the term "radical." Moreover, the term "root" refers to the lexical morpheme consisting of consonants, while "stem" refer to morphemes formed by intercalating vowels into root consonants. A "verb/surface form" refers to a verb obtained by adding affixes.

2.4 Verbs

The Ge’ez language is characterized by non-concatenative morphology, particularly in its main word formation process. Ge’ez verbs represent the most complex POS, with a single verb undergoing inflection into hundreds of different word forms (Adihana, 2015). Similar to other Semitic languages, Ge’ez verbs can be described using root-pattern morphology (McCarthy, 1981). In this system, a "root" comprises a set of consonants representing the lexical morpheme, while a "pattern" consists of vowels that are inserted into the root to create a stem. The root-pattern morphology is applied in a number of Semitic languages including Amharic (Amsalu & Demeke, 2006; Amsalu & Gibbon, 2005) and it can also be applied to Ge’ez verbs.

For example, consider a tri-radical root ቅትል - *qtl*. When this root is combined with a vocalic pattern (vowels) 'ä – ä' placed among the radicals or consonants, it results in the stem ቀተል - *qätäl*. Adding a suffix 'ä' to the stem gives us the verb ቀተለ - *qätälä*, which means 'he killed' and represents the third person singular male perfective form of the root ቅትል - *qtl*. Thus, a template - CVCVC, where V = ä, represents a perfective form for verbs of the ቅትል - *qtl* category.

2.4.1 Root and Pattern Morphology

Ge’ez roots consist solely of consonants, and the number of radicals in these roots may vary from two to seven. While Keleb (2010) suggests that Ge’ez roots typically have two to four consonants, roots with five to seven radicals also exist, often derived from other roots (Keleb (Memhir), 2010; Mercer, 1961). However, Kifle (1956) and Adihana (2015) argue that although tri-radical roots are predominant in Ge’ez, roots with five to seven radicals do exist. Notably, the most common Ge’ez verbs are tri-radicals (Keleb (Memhir), 2010; Desta, 2010), and roots with more than four consonants are uncommon (Desta, 2010).

Certain Ge’ez verbs with two radicals are considered tri-radicals because they were originally tri-radicals (Adihana (Memhir), 2015; Kifle (Aleka), 1956; Lambdin, 1978). Typically, these two-radical verbs consist of the letters ወ-*wä* or የ-*yä*, which are also referred to as semi-vowels in Ge’ez teachings (Andualem, 2007). In the Ge’ez language, verbs containing ወ-*wä* or የ-*yä* may have one or more radicals dropped in some verb forms. However, roots with double ወ-*wä* or የ-*yä* are not truncated, although some verbs with double consonants may also undergo truncation in Ge’ez verb formation. For example, the two-radical verb ቆመ-*qomä* was originally ቀወመ-*qäwämä*, ሤመ-*śemä* was originally ሠየመ-*śäyämä*, and ሐመ-*ḫämä* was originally ሐመ-*ḫämämä*.

In this research, two-radical verbs are treated as tri-radicals by adding the dropped radicals for roots containing ው-*w* or ይ-*y*. This approach is also applied to roots with double consonants. However, in the surface form, specific rules are applied to drop the letters ው-*w* or ይ-*y* or double consonants.

A pattern is a set of vowels that are inserted in the consonant roots to form a stem. For each verb type, a template together with the vocalic pattern is inserted into the consonant roots to form a verbal stem (McCarthy, 1981). For instance, the verb root consonant *qtl* when inserted with a template CVCVC and a vocalic pattern ä ä produces a verbal stem for the perfective form of *qtl* - *qätäl*. When the root *qtl* is inserted with CVCC template and vocalic pattern ä- -, a verbal stem is produces for the indicative form of *qtl*- *qätl*. Hence, Ge’ez verbal stems are created from root consonants by intercalating vocalic or vowel patterns. The surface forms of the Ge’ez verbs are then formed by adding a prefix and/or a suffix to the stem depending on the verb tense, mood, number and gender.

In summary, Ge’ez verb formation starts from the root, consisting of only consonants, and a verbal stem is created by the intercalation of vocalic patterns. Then follows the addition of prefixes and suffixes to create a surface form of Ge’ez verb. In this study, Ge’ez language experts organized Ge’ez verbs test data set by providing the necessary structural and lexical information of each word. From this data set, the researcher organized root consonants that are used in the lexical file of the Ge’ez verb morphological analyzer. The next step is to define the vocalic pattern for the formation of the verbal stems. In order to define the Consonant Vowel (CV) pattern for the verb types, we need to identify the classification of the Ge’ez verbs.

2.4.2 Ge’ez Verb Classification

Ge’ez verbs are classified as heads and troops. Andualem (2007) studied how the Ge’ez verbs are classified in some of the prominent schools of ቅኔ-*qəne* in the Church namely ዋሸራ-*washära*, ጎንጂ-*gonji* and ዋልዳ -*walda*. While many Churches have schools for teaching various aspects of their teachings, schools like the aforementioned Ge’ez schools are recognized as centers for educating higher-level scholars in the Ge’ez language and literature ቅኔ-*qəne* (Challiot, 2009; Andualem, 2007).

All these schools classify Ge’ez verbs into heads and troops. However, there is disagreement among the schools regarding the number of head verbs. Head or model verbs are those that can represent other verbs in their category, while troops are verbs that follow the inflection and derivation patterns of the head verb in the same category. Andualem (2007) explains that Ge’ez verbs, referred to as heads and troops, are grouped based on their conjugation forms. According to the three prominent Ge’ez schools, *washära*, *gonji*, and *walda*, Ge’ez verbs may be classified into six to eight head verbs (Andualem, 2007). To be categorized as part of a head verb, troop verbs must exhibit similarity in their letter patterns (Andualem, 2007).

Table 2.4 illustrates the classifications of Ge’ez verbs according to these three Ge’ez schools. The table reveals that there are some common head verbs among the three schools. However, the number and type of troops associated with each head verb differ in each school. For example, *washära* scholars consider *gäbrä, äəmärä, śemä*, and *qomä* as head verbs, whereas *walda* and *gonji* scholars classify them as troops of the head verb *qätälä. walda* and *gonji* scholars argue that *śemä* and *qome* have the semi-vowels – *y* and – *w* and were originally *säyämä* and *qäwämä*, demonstrating that they share the same pattern as *qätälä*, while *äəmärä* is a causative form of the verbs *märä* (Andualem, 2007). In *walda* Ge’ez verb classifications, troops exhibit similar conjugation patterns as their heads, but some troops have different perfective forms from their heads (Andualem, 2007). Similarly, in the *gonji* classification, troops follow the same conjugation patterns as their heads but may differ in the number of radicals and assimilation of radicals from their heads (Andualem, 2007).

While troops in all three schools typically follow the same conjugational pattern as their heads, some troops in the *walda* and *gonji* classifications have different forms from their heads. For example, in the *walda* classification, the perfective form of the head verb ቅድስ-*qds* is ቀደሰ-*qäddäsä*, whereas the perfective form of its troop verb እንግልግ-*ʿənglg* is አንገለገ-*ʿängälägä* – a form distinct from its head. However, in the *washära* Ge'ez classification, the troops adhere to the same conjugation pattern as their heads and exhibit identical forms.

Table 2.4: Ge'ez Verbs Classification according to the three schools

washära	***walda***	***gonji***
ቀተለ- *qätälä*	ቀተለ- *qätäl*ä	ቀተለ- *qätäl*ä
ቀደሰ- *qäddäsä*	ቀደሰ- *qäddäsä*	ቀደሰ- *qäddäsä*
ገብረ- *gäbrä*	ማህረከ- *mahräkä*	-
አእመረ- *äəmärä*	ተንበለ- *tänbälä*	ማህረከ- *mahräkä*
ባረከ- *baräkä*	ባረከ- *baräkä*	ባረከ- *baräkä*
ሤመ- *śemä*	ሤሠየ- *śeśäyä*	ጌገየ- *gegäyä*
ብህለ- *bəhlä*	ክህለ- *kəhle*	ጠብጠበ - *ṭäbṭäbä*
ቆመ- *qomä*	ጦመረ- *ṭomärä*	ኖለወ- *noläwä*

While the number and type of head verbs and the number and type of troops associated with each head verb vary among the three schools, the conjugation pattern for a particular verb remains consistent. For example, Kifle (1956) classifies *gäbrä* as a troop of *qätälä*, whereas Andualem (2007) and Adhana (2015) consider *gäbrä* as the head verb. However, the conjugation pattern for *gäbrä* remains the same, regardless of whether it is classified as a head or a troop.

One of the renowned Ge'ez schools, *washära*, categorizes Ge'ez head verbs into eight distinct heads (Andualem, 2007). Additionally, other Ge'ez language scholars (Adihana (Memhir), 2015; Berhanu, 2006; Sewasew, 1993) also classify Ge'ez head verbs into eight categories. Table 2.5 shows Ge'ez verb classification according to the *washära* School of Ge'ez (Adihana, 2015; Andualem, 2007; Kifle, 1956). In Ge'ez, the perfective form of the third person singular male is considered the main verb, and Table 2.5 presents the head verbs accordingly.

Table 2.5: Ge’ez Verbs Classification

washära	**Adhana**	**Kifle**
ቀተለ- *qätälä*	ቀተለ- *qätälä*	ቀተለ- *qätälä*
ቀደሰ- *qäddäsä*	ቀደሰ- *qäddäsä*	ቀደሰ- *qäddäsä*
ገብረ- *gäbrä*	ገብረ- *gäbrä*	-
አእመረ- *äəmärä*	ተንበለ- *tänbälä*	ማህረከ- *mahräkä*
ባረከ- *baräkä*	ባረከ- *baräkä*	ባረከ- *baräkä*
ሤመ- *śemä*	ኤለ- *ʾelä*	ዴገነ- *degänä*
ብህለ- *bəhlä*	ክህለ- *kəhlä*	ደንገጸ- *dängäṣe*
ቆመ- *qomä*	ኦደ- *ʾodä*	ኖለወ- *noläwä*

Table 2.5 shows that the *washära* Ge’ez school and Adihana (2015) share eight similar head verbs, while Kifle (1956) presents seven head verbs. For the purposes of this study, the verb classification aligned with the *washära* School is adopted. This decision is based on the fact that, unlike *gonji* and *walda* classifications, the *washära* classification maintains uniform conjugation patterns and forms between troops and their respective head verbs, making it well-suited for computational representation. This section not only describes the three verb classifications but also highlights the suitability of the *washära* verb classification for computational representation. This choice effectively addresses the research sub-question concerning the most appropriate Ge’ez verb classification for computational morphology. Table 2.6 presents the eight head verbs adopted for this research work.

Table 2.6: Head verbs

Head Verbs	**Root**	**Template (Perfective)**	**Vocalic Pattern**	**Suffix**	**Glossary**
ቀተለ- *qätälä*	*qtl*	CVCVC	*ä - ä*	*ä*	He killed
ቀደሰ- *qäddäsä*	*qds*	CVCVC	*ä - ä*	*ä*	He consecrated
ገብረ- *gäbrä*	*gbr*	CVCC	*ä - -*	*ä*	He did
አእመረ- *äəmärä*	IImr	CVCCVC	*ä - -ä*	*ä*	He knew
ባረከ- *baräkä*	*brk*	CVCVC	a - *ä*	*ä*	He praised
ሤመ- *śemä*	*śym*	CVCV	*ä - ä*	*ä*	He set/placed
ብህለ- *bəhlä*	*bhl*	CCC	-	*ä*	He said
ቆመ- *qomä*	*qwm*	CVCVC	*ä - ä*	*ä*	He stood

Table 2.6 presents a list of the eight head verbs, their respective root consonants, templates, and vocalic patterns for the perfective tense-mood. These vocalic patterns are intercalated into the root consonants based on the specified templates to create verbal stems.

For this study, the *washära* Ge’ez verb classification comprising eight head verbs was chosen. In collaboration with Ge’ez language experts, the vocalic patterns for each verb type (such as perfective, indicative, subjunctive, and others) were carefully defined for each head verb. Additionally, troop verbs associated with each head verb were organized from the test dataset, along with their root consonants. The vocalic patterns applied to the head verbs were also applied to the troop verbs within the same category. For example, the verb *'hädägä'* is a troop of the head verb *'qätälä*,' and therefore, *'hädägä'* follows the same conjugation pattern as *'qätälä*.' The process of intercalating vowels into the root consonants was accomplished using the alternation rule component of the Ge’ez morphological analyzer.

2.4.3 Verbal Stems

A verbal stem is created by the interdigitation or intercalation of vocalic patterns into root consonants. Ge’ez verbs, similar to the categorization of Amharic verbs described by Tachbelie (2010) for Amharic, can be categorized as either simple or derived verbs. According to Ge’ez scholars and prominent Ge’ez schools, simple verbs are formed from all head verbs, while derived verbs may not be formed from all head verbs and/or their troops (Adihana (Memhir), 2015). This distinction arises from the fact that only transitive verbs can be derived into both simple and derived verb forms. In the Ge’ez language, simple and derived verbs are referred to as 'አእማድ' (*ʾaəmad*) - pillars (Adihana (Memhir), 2015). Simple verbs are further categorized as 'አድራጊ' (*ʾadragi*) - base, while derived verbs are categorized as 'አስደራጊ' (*ʾasdäragi*) - causative, 'አስደራራጊ' (*ʾasdäraragi*) - causative-reciprocal, 'ተደራጊ' (*tädäragi*) - reflexive, and 'ተደራራጊ' (*tädäraragi*) – reciprocal.

The simple verbal stems encompass forms such as perfective, indicative, subjunctive, jussive, gerundive, and infinitive, while the derived verbal stems include causative, reflexive, reciprocal, and causative-reciprocal.

2.4.3.1 Simple Verbal Stems

In Ge’ez, the primary verb forms are known as 'ዐበይት አናቅጽ' (*ʿabäyt ʾanaqṭ*) - main verbs, and 'ንኡሳን አናቅጽ' (*n ʾusan anaqṭ*) - subordinate verbs, as per Ge’ez language teachings (Adihana (Memhir), 2015). The main verb forms include the perfective, indicative, and subjunctive tense-moods, while the subordinate verb forms encompass the jussive, gerundive, and infinitive.

Both transitive and intransitive verbs in Ge’ez exhibit six simple verbal stems, which are the perfective, indicative, subjunctive, jussive, gerundive, and infinitive forms. These verbal stems are created by intercalating vowels into consonant roots, following their respective conjugation patterns. The type and number of vowels required for intercalation vary depending on the verb type, with some verbs requiring no vowels, while others may require one or more.

For instance, the perfective form of the ብህለ *-bəhlä* verb type requires no vowel for the intercalation whereas the perfective form of the ባረከ *-baräkä* verb type requires two vowels a and ä for the intercalation. Table 2.7 shows the simple verbal stems formed from the eight head verbs by intercalating vowels into the consonant roots.

Table 2.7: Simple verbs conjugation

አንቀጽ **Tense-mood**	**Verbal stems**							
Base	ቀተለ *qätälä*	ቀደሰ *qäddäsä*	ገብረ *gäbrä*	አእመረ *äəmärä*	ባረከ *baräkä*	ሤመ *śemä*	ብህለ *bəhlä*	ቆመ *qomä*
ቀዳማይ (ኃላፊ) Perfective	ቀተል *qätäl*	ቀደስ *qäddäs*	ገብር *gäbr*	አእመር *ä̉əmär*	ባረክ *baräk*	ሤም *śem*	ብህል *bəhl*	ቆም *qom*
ካላአይ (ትንቢት) Indicative	ቀትል *qätl*	ቄድስ *qeds*	ገብር *gäbr*	አምር *ämr*	ባርክ *bark*	ሠይም *s̋äym*	ብል *bəl*	ቀውም *qäwm*
ሣልሣይ (ትዕዛዝ) Subjunctive	ቅትል *qtl*	ቀድስ *qäds*	ግበር *gbrä*	አእምር *ä̓əmr*	ባርክ *bark*	ሥይም *śəym*	ብህል *bəhl*	ቁም *qum*
ዘንድ Jussive	ቅትል *qtl*	ቀድስ *qäds*	ግበር *gbrä*	አእምር *ä̓əmr*	ባርክ *bark*	ሥይም *śəym*	ብህል *bəhl*	ቁም *qum*
ቦዝ Gerundive	ቀቲል *qätil*	ቀዲስ *qäddis*	ገቢር *gäbir*	አእሚር *ä̓əmir*	ባሪክ *barik*	ሠዪም *s̋äyim*	ብሂል *bəhil*	ቀዊም *qäwim*
አርእስት Infinitive	ቀቲል *qätil*	ቀድስ *qäds*	ገቢር *gäbir*	አእምር *ä̓əmir*	ባርክ *bark*	ሠይም *s̋äym*	ብሂል *bəhil*	ቀዊም *qäwim*
	ቀቲሎት *qätilot*	ቀድሶት *qädsot*	ገቢሮት *gäbirot*	አእምሮት *ä̓əmrot*	ባርኮት *barkot*	ሠይሞት *s̋äymot*	ብሂሎት *bəhilot*	ቀዊሞት *qäwimot*

Subsequently, affixes are appended to these verbal stems to create the surface form of the simple verbs. For instance, the verb ይቀትል - *yəqätl*, meaning 'he will kill') is a surface form derived from the stem *'qätl*,' which represents the indicative form of *'qtl*,' with the prefix *'y'* added.

2.4.3.2 Derived verbal stems

In Ge’ez, there are four derived verbal stems: causative, causative reciprocal, reflexive, and reciprocal. These derived verbal stems are formed by adding prefixes and/or modifying the vowel pattern of the simple verbal stems. The prefixes used for derived verbs are - ኣ-*ä*, ኣስተ -*äste*, or ተ -*tä*. Table 2.8 illustrates the five verbal stems, specifically ኣዕማድ-*äəmad* (pillars), in the perfective tense-mood for the ቅትል-*qtl* verb types.

Table 2.8: ኣዕማድ- The five pillars

Perfective tense	**Pattern**	**Vowels**	**Stem**	**Prefix**	**Suffix**	**Surface form**	**Glossary**
ኣድራጊ Base	CVCVC	ä - ä	ቀተል *qätäl*	-	ä	ቀተለ *qätälä*	He killed
ኣስደራጊ Causative	CCVC	ä	ቅተል *qətäl*	ኣ *ʾa*	ä	ኣቅተለ *äqtälä*	He caused someone to be killed
ኣስደራራጊ Causative reciprocal	CVCVC	a - ä	ቃተል *qatäl*	ኣስተ *ʾastä*	ä	ኣስተቃተለ *ästäqatälä*	He caused them to kill each other
ተዳራጊ Reflexive	CVCC	ä	ቀትል *qätl*	ተ *tä*	ä	ተቀትለ *täqätlä*	He is killed
ተደራራጊ Reciprocal	CVCVC	a - ä	ቃተል *qatäl*	ተ *tä*	ä	ተቃተለ *täqatälä*	Killed each other (he and someone else)

Both simple and derived verbs can be inflected for person, gender and number.

2.4.3.3 Person

The Ge’ez language has ten pronouns: first person singular and plural, second person male singular and plural, second person female singular and plural, third person male singular and plural, and third person female singular and plural. Table 2.9 displays the pronouns.

Table 2.9: Pronouns

Pronouns		**Person**
ኣነ - änä	I	First person singular
ንሕነ- nəḥnä	We	First person plural
ኣንተ - äntä	You	Second person singular male
ኣንትሙ - äntmu	You	Second person plural male
ኣንቲ - änti	You	Second person singular female
ኣንትን - äntn	You	Second person plural female
ዉእቱ - wəʿətu	He	Third person singular male
ውእቶሙ- wəʿətomu	They	Third person plural male
ይእቲ - yəʿəti	She	Third person plural female
ውእቶን - wəʿəton	They	Third person plural female

Ge’ez verbal stems are inflected for both subject and object person. Affixes are added to the verbal stem according to the tense-mood to indicate the person. Subject markers can be added as prefixes, suffixes, or circumfixes, depending on the tense-mood of the verb. In the indicative, subjunctive, and jussive tense-moods, the subject markers are added as prefixes or circumfixes. However, in the perfective tense-moods, they serve as suffixes. Table 2.10 shows the subject marker affixation in different tense-moods, including gender and number.

Table 2.10: Subject markers

Person	**Number**	**Gender**	**Perfective**	**Indicative**	**Jussive**	**Subjunctive**	**Gerundive**	**Infinitive**
First	Singular	M/F	-ኩ -ku	እ- ʾə -	እ- ʾə -	እ- ʾə -	-የ -yä	-ዬ -ye
First	Plural	M/F	-ነ *-nä*	ን- *n-*	ን- *n-*	ን- *n-*	-አነ *-äne*	-ነ *-ne*
Second	Singular	Male	-ከ *-kä*	ት- *t-*	-	ት- *t-*	-አከ *-äke*	-ከ *-ke*
Second	Plural	Male	-ክሙ *-kmu*	ት-ኡ *t-ʾu*	-ኡ *- ʾu*	-ትኡ *t-ʾu*	-አክሙ - *äkmu*	-ክሙ - *kmu*
Second	Singular	Female	-ኪ *-ki*	ት-ኢ *t-ʾi*	-ኢ *- ʾi*	ት-ኢ *t-ʾi*	-አኪ *-äki*	-ኪ *-ki*
Second	Plural	Female	-ክን *-kən*	ት-ኣ *t-ʾa*	-ኣ *- ʾa*	ት-ኣ *t-ʾa*	-አክን *-äkn*	-ክን *-kn*
Third	Singular	Male	-አ *- ä*	ይ- *y-*	ይ- *y-*	ይ- *y-*	-ኦ *-ʾo*	-ኦ *-ʾo*
Third	Plural	Male	-ኡ *- ʾu*	ይ-ኡ *y-ʾu*	ይ-ኡ *y-ʾu*	ይ-ኡ *y-ʾu*	-ኦሙ *-ʾomu*	-ኦሙ *-ʾomu*
Third	Singular	Female	-ኤት *-ʾet*	ይ- *y-*	ይ- *y-*	ይ- *y-*	-ኣ *-ʾa*	-ኣ *-ʾa*
Third	Plural	Female	-ኣ *-ʾa*	ይ-ኣ *y-ʾa*	ይ-ኣ *y-ʾa*	ይ-ኣ *y-ʾa*	-ኦን *-ʾon*	- ኦን *-ʾon*

In contrast, object markers are consistently used as suffixes, attached to the verbal stem following the subject marker. The object marker affix cannot appear without the presence of the subject marker. Table 2.11 shows the object markers.

Table 2.31: Object markers

Person	**Object marker suffix**
1PSG	-ኒ *-ni*, -አኒ *-äni*
1PPL	-ነ *-nä*, -አነ *-änä*
2PSGM	-ከ *-kä*, -አከ *-äkä*
2PPLM	-ክሙ *-kəmu*, - አክሙ - ä*kmu*
2PSGF	-ኪ *-ki*, -አኪ *-äki*
2PPLF	-ክን *-kn*, -አክን *-äkn*
3PSGM	-ኦ - *ʾo*, -ሁ *-hu*, -˙ዎ *-wo*, ዮ - *yo*
3PPLM	-ኦሙ *-ʾomu*, -ሆሙ *-homu*, ዎሙ *-womu*, ዮሙ *-yomu*
3PSGF	-ኣ *-ʾa*, -ሃ *-ha*, -ዋ *-wa*, -ያ *-ya*
3PPLF	-ኦን *-ʾon*, -ሆን *-hon*, -ዎን *-won*, -ዮን *-yon*

2.4.3.4 Negation

In Ge’ez, negating a verb typically involves adding the prefix ኢ- *ʿi*, as described by Adihana (Memhir, 2015), Kifle (Aleka, 1956), Lambdin (1978), and Mercer (1961). However, there is an exception when dealing with jussive-type verbs in the second person. For these verbs, negation is achieved by using the prefix ኢት- *ʿitə* (Adihana (Memhir), 2015). For example, consider the verb ቅትል - *qtl*, which means 'kill' in the second person singular male form. To negate it, you would add the prefix ኢት- *ʿitə*, resulting in ኢትቅትል - *ʿitəqtl*, meaning 'don’t kill’. For all other verbs, the standard negation prefix is ኢ- *ʿi.* For instance, the verb ቀተለ - *qätälä*, which means 'he killed,' would be negated by adding the prefix ኢ- *ʿi.*, resulting in ኢቀተለ - *ʿiqätälä*, meaning 'he did not kill.' The negative marker is always inserted at the beginning of the verb.

2.4.3.5 Affixation

As discussed earlier, Ge’ez verb formation follows a process that begins with the insertion of vowels into root consonants (stems). Affixes, including prefixes, suffixes, and circumfixes, are then added to these stems to create a complete verb in its surface form. This process is detailed in the works of Desta (2010), Lambdin (1978), and Mercer (1961).

A Ge’ez verb typically includes a sequence of morphemes, which may consist of a prefix, the negation morpheme ኢ- *ʿi*, followed by the subject marker, the derived verb morpheme, the verbal stem, the subject marker again, and finally the object marker. For example, consider the verb ኢይኅድገኒ - *ʿiyəḫädgäni*, meaning 'he will not leave/desert me’. This word is a combination of several morphemes, each serving a specific role:

- *ʿi* -negative marker
- *y*- subject marker
- *ḫädg* -verbal stem
- *äni* -object marker

As mentioned earlier, these affixes are added to Ge’ez verbal stems to construct the surface form of a Ge’ez verb. The specific arrangement of these affixes—whether they function as prefixes, suffixes, or circumfixes—depends on factors such as tense, mood, subject, and object markers.

In this study, the Ge’ez roots, negative, subject, and object markers, as well as derived-verb prefix morphemes and the rules governing the concatenation of these morphemes, are clearly defined in the lexical file. This lexical definition is then employed in the Ge’ez morphological analyzer to effectively combine roots with affixes.

2.4.4 Phonological Alternation

In the Ge’ez language, phonological alternations occurs during the process of affixation. These alternations involve changes in vowels and may also result in the removal of consonants. The occurrence of phonological alternations depends on various factors, including the type of head verb, the tense-mood, and whether the verb is simple or derived. These phonological alternations occur in the following cases:

1. When subject marker prefixes such as እ- (*ə*), ን- (*n*), ት- (*t*), or ይ- (*y*) are combined with derivational verb morphemes like አ- (*ä*), አስተ- (*ästä*), or ተ- (*tä*). Table 2.12 shows the phonological alternations.

Table 2.12: Phonological alternation

Subject marker prefix	**Derived verbs morphemes**		
	ä	*ästä*	*tä*
ə	*əä→ ä*	*əäste→ ästä*	*ətä→ ət*
n	*nä→ na*	*näste→ nastä*	*nt→ nt*
t	*tä→ ta*	*täste→ tastä*	*ttä→ tt*
y	yä→ ya	yäste→ yastä	ytä→ yt

2. When the root of the verb contains the letters እ- (*ə*), ዕ- (*’ə*), ሀ- (*h*), ኀ- (*ḫ*), or ሐ- (*ḥ*) either in the first, second, or third radical of a tri-radical root. Table 2.13 shows the phonological changes.

Table 2.43: Phonological alternation

Roots containing *ə, ’ə, h, ḫ* or *ḥ*	**Verb type** **& tense-mood**	**Changes**
Beginning with *ə, ’ə, h, ḫ* or *ḥ*	Indicative for verb types: - *qätälä* - *gäbrä* - *qäddäsä*	*(n,t,y-) → (n,t,y)ä -*
Containing *ə, ’ə, h, ḫ* or *ḥ* in the middle	Indicative for verb types: - *qätälä* - *śemä* - *qomä*	C1 C2 C3→ C1 C2 C3, (If C1=w, then w *ä* C2 C3→ w C2 u C3)
	subjunctive and jussive for verb types:	C1 C2 C3→ C1 C2 *ä* C3

Roots containing *ə, ʾə, h, ḫ* or *ḥ*	**Verb type** **& tense-mood**	**Changes**
	- *qätälä* - *śemä* - *qomä*	(If C1 = w and C2 not *ə, ʾə,* *h, ḫ* or *ḥ* then w C2 C3→ w C2 *ä* C3)
	Perfective for verb type: - *baräkä* Indicative - *baräkä*	C a C *ä* C→ C a C C C a C C→ C *ä* C C
Ending with *ə, ʾə, h, ḫ* or *ḥ*	Perfective for verb type: - *qäddäsä* - *śemä* - *qomä*	C *ä* C *ä* C→ C *ä* C C
	Indicative for verb types: - *qäddäsä*	C C e C *ä* C→ C C e C a C
	subjunctive and jussive for verb types: - *gäbrä*	C C *ä* C→ C C a C
Containing *h* in the middle	Indicative for verb types	C h C→ C C
Containing ə and *ʿə* in the middle	Indicative for *rəyä* verb:	C ə C→ C ä ə C C ʾə C→ C e ʾə C
Beginning with *ə*	For *äəmärä* verbs types	*(y, t, n, ə) ə ä ə ä →* *(y, t, n, ə) ä ə ä*

3. When the root of the verb contains the letters ው- (*w*) or ይ- (*y*), but not double consonants like ው- (*w*) or ይ- (*y*). Table 2.14 shows the phonological changes

Table 2.54: Phonological alternation

Changes in Verbal stems containing *w or y* **between consonants**	**Verbs ending with** *äw, w or y*
qome verb types C *ä w ä* C→ C *o* C C *ä w* C C→ C *o* C C C *w* C C→ C *o* C C C *w ä* C2→ C *o* C2 C *w* / [C6 — C5] → C *o* [C6 — C5] C *w* C6→ C *a* C6	- *w→ u* - *äw→ o*
ʾsEme verb types C e y e C→ C *E* C C *e y* C C→ C *E* C C C *y* C C→ C *E* C C C *y* C→ C *i* C	- *y→ i*

Changes in Verbal stems containing *w or y* **between consonants**	**Verbs ending with** *äw, w or y*
Roots beginning with *w*	Indicative, subjunctive and jussive for verb type: *qetele w* C C→ C C *gebre* - *w* C C→ C C

4. When the derivational verb morpheme ት- (*t*) is combined with verbal stems that begin with ስ- (*s*), ት- (*t*), ይ- (*y*), ድ- (*d*), ጥ- (*ṭ*), or ጽ- (*ṣ*) in indicative, subjunctive, or jussive tense-moods. Table 2.15 shows the phonological changes.

Table 2.65: Phonological alternation

Subject marker prefix morpheme + Derived-verb prefix morpheme	**Verbal stem beginning with** *s, t, y, d, ṭ, or ṣ*
ət	*ət + [s, t, y, d, T or ṣ] → ə[s, t, y, d, ṭ or ṣ]*
nt	*nt + [s, t, y, d, T or S] → n[s, t, y, d, ṭ or ṣ]*
tt	*tt + [s, t, y, d, T or S] → t[s, t, y, d, ṭ or ṣ]*
yt	*yt + [s, t, y, d, T or S] → y[s, t, y, d, ṭ or ṣ]*

5. When subject marker prefixes are attached to verbal stems that begin with ኣ- (*ä*), ዐ- (*’ä*), ሀ- (*hä*), ኀ- (*ḫä*), or ሐ- (*ḥä*). Table 2.16 shows the phonological alternations.

Table 2.76: Phonological alternation

Subject marker prefix	**Verbal stem beginning with** *ä, ’ä, hä, ḫä or ḥä*
I	*I + [hä, ḫä or ḥä] → a[hä, ḫä or ḥä]*
n	*n + [ä, ’ä, hä, ḫä or ḥä] → nä[ä, ’ä, hä, ḫä or ḥä]*
t	*t + [ä, ’ä, hä, ḫä or ḥä] → tä [ä, ’ä, hä, ḫä or ḥä]*
y	*y + [ä, ’ä, hä, ḫä or ḥä] → yä[ä, ’ä, hä, ḫä or ḥä]*

6. When a verbal stem ending with the ክ- (*k*), ግ- (*g*), ቅ- (*q*), or ን- (*n*) is combined with a subject marker morpheme beginning with the letter ክ- (k). Table 2.17 shows the phonological alternations.

Table 2.87: Phonological alternation

Subject marker suffix	**Verbal stem ending with** *k, g, q or n*
ku	*[k, g or q] + ku→ [k, g or q]u*
kä	*[k, g or q] + kä→ [k, g or q]ä*
kmu	*[k, g or q] + kmu→ [k, g or q]mu*
ki	*[k, g or q] + ki→ [k, g or q]i*
kn	*[k, g or q] + kn→ [k, g or q]n*
n	*[n] + nä→ nä*

In Ge'ez verbs, phonological alternations, including alterations in vowels and the removal of consonants from the verbal stem, are observed during the process of affixation. To account for these phonological changes, a specific rule has been established to govern the transformations in the verbal stem when affixes are added. This rule has been effectively integrated into the alternation rule component of the Ge'ez morphological analyzer.

2.4.5 Irregular Verbs

In the Ge'ez language, there are irregular verbs whose conjugation patterns differ from those of the head verbs within their respective categories. These irregularities often arise due to the presence of specific consonants such as *አ*-ä, *ዐ*-'ä, *ሀ*-h, *ኀ*-ḫ, *ሐ*-ḥ, *ው*-w, and *ይ*-y in the root of the verb. Section 2.4.4 outlines the changes that occur in verbs due to the presence of these consonants. However, there is one exceptional irregular verb, ቤለ (belä), which consists of two consonants and possesses a conjugation pattern distinct from all the head verbs. While Washera (Kifle, 1956) categorizes ቤለ (belä) as one form of the head verb ብህለ (bəhlä), it is important to note that ቤለ (belä) exhibits irregularities not found in the ብህለ (bəhlä) head verb.

In this study, even though two-consonant roots are typically considered as three consonants (as discussed in section 2.4.1), ቤለ (belä) is treated as a separate irregular verb due to its unique features. Table 2.18 shows the conjugation pattern of ቤለ - *belä*.

Table 2.98: ቤለ -*belä* conjugation pattern

	1PSG	**2PSGM**	**3PSGM**
Perfective	እቤ-*əbe*	ትቤ-*təbe*	ይቤ-*yəbe*
Indicative	እብል-*əbl*	ትብል-*təbl*	ይብል-*yəbl*
Subjunctive	እበል-*əbäl*	ትበል-*tbäl*	ይበል-*yəbäl*
Jussive	እበል-*əbäl*	በል-*bäl*	ይበል-*yəbäl*

As shown in Table 2.18, the conjugation pattern of ቤለ (belä) exhibits several irregularities, in addition to being a two-consonant root. These irregularities are as follows:

- Unlike other head verbs, where the subject marker for perfective type verbs is a suffix, ቤለ (belä) has the subject marker as a prefix for perfective type verbs.

- The second consonant of the root is dropped for the perfective type verbs in the first person singular and plural, second person singular male and female, and third person singular male and female forms, resulting in a change from C V C to C V.

2.4.6 Verb Formation

In the Ge’ez language teaching, the first step is to list the head verbs. Following the identification of the head verbs, verb formation starts by listing the six simple verbs, namely, perfective, indicative, subjunctive, jussive, Gerundive and Infinitive (Adihana (Memhir), 2015; Desta 2010; Lambdin, 1978; Mercer, 1961). Subsequently, each of these simple verbs is inflected using the ten pronouns (subject), resulting in a total of sixty verbs with subject markers.

With the exception of the gerundive and infinitive simple verbs, the other simple verbs can further be inflected to indicate an object. Among the ten pronouns, the four pronouns representing third-person subjects (both singular and plural for male and female) can be combined with an inflectional verb that includes an object marker, resulting in a total of 40 verbs. The first person singular and plural pronouns can each be inflected in 8 different ways, bringing the total to 16 verbs. Similarly, the second person singular and plural pronouns, both male and female, can each be inflected in 6 different ways, totaling 24 verbs. In summary, a verb with a simple verb formation can yield a total of 380 inflected simple verbs.

In this chapter, we have discussed Ge’ez language verbs, highlighting their concatenative and non-concatenative features in Ge’ez verb formation. Our study utilizes finite-state technology, a rule-based approach, for implementing the Ge’ez verb morphological analyzer. This rule-based methodology draws on linguistic theory and employs linguistically motivated rules to analyze words.

For this study, we organized Ge’ez verb roots, vocalic patterns for the intercalation of roots with vowels, and a list of affixes through consultation with Ge’ez experts. Furthermore, we manually identified a set of rules that define how Ge’ez verbs are formed from their roots.

2.5 Chapter Summary

This chapter commenced by introducing the Ge’ez language in general, including its alphabet. While IPA transliteration of Ge’ez is used for this research paper, SERA is employed for Ge’ez transliteration in the implementation of the Ge’ez morphological analyzer. The Ge’ez language features both concatenative and non-concatenative elements in its word formation. Consequently, root-pattern morphology was applied to address its non-concatenative characteristics, while affixation played a crucial role in handling the concatenative aspects of word formation.

This chapter also explored Ge’ez verb classifications, with a focus on the washära Ge’ez verbal classification, which was chosen for implementation in the Ge’ez morphological analyzer. This decision was made because the troops within this classification adhere to the conjugational pattern and form of their respective heads. Furthermore, the chapter discussed phonological changes resulting from the assimilation of affixes and discussed how certain letters within verb roots can give rise to irregularities in verb forms. Notably, the two-consonant verb ቤለ - belä was examined separately due to its irregular features in verbal formation.

These discussions laid the foundation for the design of the Ge’ez morphological analyzer, encompassing the Ge’ez alphabet, verb formation processes, and the irregularities associated with certain letters and the two-consonant verb ቤለ (belä).

The next chapter discusses the computation morphology and provides a literature review of studies done on Amharic and Ge’ez languages.

.

Chapter 3 – Literature Review

3.1 Introduction

The literature study in this chapter provides a brief introduction to Natural Language Processing (NLP) and the utilization of morphological analyzers in various natural language applications. A morphological analyzer constitutes a fundamental component of NLP, finding its applications in tasks such as machine translation, parsing, and automatic dictionaries. This section presents an overview of the diverse approaches employed in morphological analysis. Subsequently, a discussion follows concerning the endeavors undertaken in the realm of morphological analysis within Semitic languages, with a particular focus on Ge'ez and Amharic languages.

3.2 Approaches to Morphological Analysis

Language is an important means pf communication between human beings. We use language in our day-to-day life to communicate with others and to express our ideas.

NLP or human language processing is a field that aims to use the computer to perform important tasks involving human language such as human-machine communication, human-human communication or processing of text and speech (Trost, 2003). NLP deals with both language analysis and language generation. Language analysis deals with the processing of a given stream of text to provide meaning while language generation deals with producing a meaningful text from some form of representation based on the language information.

Dale (2010) states that there are a number of stages of analysis in NLP:

- tokenization – deals with the breaking up of sequences of texts into words
- lexical analysis – deals with morphological analysis of a word where it breaks the word into morphemes together with structural information about the word
- syntactic parsing – provides structural description suitable for semantic analysis
- semantic analysis – deals with the meaning of the words in a sentence
- pragmatic analysis – determines the meaning of the words in a context

The output of one stage serves as an input to the subsequent stage. For instance, the output of tokenizer will serve as an input for morphological analyzer. Morphological analyzer deals with morphemes, the smallest units of meaning, which remains the same across words. This provides NLP system to understand the meaning of a particular word and then use other structural information to determine the meaning of a sentence. One of the major components for many NLP application, especially for systems that involve parsing and / or generation of natural languages in written and spoken form (Jurafsky & Martin 2008), is a morphological analyzer. Hence, morphological analyzer is an important component of NLP.

According to Kazakov (2001), methods for word segmentation, also known as word morphology, can be categorized into two primary paradigms: rule-based and data-driven. The rule-based methodology is based on linguistic principles and employs rules that are grounded in linguistic theory to analyze words. In contrast, data-driven approaches use the text data (corpus) to learn how to analyze or segment the words with little or no consideration about the knowledge of the language. The data-driven approach relies significantly on machine learning and statistical techniques (Liddy, 2001). On the contrary, the rule-based approach finds its roots in the concept of two-level morphology (Koskenniemi, 1983).

Both the rule-based and data-driven methodologies find applications in the development of morphological analyzers for languages across the globe. Finite-state morphological tools and techniques play a widespread role in the analysis of diverse languages, including Semitic languages like Arabic (Beesley, 1998), Hebrew (Yona & Wintner, 2008), and Amharic (Amsalu & Gibbon, 2005). The Amharic language, one of the Ethiopian Semitic languages that does not have its own alphabet and uses the Ge’ez language alphabet, has a similar morphology and verb inflections as the Ge’ez language. Due to these linguistic similarity, research conducted on Amharic language computational morphology is discussed along with research on the Ge’ez language computational morphology.

3.3 Amharic

Numerous endeavors have been undertaken to develop a morphological analyzer for Amharic, employing both data-driven and rule-based approaches. Within the data-driven domain, the unsupervised learning approach (Bayou, 2002) and supervised learning approaches (Gasser and Mulugeta, 2012; Abate and Assabie, 2014) were adopted to develop Amharic morphological analyzers.

Bayu (2002) developed an Amharic morphological analyzer using unsupervised learning approach for the concatenative morphology of Amharic and using the theory of auto-segmental morphology to analyze the non-concatenative morphology of Amharic. The utilized corpus encompasses 5236 items of corpus data to learn the Amharic morphology using Linguistica, the freely available language-independent tool. However, the limitation of Linguistica in accommodating Amharic's non-concatenative features led Bayou (2002) to create a supplementary stem analyzer, named Amharic Stem Morphological Analyzer (ASMA). The role of the ASMA was to analyze the stems to their respective root and pattern constituents. The Amharic morphological analyzer employs a two-step process: initial segmentation into prefixes, stems, and suffixes using Linguistica, followed by further analysis of Linguistica's output stems via ASMA to derive roots and their corresponding pattern morphemes. However, it is essential to clarify that the output stems generated by Linguistica were not used as input for ASMA, as previously suggested. Instead, separate datasets were utilized for Linguistica and ASMA. This approach led to a successful parsing of 87% for the test data (500 words) by Linguistica and a correct analysis of 97% for the test data (255 words) by ASMA. The inadequacy of Linguistica in producing linguistically accurate stems necessitated the utilization of distinct corpora. This could potentially be attributed to Bayou (2002) adhering to Linguistica's minimum recommended training corpus size (5,000 – 1,000,000). This sheds light on the significance of corpus size, particularly when dealing with highly inflectional languages like Amharic, within the realm of data-driven approaches. Additionally, another notable obstacle involved the inherent separation of the two systems, which prevented seamless integration.

The study by Gasser and Mulugeta (2012) focused on using a supervised machine-learning approach to analyze the morphology of Amharic verbs. They utilized the CLOG tool, which is a Prolog-based Inductive Language Programming, to learn decision lists or rules based on positive examples only. Training the CLOG involved the manual annotation of 216 Amharic verbs, encompassing all possible tense and subject markers or features. Following the training phase, the CLOG produced a total of 108 rules for affix extraction, 18 rules for root template extraction, and an additional three rules governing internal stem alternation. Subsequently, the analyzer underwent testing using 1784 verb forms, resulting in a commendable accuracy rate of 86.99%. However, it is essential to note that the scope of this morphological analyzer is confined to Amharic verbs with subject marker affixes. Additionally, the researchers had to annotate all possible combinations of subject and tense in the training set, which not only make the process time-consuming but also presents challenges in terms of its broader implementation.

Another data-driven approach employed in the development of an Amharic morphological analyzer through a supervised learning framework is exemplified in the work conducted by Abate and Assabie (2014). In their study, they harnessed the memory-based learning methodology, specifically employing the IB1 and IGtree algorithms. In constructing their methodology, the researchers compiled a morphological dataset that encompassed Amharic verb stems, as well as the shared attributes of all morphological functions pertaining to Amharic nouns, and the grammatical features exhibited by all morphemes. For the training phase, Abate and Assabie (2014) annotated a training dataset comprising 1,022 items, with 841 of those items being verbs and 181 classified as nouns. The outcome of this training yielded 26 distinct class labels, totaling 1,356 instances of nouns and 6,719 instances of verbs. The evaluation process, conducted by Abate and Assabie (2014), involved employing the 10-fold cross-validation technique using both the IB1 and IGtree algorithms. The obtained results showed an overall accuracy rate of 93.6% for IB1 and 82.3% for IGtree. Abate and Assabie (2014) chose to work with the stems of the verbs rather than the roots, and they annotated sample word stems along with their corresponding patterns. This approach was adopted due to the intricate nature of Amharic verbs. Given the language's high degree of inflection, particularly within the domain of Amharic verbs, the application of a data-driven methodology for developing a morphological analyzer necessitates a substantial dataset. Additionally, the task of annotating all conceivable inflections of verb types poses a considerable challenge.

Researchers have employed a rule-based approach in the development of Amharic morphological analyzers, as evidenced by the works of Argaw and Asker (2007), Amsalu and Gibbon (2005), Amsalu and Demeke (2006), and Gasser (2011).

An Amharic stemmer was developed using a rule-based approach by Argaw and Asker (2007), aiming to transform words into their citation forms for the purpose of dictionary lookup. Within their study, the researchers employed a rule-based strategy to segment words into all potential stems or segments, utilizing statistical methods to address segmentation ambiguities. Argaw and Asker (2007) constructed an extensive collection of 65 rules, encompassing the entirety of Amharic morphology. These rules consisted of straightforward affixation guidelines, accommodating the arrangement of prefixes and suffixes for different word categories and sets of affixes. During the stemmer's operation, it first generated various segmentations of a given word based on the morphological rules. Each candidate segmentation underwent validation against a machine-readable dictionary. If a single stem matched the dictionary, that particular segmentation was adopted as the stemmer's output. Alternatively, in cases where multiple matching stems emerged, the most probable stem was selected after disambiguation. Following their evaluation, Argaw and Asker (2007) reported an accuracy rate of 60% for the test set extracted from the Amharic novel "ፍቅር እስከ መቃብር" (Fikir Iske Meqabir), and a higher accuracy rate of 75% for the test set acquired from a news article. The researchers devised a set of 65 rules that drew upon the entirety of the Amharic language's morphology. However, their limitation of focusing these morphological rules solely on affixes contributed to their relatively lower outcomes. Expanding the scope of rule construction would lead to a heightened output in word segmentation, consequently enhancing the probability of achieving a match within the dictionary.

A finite-state-based morphological analyzer was developed for Amharic, encompassing words from all parts of speech (POS). Amsalu and Gibbon (2005) created this analyzer using the XFST tool. Notably, this morphological analyzer addressed both the concatenative and non-concatenative aspects inherent to the Amharic language. Addressing the intricate root-pattern morphology of Amharic verbs, the researchers incorporated vocalic patterns between root consonants by locating appropriate positions within consonant sequences. Furthermore, the morphological analyzer extended its coverage to include words from all parts of speech. In their evaluation, Amsalu and Gibbon (2005) reported precision levels of 94% for nouns, 81% for adjectives, 91% for adverbs, and 54% for verbs based on a test set comprising 1,620 words. The observed lower precision in Amharic verb analysis can be attributed to the insufficiently comprehensive definition of rules specifically applicable to verbs. This highlights the critical role of comprehensive rule definitions in achieving accurate and reliable results within the context of a rule-based approach to morphological analysis. Nonetheless, this research marks a pioneering effort in creating a morphological analyzer that encompasses all parts of speech in Amharic, achieved through the utilization of a finite-state method. This significant achievement not only underscores the feasibility of employing finite-state methods in the development of morphological analyzers for languages like Amharic, but also signals the potential success of such methods for similar languages such as Ge’ez.

One of the main challenges encountered when striving to develop a comprehensive morphological analyzer for Amharic is the inherent limitation in creating proper and comprehensive rule definitions, along with a deficiency in a comprehensive lexicon. In response to these challenges, Amsalu and Demeke (2006) undertook the development of a morphological analyzer tailored specifically for non-concatenative simple Amharic verbs, utilizing the XSFT tool. In their endeavor, Amsalu and Demeke (2006) revealed that they were able to generate approximately 6,400 simple verb stems from a collection of 1,300 roots that spanned both regular and irregular verbs. The core objective of their research was to establish a fully functional morphological analyzer, with a progressive approach beginning by focusing on simple verbs. This step-by-step methodology aimed to lay the foundation by successfully addressing the analysis of straightforward verbs first, paving the way for subsequent advancements towards realizing a complete morphological analyzer for the Amharic language.

Similarly, Gasser (2011) made substantial contributions by developing a finite-state morphological analyzer named HornMorph for three Ethiopian languages: Amharic, Tigrigna, and Oromigna. Among the three languages, two belong to the Semitic language family, while Oromigna represents a Cushitic language. Separate finite-state transducers (FSTs) was developed for each language. Each language was assigned its own distinct finite-state transducer (FST). To address the non-concatenative aspect of the two Semitic languages, Gasser (2011) adopted a weighted FST approach. The lexicons for all three languages were compiled from online dictionaries. In the case of Amharic, the lexicon encompassed 1,851 verb roots and 6,471 noun stems. The Tigrigna lexicon consisted of 602 root verbs, and the Oromo lexicon included 4,112 verb roots and 10,659 noun stems. In terms of evaluation, HornMorph underwent testing using 200 Tigrigna verbs, 200 Amharic verbs, and 200 Amharic nouns and adjectives. Gasser (2011) reported high levels of accuracy in the results: 96% accuracy for Tigrigna verbs, 99% accuracy for Amharic verbs, and 95% accuracy for Amharic nouns and adjectives. An important aspect to note is that HornMorph is available for free testing on the World Wide Web. This research underlines the successful application of finite-state methods in the development of morphological analyzers for Semitic languages like Amharic and Tigrigna, which in turn implies the potential utility of such methods in constructing morphological analyzers for languages such as Ge’ez.

Various methods, including statistical, machine learning, and finite-state based approaches, have been employed in the morphological analysis of the Amharic language. Despite this, a comprehensive morphological analyzer for Amharic is still not available. Several factors contribute to the lack of a fully realized Amharic morphological analyzer, including the absence of a comprehensive lexicon for Amharic, the incomplete definition of morphological rules in rule-based approaches, and insufficient training corpora in data-driven approaches. However, the collaboration of researchers in this field holds the potential to the development of a fully-fledged morphological analyzer for Amharic

3.4 Ge’ez

In contrast to the Amharic language, Ge’ez stands as one of the lesser-studied languages. Currently, Ge’ez is exclusively utilized within the Ethiopian and Eritrean Orthodox Tewahido Church. The church maintains its own educational framework (Challiot, 2009) for teaching the Ge’ez language, its grammar, as well as its associated literature, known as ቅኔ-*qəne*. Furthermore, the educational curriculum covers spiritual singing, referred to as ዜማ-*zema*, physical expressions of devotion termed አቋቋም-*äkuwäkam*, and the spiritual teachings of the Bible. It is important to note that the teaching of the Ge’ez language remains confined to the domain of the Church.

Studies have been conducted on the Ge’ez language, concerning the classification of Ge’ez verbs and morphological analysis. One noteworthy study has focused on the classification of Ge’ez language within the context of the Ethiopian Orthodox Tewahido Church. Andualem (2007) conducted a significant study focused on the classification of Ge’ez verbs within the context of the Ethiopian Orthodox Tewahido Church. The investigation delved into the practices of prominent ቅኔ - *qəne* schools within the Church, including ዋሸራ - *washära*, ጎንጂ - *gonji*, and ዋልዳ - *walda*. While numerous churches have educational institutions dedicated to teaching various aspects of the Church's teachings, certain institutions, like the aforementioned Ge'ez schools, are specifically esteemed for cultivating advanced scholars in the domain of Ge'ez language literature ቅኔ - *qəne*. Andualem's study (2007) explained upon the manner in which Ge'ez verbs, characterized as "heads" and "troops," are organized into distinct groups based on their conjugational forms. Model verbs, referred to as "heads," possess the capacity to represent other verbs within their categorical domain, whereas "troops" are verbs that adhere to the inflectional and derivational patterns set by the head verb of their respective category. According to the conventions of the three traditional Ge'ez schools, the classification of Ge'ez verbs may encompass six to eight head verbs (Andualem, 2007). For a troop verb to be classified under a specific head verb, it must mirror the pattern of letters of that head verb (Andualem, 2007; Sewasew, 1993).

For this research, the Ge'ez verb classification methodology of the Washera school was adopted. This choice was made due to the alignment of washära's Ge'ez verb classification with uniform conjugation patterns within the same category, rendering it suitable for computational morphology. Despite the inherent complexity and inflectional nature of Ge'ez verbs, their formations adhere to regular patterns, thereby rendering them rendering them compatible with computational morphology.

The initial Ge'ez morphological analyzer (Desta, 2010) was created with a focus on a specific Ge'ez head verb. This Ge'ez analyzer was devised using a rule-based methodology, integrating the two-level morphology framework in conjunction with a CV-based approach. The analyzer comprised two fundamental components: a knowledgebase and linguistically motivated rules. In the developmental process, Desta (2010) employed Java NetBeans IDE 6.7.1 to construct a prototype for evaluating the algorithm's precision. The accuracy of the algorithm was assessed through testing, using a dataset manually collected from the Ethiopic New Testament Ge'ez Bible. Within this dataset, 415 verbs from the designated head verb category qtl were selected for testing. The results of the experimentation revealed that the analyzer demonstrated a 73.98% accuracy rate. Out of the 415 verbs in the test set, a total of 307 verbs were accurately analyzed, as reported by the researcher. It is crucial to emphasize that Desta (2010) took a pioneering step in his research by successfully creating a morphological analyzer specifically designed for a Ge'ez head verb. However, it is worth acknowledging that the research focused on one particular head verb. Furthermore, it is noteworthy that the Ge'ez morphological analyzer developed in the study is primarily designed for analysis tasks and lacks the capacity to generate verbs using a given root and its associated structural details. This distinction highlights that while the research achieved significant progress in the field, its scope remained restricted to the analysis of a single verb category, without encompassing the functionality of verb generation from a given roots and its structural information.

In the development of a Ge’ez morphological analyzer encompassing all Ge’ez verb categories, a data-driven approach was undertaken by Abate (2014). Abate (2014) employed a memory-based supervised machine-learning methodology, utilizing the IB2 and TRIBL2 algorithms. The study involved the manual annotation of a dataset comprising 11,105 items, totaling 12,135 instances, and encompassing 31 distinct class labels. Within this dataset, 90% of the manually annotated data was allocated for training the analyzer, while the remaining 10% was reserved for testing purposes. To facilitate the training and testing of the dataset, Abate (2014) employed the Tilburg memory-based learner (Tilburg), a software package developed and maintained by the Induction of Linguistic Knowledge group at Tilburg University. The actual implementation of the analyzer was realized using the Python programming language. Upon testing, the analyzer demonstrated an overall accuracy of 93.24% with optimized parameters using the IB2 algorithm, and 92.31% with the TRIBL2 algorithm. Furthermore, the optimized parameters using IB2 yielded an overall precision of 55.6%, recall of 56.3%, and an F-score of 59.95%. Similarly, employing the TRIBL2 algorithm, the precision, recall, and F-score were calculated as 58.8%, 60.3%, and 59.54%, respectively.

It is indeed crucial to highlight certain key aspects of the study. One of the points to note is that the research failed to account for verbs that undergo phonological changes, a significant factor within Ge’ez language analysis. Additionally, it is important to emphasize that the Ge'ez morphological analyzer's capabilities are confined to analysis tasks only. It lacks the functionality to generate verbs based on provided root forms and the associated structural information. Despite the intention of creating an analyzer encompassing all verb categories, the dataset employed in the study (Abate, 2014) predominantly consisted of a comprehensive derivation of just one verb category, supplemented by sample verbs from other categories. This dataset composition significantly influenced the outcomes of the analysis. One of the requirements of the supervised learning approach is the availability of ample annotated text to ensure accurate predictions for unknown test data. The data-driven approach for morphological analysis carries the drawback of necessitating a substantial volume of data that is both relevant and representative of the subject matter. Moreover, the supervised learning approach mandates the presence of an annotated text corpus, which is regrettably lacking for the Ge’ez language.

The existing efforts in developing Ge'ez morphological analyzers are associated with notable challenges and limitations:

1. **Limited to Analysis:** The Ge'ez morphological analyzers currently available are confined to the analysis of verbs and lack the capability to generate verbs.
2. **Rule-Based Analyzer's Focus:** The rule-based Ge'ez morphological analyzer developed by Desta (2010) is primarily centered around the analysis of a single Ge'ez head verb. This restricts the analyzer's scope, as it does not cover the entirety of verb categories within the language.
3. **Data-Driven Analyzer's challenge:** The data-driven Ge'ez morphological analyzer introduced by Abate (2014) lacks the ability to analyze verbs with irregular conjugations. Furthermore, the limitation of the dataset, which encompasses a detailed derivation of only one head verb category and a mere sample of other head verb categories, hinders the analyzer's performance in predicting unknown data for verbs in other categories.

In the context of a language characterized by intricate morphology like Ge'ez, the rule-based approach emerges as an effective means of representation. One of the notable attributes of the rule-based approach is its strong foundation in linguistics. The advantage of its linguistically motivated nature lies in the clear definition of all morphological rules, enabling a comprehensive understanding of the language's intricacies. Therefore, for the development of a Ge'ez morphological analyzer, a rule-based approach utilizing finite-state tools and techniques has been chosen. The finite-state methodology possesses certain compelling attributes, chief among them being its bidirectional capabilities and inherent simplicity. Furthermore, these techniques have demonstrated successful outcomes in constructing morphological analyzers for languages spanning from widely spoken commercial languages to less-studied languages across the globe (Beesley, 2004).

Although attempts have been made in creating Ge'ez morphological analyzers (Desta, 2010; Abate, 2014), the development of a comprehensive morphological model for the Ge'ez language is still in progress. Consequently, the focus of this study lies in harnessing the potential of the finite-state approach to develop a morphological analyzer tailored to Ge'ez verbs. A prominent distinction in our Ge'ez morphological analyzer is its dual functionality. Unlike previous works, our analyzer is not solely confined to word analysis but also generating words based on given root forms along with their associated structural information. What's more, our analyzer is designed to encompass all categories of Ge'ez verbs.

This research project employed the finite-state technique and associated tools to construct a finite-state Ge’ez verb morphological analyzer and generator. The resulting tool encompasses the following key attributes:

- the rule-based approach has been implemented to develop the morphological analyzer for all categories of Ge’ez verbs.
- the well-established finite-state tools and techniques, that have demonstrated their efficacy across numerous languages even those with non-concatenative features, were employed to develop the Ge’ez verb morphological analyzer
- the development of the Ge’ez morphological analyzer made use of the freely available Foma tool.
- the morphological analyzer exhibits the capacity to perform both the analysis and generation of Ge’ez verbs.

3.5 Chapter Summary

NLP encompasses both language analysis and language generation. Language analysis involves processing a given text stream to derive meaning, while language generation pertains to crafting coherent text from some form of language-based representation. Morphological analysis, a crucial component of natural language applications like machine translation and automatic dictionaries, focuses on morphemes – the fundamental units of meaning that remain consistent across words. The field of morphological analysis is typically approached via two fundamental methods: rule-based and data-driven approaches.

The chapter begins by exploring the two fundamental approaches to morphological analysis: the rule-based approach and the data-driven approach. Subsequently, the discussion transitions to the research conducted on Amharic and Ge’ez languages. Notably, Ge’ez stands as a less-examined language, and the development of a morphological analyzer tool assumes significance as it can effectively contribute to various natural language applications, including machine translation and automatic dictionaries.

Finite-state morphological tools and techniques hold widespread utility in the analysis of diverse languages, including Semitic languages like Hebrew and Amharic. Within the realm of Ge’ez, a finite-state approach has been chosen for the development of the morphological analyzer.

The subsequent chapter discusses the research methodology employed in this study.

Chapter 4 – Research Methodology

4.1 Introduction

In conducting this research, the Design and Creation research methodology was chosen as the guiding approach. This chapter is dedicated to discussing the chosen research methodology and its application within this study. To consider an IT artifact as a valid research contribution, it must possess academic rigor and adhere to the formal methods and principles of system development during its design and creation. This chapter delves into the design and research methodology applied in this research, emphasizing its application in addressing the core research question: 'How can we create efficient finite-state transducers that accurately represent the morphotactics and orthographic rules governing Ge’ez verbs?’

The subsequent sections will delve into the research methodology, explaining how the stages of the design and creation methodology was applied during the development of the Ge’ez verbs morphological analyzer.

4.2 Design and Creation Research Strategy

Research methodology is a framework that clearly states the paradigm, the strategies and tools used in conducting the research. It is essential to adopt a well-established and recognized research methodology to ensure the credibility and rigor of the research process.

Within the realm of information systems research, Henver et al. (2004) identify two primary paradigms: behavioral science and design science. The behavioral science paradigm aims to develop and substantiate theories that explain or predict organizational and human phenomena associated with the analysis, design, implementation, management, and utilization of information systems (Henver et al., 2004, p. 76). In contrast, the design science paradigm centers on problem solving and the creation of innovative IT artifacts (Henver et al., 2004).

Oates (2005) distinguishes six research strategies or approaches that systematically guide research in the field of information systems and computing. These strategies include surveys, design and creation, experiments, case studies, action research, and ethnography. For this research, we have chosen to adopt the design and creation research methodology, as outlined by Oates (2005). The rationale behind this choice is that the primary objective of this research is the development of an IT artifact—a morphological analyzer for Ge’ez verbs.

The design and creation research methodology is centered on the development of an IT artifact as a solution to a research problem, with the intention of contributing to the body of knowledge. These IT artifacts can encompass constructs, models, methods, and instantiations (Creswell, 2003).

According to Henver et al. (2004), design science is a problem-solving process and provides seven guidelines for effective design-science research. It is worth noting that researchers should exercise their judgment in applying these guidelines to the specific research problem at hand. These guidelines include:

1. Design as an artifact – the primary aim of design-science research is to produce an IT artifact that may not necessarily be a complete system but should effectively address the identified research problem.
2. Problem relevance – the IT artifact produced must solve a relevant problem.
3. Design evaluation – evaluation is an important part of a research process. The IT artifact must be evaluated by appropriate evaluation methods for quality, efficiency and usability.
4. Research contributions – the design-science research should contribute to the body of knowledge by creating an artifact that resolves an unresolved problem, following a well-structured research process in its creation, and employing suitable evaluation methods.
5. Research rigor: rigor pertains to the manner in which research is conducted. Appropriate research processes and methodologies must be applied in developing and evaluating the IT artifact.
6. Design as search process – a thorough search should be conducted to discover effective solutions to the problem. The iterative nature of design science allows for the exploration of multiple potential solutions
7. Communication of research – the result of the research must be effectively communicated to all relevant stakeholders.

By adhering to the aforementioned guidelines, researchers can effectively conduct their research, ultimately contributing to the body of knowledge. This approach stands in contrast to typical software development, where the primary goal is often the creation of a functional system without the same academic rigor. Oates (2005) emphasizes that research involving an IT artifact as its output must make a substantial contribution to the body of knowledge to differentiate it from standard software development efforts. To qualify as a research contribution, an IT artifact must possess academic qualities and adhere to the formal methods and principles of system development during its design and creation. Furthermore, the artifact should either introduce something entirely new or improve upon existing products. One key advantage of this approach is that it results in a tangible artifact, serving as proof of the research's value. However, it is imperative for the researcher to establish how the research distinguishes itself from standard artifact development.

Oates (2005) outlines five crucial steps (Vaishnavi & Kuechler, 2004 in Oates, 2005, p.101) which should be followed when employing the design and creation research method. These steps include awareness, suggestions, development, evaluation, and conclusion.

In the context of this research, we have applied the design and creation research methodology to design and develop our IT artifact—the finite-state based morphological analyzer for Ge’ez verbs. This approach revolves around creating an IT artifact that serves as a solution to a research problem and, in doing so, contributes to the body of knowledge in the field of NLP.

4.3 The Application of Research Design and Creation in This Study

This section describes how the research design and creation stages are applicable to this study. The research process is illustrated in Figure 1.1.

4.3.1 Awareness of the Problem

The initial step in the research and creation method involves gaining awareness of the problem at hand. In the context of this study, the problem is the need to develop a morphological analyzer for Ge’ez verbs. Ge’ez is among the world's ancient languages and is relatively understudied. Even though there are no native speakers of the language, Ge’ez is still alive in the Ethiopian Orthodox Tewahido Church (የኢትዮጵያ ኦርቶዶክስ ተዋህዶ ቤተክርስትያን) in Ethiopia.

The Ge’ez language holds a wealth of knowledge, including ancient philosophy, culture, history, and spiritual teachings of Ethiopia. To ensure that this valuable resource and knowledge remain accessible to future generations, the development of NLP applications, such as machine translation and spell-checking, is crucial. Creating a morphological analyzer serves as a fundamental building block for the development of various NLP applications.

Chapter 1 of this research study presents a comprehensive problem statement that underscores the need for a morphological analyzer for Ge’ez verbs.

4.3.2 Suggestions

After becoming aware of the problem at hand, the next crucial stage in the design and creation method is generating suggestions.

Given that the Ge’ez language belongs to the Semitic language group and is highly inflectional, a rule-based approach emerges as an effective means of representing Ge’ez. Consequently, the widely recognized rule-based technology known as finite-state is suggested as the foundation for developing the Ge’ez morphological analyzer. Specifically, a finite-state based morphological analyzer for Ge’ez language verbs is recommended. Through a comprehensive review of existing literature, the following suggestions have been proposed for this study:

- Among the various Ge’ez verb classifications put forth by prominent Ge’ez schools and scholars, the choice is made to adopt the washära Ge’ez school's verb classification as the basis for this research.
- Finite-state technology is selected as the most suitable platform for developing the Ge’ez morphological analyzer
- To facilitate the development of the Ge’ez verb morphological analyzer, the freely available finite-state tool, Foma, is chosen.
- The ultimate goal is to create a comprehensive Ge’ez morphological analyzer capable of performing both morphological analysis and generation for Ge’ez verbs.

Chapter 2 provides an in-depth overview of the Ge’ez language, with a particular focus on Ge’ez verbs. Chapter 3 presents a literature review on computational morphology, with an emphasis on computational morphology within Semitic languages and, more specifically, the Ge’ez language. This chapter reaffirms the rationale behind selecting the rule-based approach based on finite-state technology for the development of the morphological analyzer.

4.3.3 Development

The development stage is a crucial phase where the proposed solution is meticulously examined, designed, developed, and implemented using formal development methodologies (Oates, 2005).

Following the suggested approaches outlined earlier, the next logical step is the actual development of the IT artifact—namely, the Ge’ez verb morphological analyzer. To achieve this, the principles of the adaptive software development method are applied, with a strong emphasis on modularization, unit testing, composition, system testing, and iterative processes.

Using the Foma finite-state tool, finite-state transducers are meticulously crafted for each verb type, and rigorous testing is conducted upon the completion of each module. Subsequently, a composite test is carried out once the finite-state transducers are composed. This comprehensive testing process helps identify errors and inconsistencies, which are promptly corrected. Following error corrections, further rounds of testing are conducted to ensure that the morphological analyzer consistently produces the expected outputs. This iterative approach involves the continuous cycle of developing finite-state transducers, conducting unit testing, composing transducers, and performing composite testing until the final morphological analyzer is achieved.

The primary objective of the IT artifact is to take the surface form of a Ge’ez verb as input and produce the corresponding verb root, along with structural information about the verb. Additionally, the analyzer is designed to perform the reverse operation—generating the surface form of a Ge’ez verb when provided with the root verb and structural information.

Chapter 5 of the research study provides a comprehensive overview of finite-state technology, introducing Foma, and outlining its application in the development of morphological analyzers. Chapter 6 offers a detailed account of the design and development of the Ge’ez verb morphological analyzer.

4.3.4 Evaluation

In accordance with the research and creation strategy, the subsequent phase involves the rigorous evaluation of the IT artifact. In this research, the hypothesis underpinning the Ge’ez morphological analyzer is that it can accurately generate the correct lexeme/root of a word, along with feature tags, from a given surface form of a word, and vice versa.

To rigorously assess the accuracy of the morphological analyzer, a comprehensive test set was painstakingly assembled. This test set was manually organized from the Ethiopic Ge’ez New Testament Bible and the Ge’ez prayer book, known as 'ውዳሴ ማርያም' - wudase maryam. It comprised a total of 1,519 verbs extracted from these books. From this collection, 1,365 unique verbs (non-repeat words) were meticulously selected for the test data set. To ensure the test data set's reliability and completeness, Ge’ez language experts were enlisted. These experts provided the necessary structural and lexical information for each word within the data set. This meticulous organization by experts serves as the foundation for assessing the morphological analyzer's accuracy.

Each word processed by the morphological analyzer is subjected to a rigorous accuracy assessment. The output generated by the analyzer is meticulously compared against the data set. This comparative analysis helps determine the extent to which the analyzer correctly produces the lexeme/root and associated feature tags from a given surface form of a word.

Chapter 7 of this study is dedicated to the exploration of the test data collection process, the meticulous evaluation of the Ge’ez verb morphological analyzer, and the presentation of findings.

4.3.5 Conclusion

The ultimate phase in the research and creation method centers on drawing conclusions and presenting the research outcomes. It is important to clearly specify the contributions that this research made to the body of knowledge on NLP.

In the culmination of this research, our endeavors yield not only valuable insights but tangible contributions to the domain of NLP. Our Ge’ez morphological analyzer demonstrated an impressive 92.7% accuracy in the analysis of verbs, coupled with a precision rate of 80.24%, encompassing all eight head verbs. Furthermore, the Ge’ez morphological analyzer exhibited dual capabilities, enabling both the analysis and generation of Ge’ez verbs. This study assumes its role in advancing the development of a comprehensive Ge’ez morphological analyzer, having successfully created a morphological analyzer tailored specifically for Ge’ez verbs.

The detailed findings of this research, along with recommendations for future research endeavors, are meticulously presented in Chapters 7 and 8 of this study.

4.4 Chapter Summary

This chapter explores the research and design methodology employed in this study. The rationale for choosing this approach is explained, emphasizing the importance of contributing to the body of knowledge when an IT artifact serves as the research output. Additionally, the chapter outlines the five essential steps within the design and creation methodology, demonstrating their application in developing the Ge’ez verbs morphological analyzer.

The subsequent chapter delves into finite-state technology, providing a comprehensive understanding of its relevance and application within the context of this research.

Chapter 5 – Finite--State Technology

5.1 Introduction

This chapter explores the field of morphology, describing the formation of words from morphemes and discussing morphological analysis. Both concatenative and non-concatenative aspects of morphology are explored, with a particular focus on the non-concatenative morphology found in the Ge’ez language. Additionally, this chapter addresses a key sub-question of the research, namely, how the non-concatenative morphology of Ge’ez verbs can be effectively represented using finite-state technology.

Finite-state technology offers a powerful means to describe natural language morphology through regular expressions. These regular expressions can then be compiled into finite-state automata and FST using finite-state tools. Within this context, we introduce finite-state technology and present the finite-state compiler known as Foma, which was utilized in the development of the Ge’ez FST.

The next section introduces the fundamental concepts of morphology and the process of word formation from morphemes.

5.2 Morphology

Morphology is the study of words in a language. Morphology studies how words are formed from the smallest meaning bearing units morphemes. Moreover, morphology tries to determine the rules that govern the formation of words from these morphemes (Jurafsky & Martin, 2008).

5.2.1 Morphemes

Morphemes are the basic building units of words. Morphemes are classified into two types: free morpheme and bound morphemes. A free morpheme is a word that can stand by itself as a meaningful word; the word buy is a free morpheme. On the other hand, a bound morpheme cannot stand by itself and must be combined with another morpheme in order to be a word and cannot by itself give meaningful word. For example, the bound morpheme -er may be combined or attached to another morpheme buy to give a word buyer. The main difference between a morpheme and a word is that a morpheme may or may not stand by itself as a word.

Morphemes may be classified as two basic forms stems and affixes (Trost, 2003). According to Jurafsky and Martin (2008), the suffixes are further classified into four namely- prefix, suffix, infix and circumfix.

- **Prefix** is a morpheme that is added at the beginning of the stem
- **Suffixes** is a morpheme that is added at the end of the stem
- **Infix** is a morpheme that is inserted inside the stem and
- **Circumfix** is a morpheme that is inserted both at the beginning and end of the stem.

Ge’ez stems combine with affixes such as prefix, suffix and circumfix to form words.

5.2.2 Root, Stem and Base

Martin (2002) provides definitions for the linguistic terms root, base, and stem as follows:

- A **base** is the foundational part of a word to which affixes are added. For instance, in the word 'activity,' 'active' is the base, and 'ity' is the suffix. However, it's worth noting that even the base 'active' can be further divided into 'act' and 'ive' (Martin, 2002).
- A base that cannot be divided or analyzed further is referred to as the **root**.
- A **stem** is an inflected form of the base. For example, 'acts' is a stem derived from the base 'act’.

In the context of Ge’ez, which exhibits complex morphology characterized by root-pattern morphology in addition to concatenative morphology, we use the terms root, base, and stem with the following definitions:

Root pertains to the fundamental lexeme of a Ge'ez word and is typically represented by its consonants or by removing the vowels from the word. For instance, the root 'ቅትል' (qtl) consists of the consonants from the dictionary word 'ቀተለ' - *qätälä*.

Stem refers to the morpheme formed through the intercalation of vowels into the root. The stem may or may not represent a complete word form, and additional concatenation operations may be necessary to produce the surface form of the word.

Base designates the word obtained by applying the required concatenation operation for the third person singular male form for a particular verb type. In Ge'ez dictionaries, words are often listed in their perfective form (simple past) for the third person singular male (3PSGM).

For example, the root 'ብርክ' -*brk*, when combined with vowels 'a' and 'ä,' yields the stem 'ባረክ' - *baräk*. When the stem 'ባረክ' - *baräk* is concatenated with the suffix 'ä,' it forms the base 'ባረከ'

- *baräkä*, which represents the 3PSGM perfective form of the verb 'ብርክ' - *brk*. Conversely, when the root 'ብርክ' - *brk* is combined with the vowel 'a,' the resulting stem 'ባርክ' - *bark* can be further concatenated with the prefix '*y*' to create the base 'ይባርክ'- *yəbark*, representing the 3PSGM indicative form of the root 'ብርክ' - *brk*.

5.2.3 Word Formation

Word formation form morphemes formation are commonly classified into four broad categories: inflectional, derivational, compounding, and cliticization (Trost, 2003).

- **Inflection**: This involves the combination of morphemes to create another word form of the same part of speech (POS). For example, the word 'eat' (verb) can be combined with the morpheme 's' to produce the inflectional word 'eats' (verb).
- **Derivation**: This refers to the combination of morphemes to create words of different POS. For instance, the verb 'sing' can be concatenated with the morpheme 'er' to form the derivational word 'singer' (noun).
- **Compounding**: Compounding involves the combination of words to create a new word. For example, 'bedroom' is a word created by combining 'bed' and 'room'.
- **Cliticization**: This category involves the combination of a word with a clitic. An example is the combination of the morphemes 'I' and ''m,' resulting in 'I'm'.

In Ge'ez word formation from morphemes, you can find inflectional, derivational, and compounding processes, but it does not include cliticization. For instance, consider the verb '- qäddäsä,' which means 'consecrated' in the third person singular male (3PSGM) form. This verb can take various word forms:

- ይቄድስ - *yəqeds* - an indicative form of the verb (inflectional)
- መቅደስ - *mäqddäs* - a noun (derivational)
- ቤተመቅደስ - *betämäqddas* - a word formed by a combination of the words ቤተ - *betä* and መቅደስ - *mäqddäs* (compounding)

5.2.4 Morphotactics

Morphotactics involves the rules governing how morphemes combine to form words, dictating how morphemes must be arranged to create valid words. For example, in English, "care-less-ness" is a correct word, while "care-ness-less" is not. In Ge'ez, there are morphotactic rules, such as the negation morpheme "ኢ - ʿi," which must always be a prefix to the word's surface form. For instance, "ሰትየ - sätyä" (he drank) combined with the negation marker "ኢ - ʿi" produces "ኢሰትየ - ʿisätyä" (he did not drink).

In addition to morphotactics, orthographic rules play a crucial role in specifying spelling changes that occur when morphemes are combined. For instance, when the word 'fly' is combined with the plural morpheme 's,' it results in 'flies,' not 'flys'.

Ge'ez exhibits both orthographic and phonological alternations during the combination of morphemes to form inflectional and derivational forms of words.

Languages generally fall into two categories of morphology: concatenative and non-concatenative. Concatenative morphology involves the formation of words by concatenating morphemes, often through prefixation and suffixation. Morphological operations in concatenative morphology typically involve adding morphemes to the left or right end of the root (McCarthy, 1981). On the other hand, non-concatenative morphology is more complex and involves morphological alternations within the root. Semitic languages like Ge'ez, Arabic, and Hebrew are known for their non-concatenative morphology, often referred to as root-pattern morphology. Ge'ez, as a Semitic language, prominently features non-concatenative morphology in its word formation processes.

Ge'ez, being a Semitic language similar to Arabic, can benefit from the prosodic theory proposed by McCarthy (1981) for non-concatenative morphology. This theory provides insights into the complex internal changes within roots that occur during word formation in Ge'ez. However, it is important to note that Ge'ez word formation also encompasses concatenative morphology.

5.2.5 Root and Pattern Morphology

Semitic languages are renowned for their non-concatenative morphology, characterized by morphological alternations occurring primarily within the stem. McCarthy (1981) introduced the prosodic theory for non-concatenative morphology, illustrating how Arabic language morphology can be characterized by a root-pattern morphology. This morphology utilizes a prosodic template, often referred to as the C V - template, where 'C' represents consonant roots (lexemes) and 'V' represents vowels. This template, along with defined vocalic patterns, governs the intercalation of vowels into consonant roots and may include affixes for inflection and derivation.

For example, let us consider the root ቅትል - *'qtl*.' It can have the following vocalic pattern and vowels to form a verbal stem based on the prosodic theory:

Root consonants	q-t-l
Vocalic Pattern	C V C V C
Vowels	ä ä
Stem	q ä t ä l

As an illustration, ቅትል - *'qtl'* represents the lexeme for the verb ቀተለ - *'qätälä*,' where 'C V C V C' and 'ä-ä' respectively represent the vocalic pattern and vowels. Through the intercalation of vowels into the root consonant, a verbal stem ቀተል - *'qätäl'* is produced, representing the perfective form of the root ቅትል - *'qtl*.' Depending on the type of verb, the CV template and/or the vowels may vary in the formation of verbal stems.

Let's consider the root ብርክ - *brk* as an example. This root can undergo a process of intercalation with a vocalic pattern represented as C V C V C, along with the vowels a - ä, resulting in the creation of a verbal stem, ባረክ - *baräk***.** Now, this verbal stem, ባረክ - *baräk*, is ready for further modification. In this case, it concatenates with the vowel 'u' to produce the verb ባረኩ - *baräku*, which translates to 'they praised' in the third person plural male form (3PPM). This process illustrates how, after forming a stem from the root through the application of root-pattern morphology, subsequent concatenation operations, including prefixation and suffixation, can be applied to the stem to generate the surface form of a word. It is essential to note that in this research, the CV-pattern morphology is exclusively applied to Ge'ez roots to create verbal stems, while additional affixes are introduced to the stem using concatenative morphology.

5.3 Morphological analysis

Computational morphology involves the processing of words and word forms (Jurafsky & Martin, 2008). Morphological analysis, a crucial component of computational morphology, is a process that produces structural information and the lexeme associated with a given surface form of a word. Conversely, morphological generation is the computational process that generates or outputs the surface form of a word from a given lexeme and its corresponding structural information. For example, when analyzing the Ge'ez verb ቀተልኩ - *qätälku* (meaning 'I killed'), the morphological analysis provides the following information:

- *qtl* - the root
- VPER indicating the perfective form of the root *qtl*

- 1PSG indicating the subject is the first person singular

Identifying the lexeme and structural information of a given surface form of a word is a fundamental task in NLP. Morphological analysis serves as a cornerstone for various NLP applications, including parsing, generation, machine translation, lemmatization, online dictionary construction, speech synthesis and recognition, document retrieval, and word processing (Sproat, as cited in Tachbelie, 2010, p. 52). Particularly for lesser-studied languages like Ge'ez, morphological analysis is a valuable initial step in NLP task. In this research, the primary goal is to develop a morphological analyzer specifically tailored for Ge'ez verbs. Notably, our Ge'ez verbs morphological analyzer has the capability to both analyze and generate Ge'ez verbs, making it a versatile tool for various NLP tasks related to the Ge'ez language.

5.3.1 Finite-state Technology

Finite-state automata or a finite-state machine is a system that has a start state and one or more final states. The transition between states is triggered by an input and the transition between states is allowed only if the input is recognized by the system. An FST is a type of finite-state automata with pairs rather than a single symbol which makes it able to map one pair to another. It follows then that an FST can implement the relation between the lexical and surface form of the word in morphological analysis.

The basic notion of the two-level-morphology is that there is a regular relation between the lexical and surface form of a word. The finite-state approach to computation morphology is based on representing a relation between the surface form and its lexical form together with the syntactical information about the word. This relation can be described using the metalanguage of regular expressions. A language's morphology can be described using regular expressions which may be compiled into finite-state transducers using a finite-state tool such as Xerox XFST and Foma. Finite-state transducers are finite-state machines where each transition is labeled in pairs indicating the lexical and surface forms.

Finite-state techniques are widely used for NLP because finite-state techniques are simple in representing morphological rules, fast and compact in size when storing and using morphological rules, and its bidirectional feature works for both analysis and generation.

However, there is a challenge in using finite-state techniques in Semitic languages (Beesley, 1996; Koskenniemi, 1983). This is because Semitic languages exhibit a non-concatenative or the root-pattern morphology in addition to concatenative morphology. As a solution to Semitic root-pattern morphology, Kay (1987) proposed FSTs that will work in parallel using four tapes. Whereas Kataja and Koskenniemi (1988) proposed using an intersection of two separate lexicons, one consisting of root words and the other of inflectional elements, that would represent the lexical form of the word. Using the ancient Akkadian language as an example, they demonstrated that the intersection of the two lexicons can represent the Semitic root-pattern morphology or interdigitation of Semitic roots. However, Beesley and Karttunen (2000) argue that the limitation of finite-state techniques in Semitic language is due to the reliance on concatenative operations for morphotactic description. Hence, they proposed a technique called compile-replace that involves reapplying the regular expression compiler to its output as a solution for the non-concatenative process. Moreover, the compile-replace algorithm together with the merge operator, a pattern-filling operator which combines the root and the pattern into a single one, has been applied to Arabic stem interdigitation (Beesley & Karttunen, 2000) and simple Amharic verbs (Amsalu & Demeke, 2006). In addition, Yona and Wintner (2008) have used the extended regular expression languages of XFST for designing the Hebrew morphological analyzer. On the other hand, Cohen-Sygal and Wintner (2006) proposed finite-state registered automata for non-concatenative morphology that extends finite-state automata with finite memory (registers) that enables it to remember a finite number of symbols and hence reduce duplicate paths. They argue that the finite-state registered automata is efficient for non-concatenative phenomena such as root-pattern word formation or circumfixation with a significant decrease in the number of states and the number of transitions in the finite-state registered network.

As a solution to the root-pattern morphology of Ge’ez verbs, the finite-state flag diacritics feature is applied in this study. Flag diacritics are features setting operations in finite-state compilers that allow one to set constraints between disjoint parts of words (flag diacritic features are discussed in section 5.4).

5.3.2 Morphological Analyzer

A morphological analyzer is a computational artifact that inputs a surface form of the word and outputs its morphemes together with the structural information about the word, usually in the form of morphological tags. Two-level morphology and finite-state morphology are widely used in morphological computational analysis (Beesley & Karttunen, 2000; Kay, 1987). Finite-state morphology relies on finite-state transducers (FSTs), which are types of finite-state automata featuring two tapes—one for input and one for corresponding output. To illustrate, Figure 5.1 provides an example of an FST mapping the lexical form of "swim +V +3PSG" to the surface form "swims”:

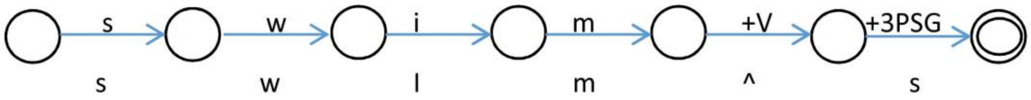


Figure 5.1: FST for the word ’swims’

In the design of a morphological analyzer using the finite-state method, three essential components play a pivotal role, as outlined by Trost (2003:

- Lexicon: A list comprising stems/roots and affixes, along with basic grammatical information (e.g., noun, verb, prefix).
- Morphotactics: A model that defines how the morphemes from the lexicon combine or fit together.
- Morphophonological alternation rules: Rules accounting for spelling changes that occur when morphemes combine to create a word.

The finite-state technology offers a powerful framework for describing the morphology of natural languages using regular expressions. These regular expressions can be converted into finite-state automata and Finite-State Transducers (FSTs) by employing finite-state tools. The adoption of finite-state technology provides several advantages, including remarkable speed, compact storage requirements for morphological rules, and support for both morphological analysis and generation.

In this study, we applied finite-state technology as the foundation for designing and implementing the Ge'ez verbs morphological analyzer, relying on the crucial components mentioned earlier. To facilitate this development, we utilized the freely available Foma finite-state tool, which proved to be a valuable asset in the creation of the Ge'ez verbs morphological analyzer.

5.3 Regular Expression

A regular expression is a standard algebraic notation for characterizing a set of strings (Trost, 2003). Moreover, regular expression denotes a set of strings or string pairs. In the same vein, regular expressions can be used to specify search strings (Trost, 2003). For instance, a regular expression may denote a single character a or may be an expression [a+] which represents an infinite set of one or more characters of a - {’a’, ’aa’, ’aaa’, ’aaaa’} etc.

A regular expression can be implemented as a finite-state automaton. A finite-state automata or finite-state machine is a system that has a start state and one or more final states. The transition between states is triggered by an input and the transition between states is allowed only if the input is recognized by the system.

For example, the regular expression of the sheep talk (Jurafsky & Martin, 2008) {baa!, baaaa!, baaaaa!} is the expression /baa+!/. This sheep talk can be modeled using a finite-state automaton by recognizing the set of strings representing the sound. The finite-state automaton can be represented as a graph which consists of starting state, an arc representing the transition for acceptable characters, one or more states and a final state. So, the finite-state machine for the sheep talk consists of q0 representing the start state, q1 to q3 representing one or more states and q4 represents the final state. Figure 5.2 shows the FSA for sheep talk.

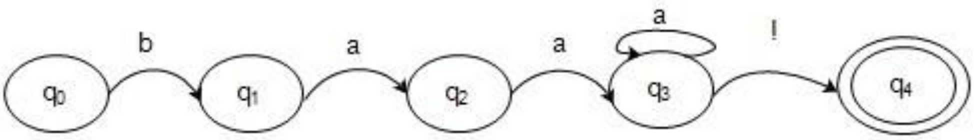


Figure 5.2: FSA for the sheep talk

While a finite-state automaton is an automation with a single symbol that accepts or rejects input, an FST is a finite-state automaton with pairs of symbols with output that defines a relation.

5.3.1 Finite state transducer

Finite-State Transducers (FSTs) are a specialized type of finite-state automaton that work with pairs of symbols, allowing them to map one pair to another. This characteristic makes FSTs particularly well-suited for capturing the relationship between the lexical form and the surface form of words in morphological analysis. For instance, an FST can be designed to map the lexical and surface forms of a word like 'ቀተለ-qätälä,' as illustrated in Figure 5.3. Table 5.1 provides a description of the lexical and surface forms of 'qätälä.

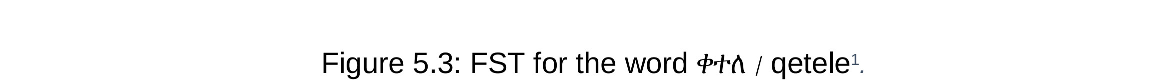


Figure 5.3: FST for the word ቀተለ / qetele[1].

Table 5.1: Lexical and surface form of -qetele

Surface Level	Lexical Level	Description

[1] The SERA transliteration is used for the implementation of the Ge’ez Morphological analyzer

- *qätälä*	[Verb1]	Verb Type 1
	[qtl]	root
	[VPER]	Perfective
	[3PSGMs	3rd person singular male subject

Creating a finite-state transducer for a natural language involves first describing the language using regular expressions and then compiling these expressions into FSTs using finite-state tools. Various computer programs are available for this purpose, including the Xerox finite-state tool (Beesley & Karttunen, 2003), AT&T's FSM library, and Foma (Hulden, 2009).

In this study, the utilization of the freely available Foma finite-state tool played a crucial role by facilitating the construction of the Ge'ez verbs morphological analyzer. Foma's capability to generate automata and transducers from regular expressions proved to be instrumental in managing the intricate morphology of the Ge'ez verbs.

5.4 Introduction to Foma

Foma is a finite-state compiler, a programming language and C library for constructing finite-state automata and transducers for various uses and has specific support for NLP (Hulden, 2009). Finite-state automata define the same language as regular expressions do and accept strings that are part of the domain while rejecting strings that are not part of the domain / language.

For example, for a regular expression /ab*/ - the language is a set of strings consisting of {ab, abb, abbb, abbbb} etc. A finite-state machine would accept strings in the language and reject others. In the same vein, an FST accepts input in the same way as a finite-state automaton with the additional feature of mapping the acceptable input to output. A finite-state transducer (FST) transforms an input to output. It follows then an FST corresponds to regular relation. For a simple regular relation [a:b], an FST maps a to b. We call ’a’ the upper side and ’b’ the lower side.

A finite-state transducer represents a relationship that is expressed by the regular expression operators of concatenation, Kleene star, union and cross product (Hulden, 2009). Complex regular expressions can be built from small regular expressions.

5.4.1 Foma Regular Expression Operators

The Foma compiler is an FST for converting regular expressions to finite automata and transducers. Some of regular expression operators supported by Foma are briefly described below (Hulden, 2011a):

- **Concatenation X Y**: The language or relation X concatenated with Y
- **Union X|Y:** The union of languages or relations X and Y
- **Intersection X & Y:** The intersection of languages X and Y
- **Optionality (X):** Defines the language or relation that contains zero or one iteration of X
- **Kleene Star X*:** Zero or more iterations of X, for example [X*] represent a language or a relation with empty set or zero or more iteration of X. so X* includes [], [X], [XX] etc.
- **Kleene Plus X+:** One or more iterations of X, for example [X+] represents a language or a relation with one or more as where X+ includes [X], [XX], [XXX], etc.
- **Substitution '[X,Y,Z]:** The language X where symbols Y are substituted for Z.
- **Term negation \X:** Any single symbol, except X
- **Cross product X:Y:** Produces a transducer that represents the relation that maps any string from X to any String in Y
- **Composition X .o. Y:** Produces the intersection of the X and Y. for example the expression [a:b].o. [b:d] results in [a:d]
- **Replacement Operator X - >Y:** The replacement of X with Y. for example a - >b replaces instances of a with b, for an input aab the output would be bbb. Replacement operators may be context specific.
- **Conditional Replace Operator X - >Y   L R:** The replacement of X with Y with restriction, X - >Y L _R indicate the replacement of X with Y when occurring between L and R.
- **Epsilon Modifier [..]:** The left-hand side (LHS) may contain epsilon modifier together with the replace operator which produces an insertion when the context is matched. For example, the rule [..] ->x will map an input aaa with xaxaxax while the rule [. a.] ->x with an input a outputs xxx
- **Word-boundary marker (.#.):** The word boundary may be used in context specification of both the context restriction operator as well as replacement rules. For example, the rule x ->y d _. #. Specifies the language where x is replaced by y when it is found between 'd' and a word boundary.
- **Define VARIABLE regular expression:** The define command can be used to define regular expression function **define function (prototype) regular expression**. Moreover, the define command can be used to define a regular language.

For instance, we can define a regular language of the words chair and table and give that language name furniture:

define *furniture chair* | *table*;

The **define** command can also be used to label a finite-state network represented by regular expression and reuse it in later expression.

For example, a network that specifies a language containing A can be defined as:

define ContainsA [?* A?*];

- **Read lexc:** The read command reads lexc files - the lexicon
- **Regex regular-expression:** The regex command compiles a regular expression into automata or a transducer.
- **Flag diacritics:** The flag diacritics provide feature setting and feature unification operation to enforce constraints (Beesley & Karttunen, 2003). “Flag Diacritics are often used to enforce separated or ”long-distance” constraints on the co-occurrence of morphemes within words, constraints which are awkward to handle in regular expressions” (Beesley & Karttunen, 2003, p. 442).

A flag dialect format is:

@ FLAGTYPE.FEATURE.VALUE@

When using the flag diacritics, the flag FEATURE and VALUE are arbitrary for users to decide. However, the FLAGTYPE diacritics have specific features. At runtime, Foma would decide whether or not a word is accepted depending on which flags co-occur in the same word. The flag types are:

- U unify features @U.FEATURE.VALUE@
- P positive set @P.FEATURE.VALUE@
- N negate @N.FEATURE.VALUE@
- R require feature or value @R.FEATURE.VALUE@ or @R.FEATURE@
- D disallow feature or value @D.FEATURE.VALUE@ or @D.FEATURE@
- C clear feature @C.FEATURE@
- E require equal feature or values @E.FEATURE.VALUE@

When using the flag diacritics, the following guidelines must be taken into consideration:

- All the flags must be declared as multi-character symbol in the lexc file
- The flags must be aligned on both the upper and lower side in the lexc
- The rules must take into account the flag symbols either by making the replace rules flag aware or by setting the command flag-is-epsilon

5.4.2 Designing Morphological Analyzer Using Foma

Foma uses the special character E epsilon or 0 or [] to denote an empty string. To mark a morpheme boundary, the caret symbol ˆ is used. It is also possible to specify and use multi-character symbols such as +VERB or [NOUN].

In order to design a morphological analyzer using Foma, two main components are needed:

- the lexical component -**lexicon**
- the alternation rule component - **rule**

5.4.2.1 Lexicon

The lexicon is a transducer that consists of the languages roots and the appropriate feature tags expressing the morphotactics. For instance, the lexicon file may contain the following mapping:

Cat +noun +plural - Cat ˆs

The lexical transducer is written in the formalism called lexc (Foma, 2011).

5.4.2.2 Alternation rule

The role of the rule component is to perform the necessary modification on the output of the lexical transducer based on the morphophonological rule of the language. For instance, the lexicon may contain a mapping indicating the concatenation of s to nouns to produce the plural form of a noun. However, some words such as watch + s result in watches rather than watchs. The alternation rule transducer does this alternation. Thus, the rule component is an intermediate between the lexical transducer and the output. The combination of lexicon transducers and rule transducers is achieved through the composition.o. operator.

Lexicon.o. Rule ->FST (morphological analyzer)

The compositions of the transducers result in a single transducer, which is the morphological analyzer of the language.

5.5 Application of Foma for Ge’ez Morphological analyzer

In the implementation of the Ge'ez morphological analyzer, Foma version 0.9.18 alpha was used effectively. It provided valuable functionalities, such as flag diacritics, which were instrumental in handling the insertion of vowels into root consonants, a fundamental aspect of Ge'ez's non-concatenative morphology. The analyzer was organized with separate lexc files for each of the Ge'ez head verbs and irregular verbs to implement the inflectional and derivational features of each verb type. Additionally, alternation rules were formulated within the Foma framework to address morphophonological alternations. The flag diacritics and composition operators played a pivotal role in managing the complexities of morphological analysis and generation in languages like Ge'ez that exhibit non-concatenative features.

5.6 Chapter Summary

This chapter provided a comprehensive overview of morphology, including both concatenative and non-concatenative aspects, and highlighted the significance of finite-state technology in morphological analysis. It addressed the specific challenges posed by languages like Ge'ez, which exhibit non-concatenative features, and explained how finite-state technology can be applied to represent the non-concatenative morphology of Ge'ez. Additionally, it introduced Foma, a finite-state compiler, which played a crucial role in the development of the Ge'ez verb morphological analyzer. This chapter serves as a foundational understanding of the concepts and tools that will be applied in the design and implementation of the Ge'ez verb morphological analyzer.

The next chapter discusses the design and implementation of the finite-state-based Ge’ez verb morphological analyzer based on the Ge’ez verb formation.

Chapter 6 – Design & Implementation of Ge’ez Morphological Analyzer

6.1 Introduction

This chapter outlines the design and implementation of the Ge’ez morphological analyzer using the Foma finite-state tool.

The main objective of the Ge’ez morphological analyzer is to take a word (typically a verb) as input and provide the corresponding lexeme (root), along with its syntactical information, as output. Additionally, this morphological analyzer has the capability to generate the surface form of a word based on a given lexeme and its associated syntactical information. Essentially, the finite-state-based Ge’ez morphological analyzer serves the dual purpose of morphological analysis and the generation of Ge’ez verbs.

The development of this finite-state-based Ge’ez morphological analyzer is grounded in the understanding of Ge’ez language morphological properties, as discussed in Chapter 2. In that chapter, we explored the fundamental characteristics of Ge'ez morphology.

This chapter addresses the research sub-questions, seeking to determine how a Ge’ez morphological analyzer can be developed using finite-state methods.

The subsequent sections will delve into the specifics of the design and implementation of the Ge’ez verb morphological analyzer.

6.2 Ge’ez Morphological Analyzer Design

Morphemes represent the simplest forms of a word, and words are constructed by combining these morphemes. A morphological analyzer dissects words into their constituent morphemes, providing the lexeme along with syntactical information pertaining to the word. Simultaneously, when given a morpheme along with its associated syntactical information, a morphological generator generates the surface form of a word.

For example, when given the word 'books,' a morphological analyzer can output two possibilities:

Books = book + s or Books = book +noun +plural

Likewise, when provided with the morpheme 'bench' and its syntactical information, followed by 'plural,' a morphological generator would produce Bench +plural = **benches**.

To illustrate the functionality of the Ge'ez morphological analyzer, let's consider the word 'ነበበሙ' -*näbäbomu*, which can be analyzed as follows:

ነበበሙ - ***näbäbomu*** = +Verb1 nbb +VPER+3PSGMs+3PPLMo

The analysis of the word ነበበሙ' –*näbäbomu*, which translates to 'he told them,' reveals the following information about the verb:

- It is a perfective form of a verb
- It indicates a third person male singular subject
- It indicates a third person male plural object
- It belongs to verb type 1
- The root of the verb is ን-ብ-ብ-nbb

The primary goal of the Ge’ez morphological analyzer is to produce the analysis of a given verb. Although the base form of Ge’ez verbs is typically the third person singular male past tense form, this study considers the constant root as the root/lexeme. Hence, the Ge’ez morphological analyzer outputs the root of the verb (as depicted in Figure 6.1).

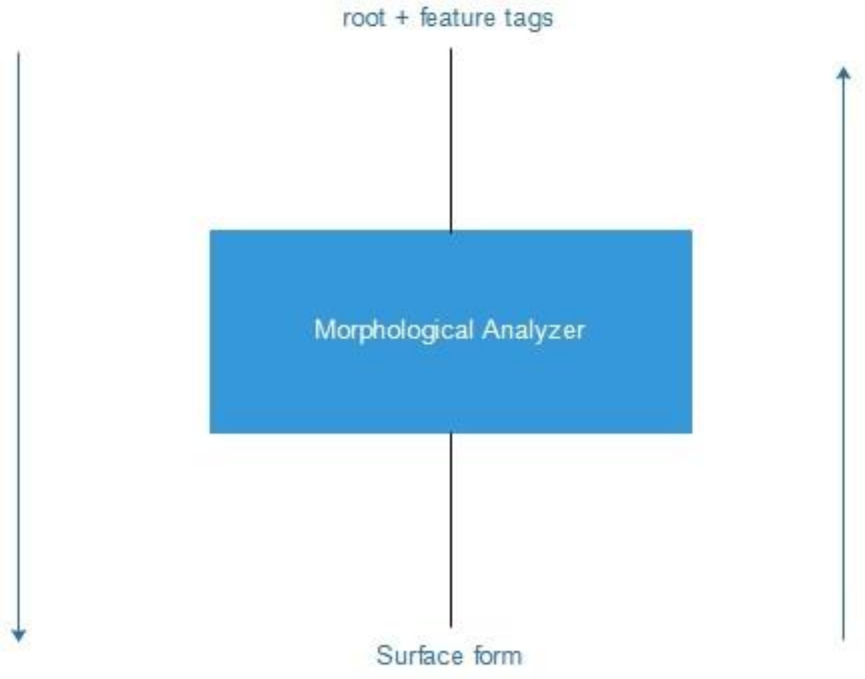


Figure 6.1: Morphological analyzer

As detailed in Chapter 2, we have outlined the process of Ge’ez verb formation, which initiates with the root and undergoes several stages, including intercalation, morphophonological alternations, and suffixation, culminating in the creation of the verb's surface form. The Ge’ez root serves as the foundation from which various surface forms are derived. Figure 6.2 illustrates the model for Ge’ez verb analysis and generation, based on the properties of Ge’ez verb formation.

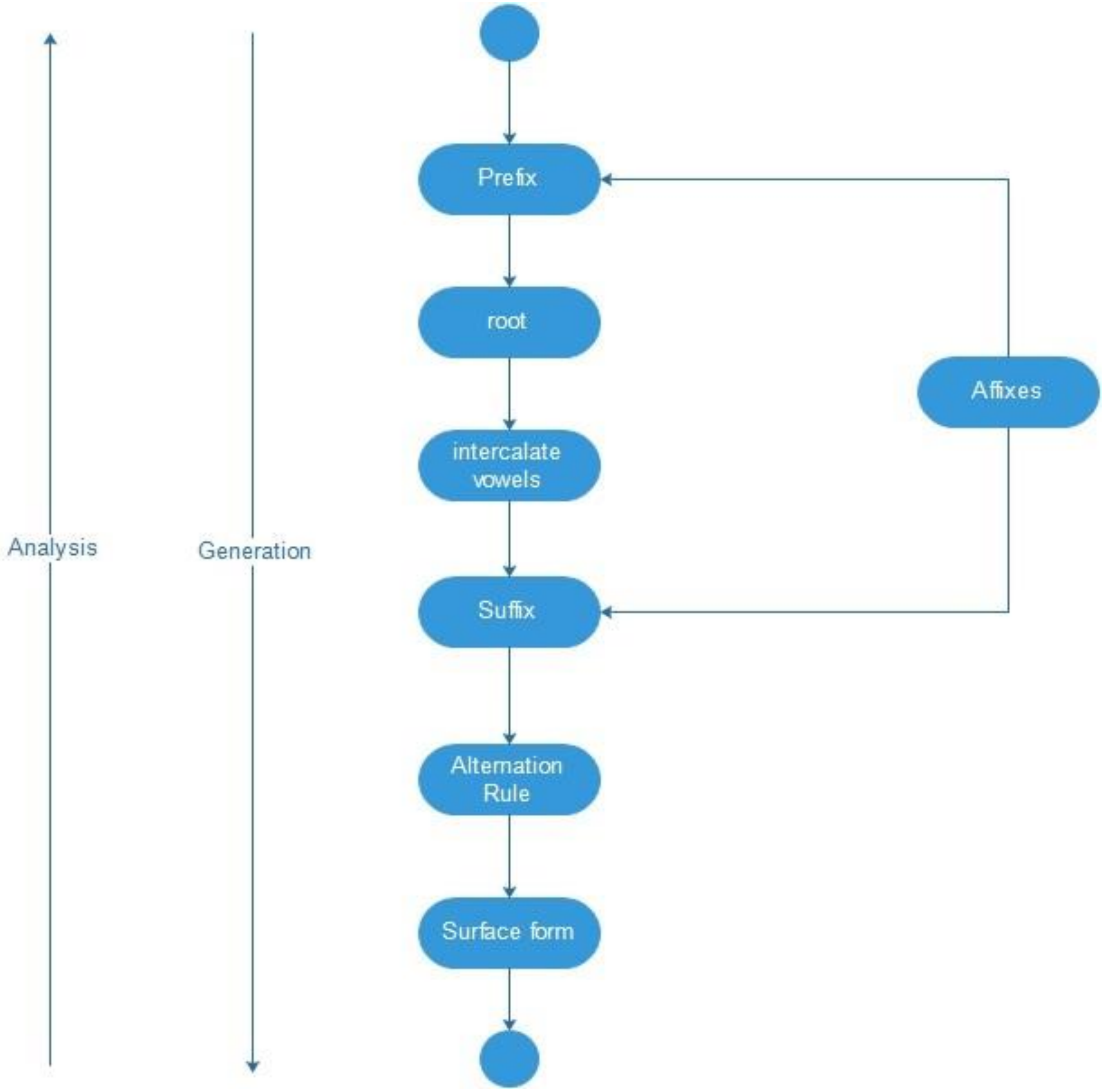


Figure 6.2: Model for Ge’ez morphological analyzer

Figure 6.2 visually represents the sequential steps involved in the formation of Ge’ez verbs. This process begins with the concatenation of prefixes with the root, followed by the insertion of a vocalic pattern into the root consonants. Subsequently, the appropriate suffix is concatenated with the morpheme, and alternation rules are applied to generate the surface form of the verb. Notably, this model is implemented using the Foma FST.

Foma serves as a fundamental tool in this research, acting as a finite-state compiler, a programming language, and a C library. Its primary function is the construction of finite-state automata and transducers, with specific applications in NLP (Hulden, 2009). To effectively use Foma within the context of this research, it is essential to be acquainted with commonly used Foma commands, notations, and rules, as outlined by Hulden (2011b). Table 6.1 provides a concise reference to these key elements.

Table 6.1: Foma commands, notations and rules

Commands	
source[filename]	compile script
down	test words in downward direction (CTRL-D exists)
up	test words in upward direction (CTRL-D exists)
define name[regular expression];	label a regular expression with name
regex[regular expression or labeled FST]	compile regular expression
words	print all the word in an FSM
lower-words	print all the possible output-side words
upper-words	print all possible input-side words
view	show all the FSM graphically
print defined	show all the labeled FSMs
quit	quit
Rule format	
NOTATION	
[ ]	grouping
?	any symbol
?*	any sequence
a	a single symbol
/ a	any symbol except a
/ C	any symbol except a constant, C presumably defined with ”define”
.#.	word edge in rule contexts
[a|b]	a or b
[C |.#.]	a constant or word edge
a	any numberof a symbol
(a)	optionally a
.o.	compose
RULE NOTATION	
a b || c _ d	rewrite a as b in the context c_ d

a ( ) || c _ d	optionally rewrite a as b in the context c _ d
[..] x || c _ d	epenthesize x in context c _ d
x 0 || c _ d	delete x in context c _ d
a b, c d || e _ f	multiple left-hand sides

Source: (Hulden, 2011b)

6.3 Implementation of the Ge’ez Morphological Analyzer

The implementation of a morphological analyzer using Foma comprises two primary components: the lexical component and the rule component. The lexical component comprises a lexicon containing roots and morphotactic features. In contrast, the rule component encompasses alternation rules that are applied to the output of the lexical Finite-State Transducer (FST).

In this chapter, as previously indicated in Chapter 2, the SERA transliteration is employed for the implementation of the Ge’ez verb morphological analyzer. The transliteration used in the implementation is shown in Table 2.2.

6.3.1 The Lexical FST

The lexicon functions as a transducer, exclusively accepting valid words or lexemes of the language along with valid feature tags, and generating an output (referred to as intermediate output) corresponding to the provided input. Additionally, this output may include supplementary symbols, such as the morpheme boundary symbol, to enhance the analysis (Foma, 2011).

A lexical script is a text file written in the formalism called lexc. The lexc formalism operates by declaring labeled lexicons, listing the contents of those lexicons and specifying the rules that govern how the lexical entries are concatenated to produce the output (Foma, 2011).

In essence, the lexical file combines the declaration of multi-character symbols, the inventory of morphemes (including roots), and the concatenation rules. Together, these elements form the foundation for generating the intermediate output in the morphological analysis process.

6.3.2 Ge’ez Lexicon

To implement the Ge’ez lexicon, it is essential to understand Ge’ez verb formation within the lexc formalism. Ge’ez verb formation begins with the addition of prefixes, when necessary, to the consonant roots. These prefixes include negation, subject, and derived-verb prefixes, applied in sequence. A Ge’ez verb can either have none of these prefixes or may contain one, two, or all of them, following the order of negation, subject, and derived-verb prefixes. For example, the perfective verb type ቀተለ - qätälä (he killed) does not have a prefix.

After adding prefixes to the root, the next step involves the concatenation of suffixes. Ge’ez language suffixes encompass subject and object suffixes. Similar to prefixes, a Ge’ez verb may or may not require suffixes.

As a result, the Ge’ez lexical transducer generates an intermediate output comprising Ge’ez consonant roots with affixes and their corresponding feature tags. The output of the Ge’ez lexical transducer can take on various forms, including:

- Prefix1 ˆPrefix2 ˆCCC
- Prefix1 ˆCCC
- Prefix2 ˆCCC ˆSuffix1 ˆSuffix2
- CCC + VPER + 3PSM

Figure 6.3 illustrates a flowchart of Ge’ez verb formation on which the lexicon is based.

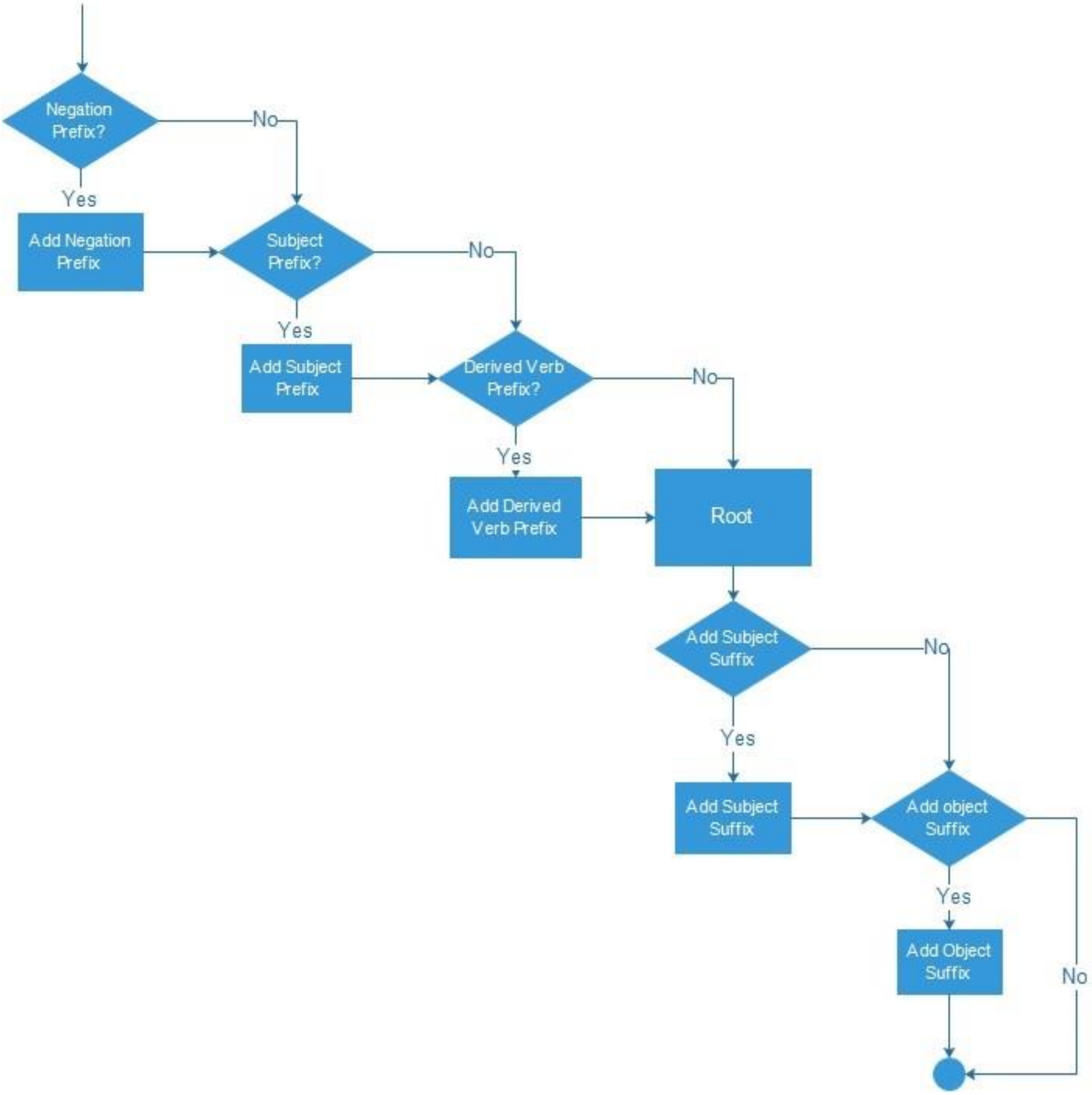


Figure 6.3: Flow chart of Ge’ez verb formation

The Ge’ez lexicon script file consists of the following:

- declaration of the multi-character symbol representing feature tags to mark grammatical information, flag diacritics that are used to add the prefixes and the rules that are used to perform vowels intercalation and morphophonological alternation
- list of prefixes together with rules for adding the prefixes to the roots
- list of roots
- list of suffixes together with rules for concatenating the suffixes to the roots

In the subsequent sections of this document, we will provide a detailed explanation of the Ge’ez lexical script.

6.3.3 Multi-Character Symbols

The Foma command for declaring multi-character symbol is ‘Multichar Symbols’. The command Multichar Symbols’ is used to declare the feature tags and the flag diacritics. For example, the following command lists the multi-character symbols used in the Lexicon.

Multichar Symbols +VIND +1PSG @U.VERBTYPE.CAUSITIVE@ @P.eInsertion.e@

Table 6.2: Multi-character symbol

Multi-Character Symbol	Description
+VIND	Indicative verb feature tag
+1PSG	1st person singular feature tag
@U.VERBTYPE.CAUSITIVE@	Flag dialect for adding causative prefix
@P.eInsertion.e@	A rue component for the insertion of vowels

A complete list of the multi-character symbols used in the Ge’ez lexical file is found in the lexical file of each verb type.

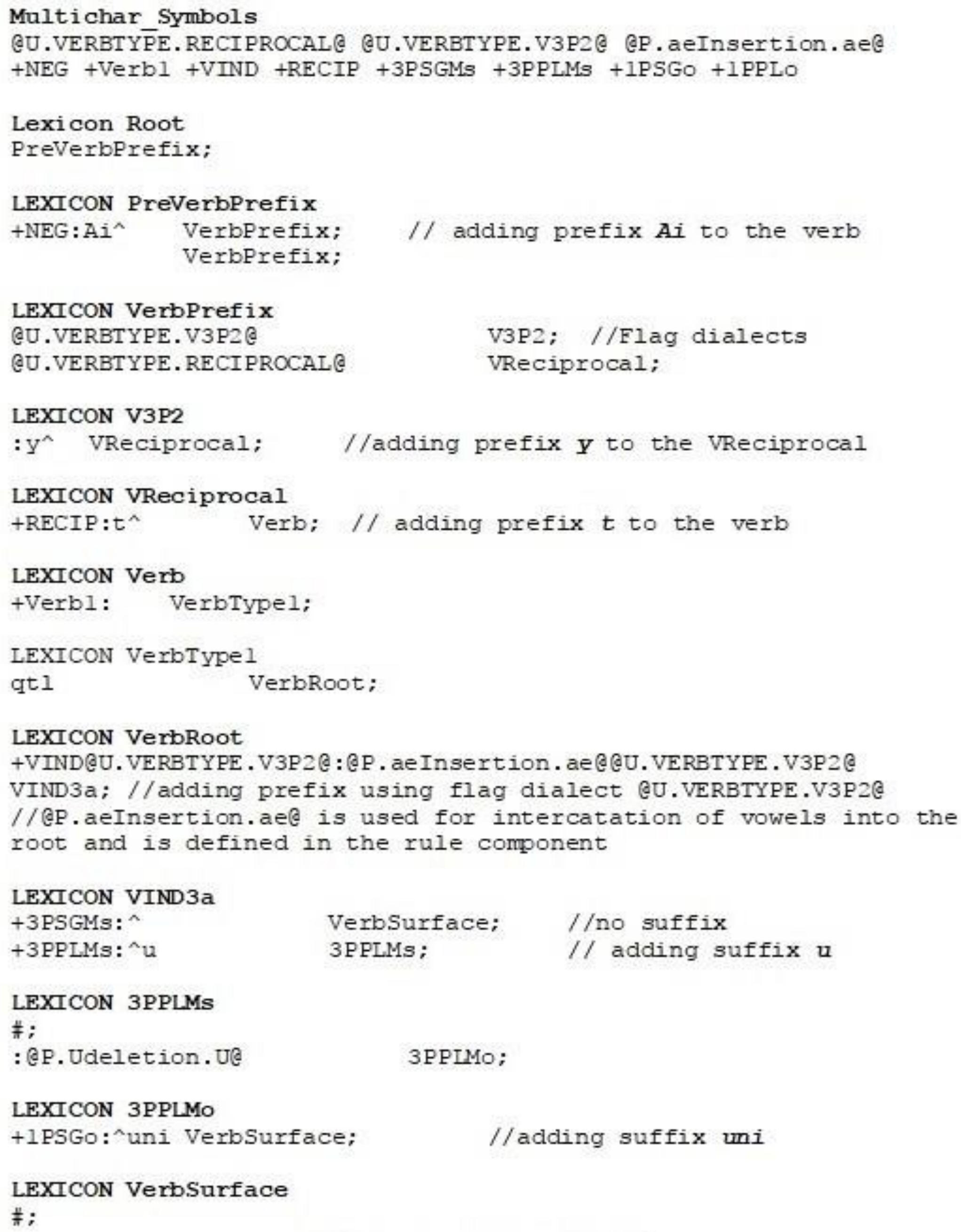
6.3.4 Affixes and Roots

Ge’ez verb formation primarily involves two fundamental processes: intercalation and affixation. Ge’ez verb affixation encompasses negation, subject, object, and derived-verb markers. Negation and derived-verb markers function as prefixes, while the object marker is a suffix concatenated to the consonant root. The subject marker, however, exhibits variability and may serve as both prefix and suffix depending on the verb tense-mood.

In the implementation of affixes within the lexical files, each affixation is separated by a morpheme boundary marked as 'ˆ'. The concatenation of prefixes to the root employs a specific type of flag diacritics known as U-type (unification flag dialect). Notably, all prefixes except the negation prefix utilize these flag diacritics. This distinction arises from the fact that the negation morpheme can be added to all verb forms, whereas subject and derived-verb prefixes may or may not be added to all verb forms.

To illustrate, consider the Ge’ez verb ይትቃተል (*yətqatäl*), which features two prefixes added to the root ቅትል (*qtl*). These prefixes, namely ይ (*y*) and ት (*t*), represent the subject prefix for third person singular male and the reciprocal derived-verb prefix, respectively. This combination is achieved through the utilization of unification flag diacritics.

The Ge’ez verb roots are listed in the lexicon, indicating the valid list of roots for each verb type. Following the listing of the roots, suffixes are concatenated to the root. The output of the Ge’ez lexical transducer results in an intermediate verb form. Figure 6.4 provides an example of a lexical file illustrating how affixation is applied to the root consonants of Ge’ez verbs



```
Multichar_Symbols
@U.VERBTYPE.RECIPROCAL@ @U.VERBTYPE.V3P2@ @P.aeInsertion.ae@
+NEG +Verb1 +VIND +RECIP +3PSGMs +3PPLMs +1PSGo +1PPLo

Lexicon Root
PreVerbPrefix;

LEXICON PreVerbPrefix
+NEG:Ai^    VerbPrefix;    // adding prefix Ai to the verb
            VerbPrefix;

LEXICON VerbPrefix
@U.VERBTYPE.V3P2@               V3P2;  //Flag dialects
@U.VERBTYPE.RECIPROCAL@         VReciprocal;

LEXICON V3P2
:y^  VReciprocal;     //adding prefix y to the VReciprocal

LEXICON VReciprocal
+RECIP:t^        Verb;  // adding prefix t to the verb

LEXICON Verb
+Verb1:    VerbType1;

LEXICON VerbType1
qtl              VerbRoot;

LEXICON VerbRoot
+VIND@U.VERBTYPE.V3P2@:@P.aeInsertion.ae@@U.VERBTYPE.V3P2@
VIND3a; //adding prefix using flag dialect @U.VERBTYPE.V3P2@
//@P.aeInsertion.ae@ is used for intercatation of vowels into the
root and is defined in the rule component

LEXICON VIND3a
+3PSGMs:^             VerbSurface;    //no suffix
+3PPLMs:^u            3PPLMs;         // adding suffix u

LEXICON 3PPLMs
#;
:@P.Udeletion.U@           3PPLMo;

LEXICON 3PPLMo
+1PSGo:^uni VerbSurface;         //adding suffix uni

LEXICON VerbSurface
#;
```

Figure 6.4: Lexical script file

A graphical representation of a lexicon for simple roots of ቅትል-*qtl*, ንብብ-*nbb* and ንብር-*nbr* is demonstrated in Figure 6.5.

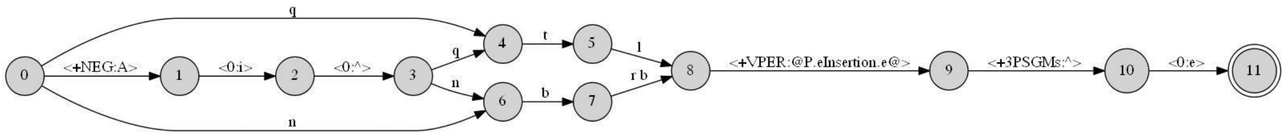


Figure 6. 5 Lexical FST

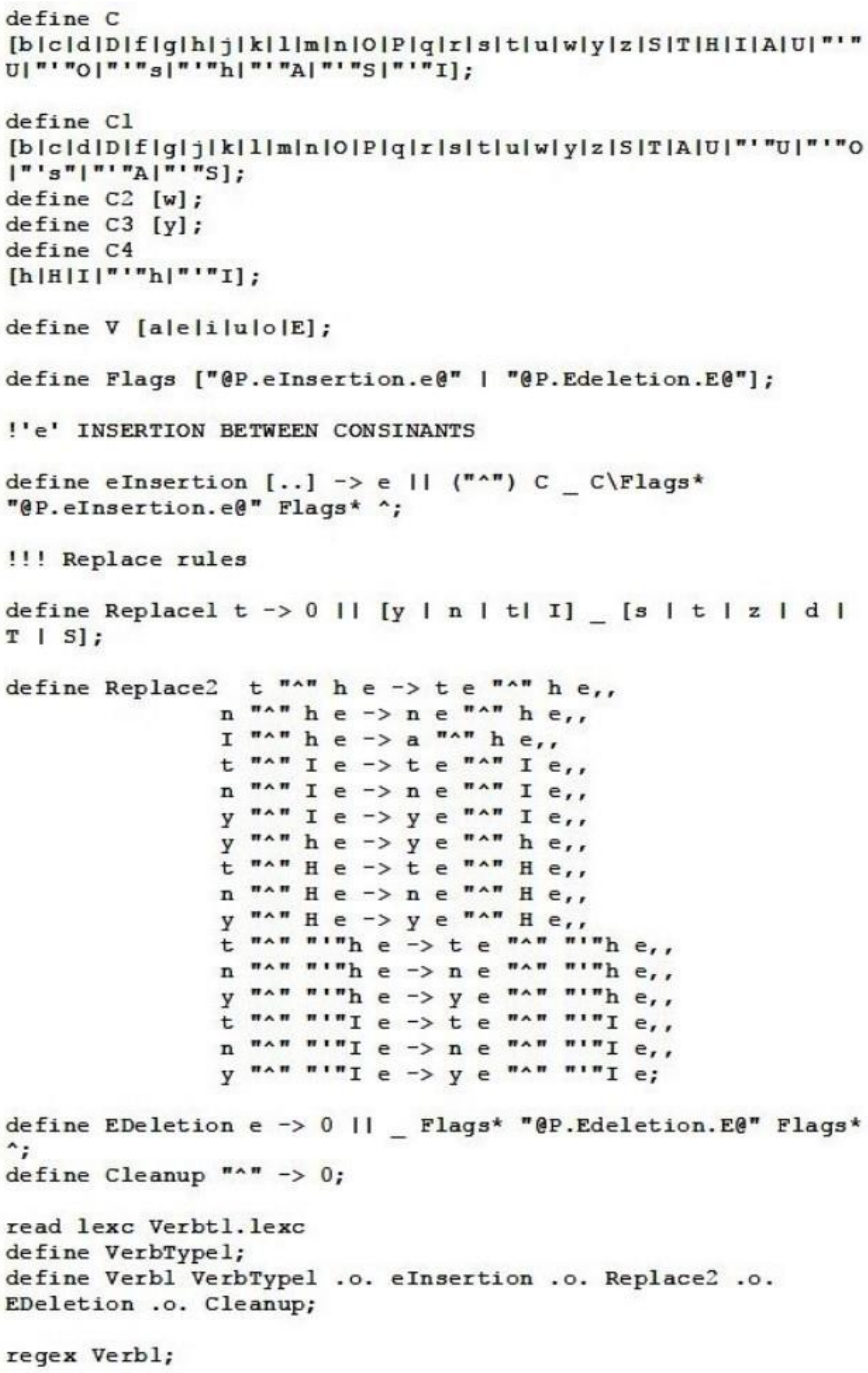
The output generated by the lexical transducer is a direct consequence of the affixation of morphemes to the root. Intercalation of vowels into the root consonants, as well as orthographic and morphophonological alterations to the verbal stems, are achieved through the use of the alternation rule component.

6.3.5 The Alternation Rule

The alternation rule component plays a crucial role in modifying the output of the lexical transducer according to morphophonological rules, ultimately generating a valid surface form (Foma: Morphological Analysis with FSTs, 2011). In the Ge'ez morphological analyzer, this alternation rule component is responsible for intercalating vowels into the root consonants and implementing morphophonological changes on the verbal stems to produce the surface form.

The alternation rules are stored in a text file, following the Foma formalism. This file comprises definitions and rewrite rules that are compiled into a rule FST. Additionally, this script file is used to read the lexical files and label them for subsequent use in regular expressions. The lexical FSTs and rule FSTs are combined using the composition operator to create the final transducer.

Figure 6.6 serves as an illustrative example of an alternation rule script file, stored in Foma format. This file comprises definitions, rewrite rules, commands for reading and labeling lexical script files, and composition operators. The graphical representation of the resulting transducer can be seen in Figure 6.7.



```
define C
[b|c|d|D|f|g|h|j|k|l|m|n|O|P|q|r|s|t|u|w|y|z|S|T|H|I|A|U|"'"
U|"'"O|"'"s|"'"h|"'"A|"'"S|"'"I];

define C1
[b|c|d|D|f|g|j|k|l|m|n|O|P|q|r|s|t|u|w|y|z|S|T|A|U|"'"U|"'"O
|"'s"|"'"A|"'"S];
define C2 [w];
define C3 [y];
define C4
[h|H|I|"'"h|"'"I];

define V [a|e|i|u|o|E];

define Flags ["@P.eInsertion.e@" | "@P.Edeletion.E@"];

!'e' INSERTION BETWEEN CONSINANTS

define eInsertion [..] -> e || ("^") C _ C\Flags*
"@P.eInsertion.e@" Flags* ^;

!!! Replace rules

define Replace1 t -> 0 || [y | n | t| I] _ [s | t | z | d |
T | S];

define Replace2  t "^" h e -> t e "^" h e,,
               n "^" h e -> n e "^" h e,,
               I "^" h e -> a "^" h e,,
               t "^" I e -> t e "^" I e,,
               n "^" I e -> n e "^" I e,,
               y "^" I e -> y e "^" I e,,
               y "^" h e -> y e "^" h e,,
               t "^" H e -> t e "^" H e,,
               n "^" H e -> n e "^" H e,,
               y "^" H e -> y e "^" H e,,
               t "^" "'"h e -> t e "^" "'"h e,,
               n "^" "'"h e -> n e "^" "'"h e,,
               y "^" "'"h e -> y e "^" "'"h e,,
               t "^" "'"I e -> t e "^" "'"I e,,
               n "^" "'"I e -> n e "^" "'"I e,,
               y "^" "'"I e -> y e "^" "'"I e;

define EDeletion e -> 0 || _ Flags* "@P.Edeletion.E@" Flags*
^;
define Cleanup "^" -> 0;

read lexc Verbt1.lexc
define VerbType1;
define Verb1 VerbType1 .o. eInsertion .o. Replace2 .o.
EDeletion .o. Cleanup;

regex Verb1;
```

Figure 6. 6 Foma script file

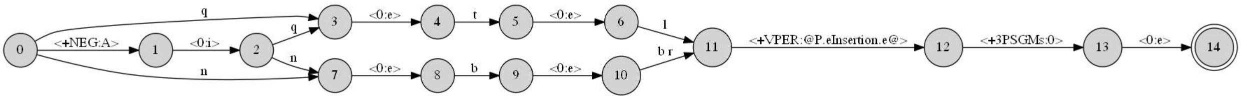


Figure 6. 7 FST

6.3.6 Ge’ez Rule Component

The Foma script file for the Ge’ez morphological analyzer encompasses the following components:

- **Definition of the language alphabet -** This section explicitly defines the consonants and vowels that constitute the language alphabet.
- **Definition of the flags -** In this section, the necessary flags are defined for the analysis process.
- **Defining and labeling the alternation rules -** The alternation rule component, a core part of the analyzer, is defined and labeled within this script file. It outlines the rules governing morphophonological alterations.
- **Reading and labeling of the lexical files -** The script is responsible for reading and labeling lexical script files, such as the one illustrated in Figure 6.4.
- **Composition of the lexical FSTs and the rule FST -** This process brings together the lexicon FSTs and alternation rule FST.
- **Compiling the FSTs to produce the final finite-state transducer -** Once all the components are in place, the script compiles the finite-state transducers to generate the final Ge’ez verb morphological analyzer.

The subsequent sections elaborate on the Ge’ez alternation component, which is saved in the .Foma file

6.3.7 Definition of the Language Alphabet

The Ge’ez letters are defined using the Foma define command.

define C [b |c |d |D |f |g |h |j |k |l |m |n |O |P |q |r |s |t |u |w |y |z |S |T |H |I|A |U | ’ U | ’ O| ’ s| ’ h| ’ A| ’ S| ’ I| x[2]];

define C1 [b |c |d |D |f |g |j |k| m |n |O |P |q |r |s |t |u |w |y |z |S |T |A |U |’U| ’ O| ’s|| ’A| ’ S];

[2] x is not a valid letter and it is used as a variable in replace rule

define C2 [b |c |d |D |f |g |h |j |k |l |m |n |O |P |q |r |s |t |u |y |z |S |T |H |I |A|U | ’ U | ’ O| ’ s| ’ h| ’ A| ’ S| ’ I];

define C3 [b |c |d |D |f |g |h |j |k |l |m |n |O |P |q |r |s |t |u |w |z |S |T |H |I |A|U | ’ U | ’ O| ’ s| ’ h| ’ A| ’ S| ’ I]; define C4 [I | ’I| h| ’h| H];

define C5 [w];

define C6 [I| ’I];

define C7 [h| ‘h| H];

define V [a |e |i |u |o |E];

6.3.8 Definition of Flags

To define the flags, the following format is used:

define Flags @P.XInsertion.X@

Where X represents the type of vowel that is inserted into the consonants. For instance, @P.einsertion.e@ indicates a flag used for the insertion of the vowel e.

6.3.9 Vowel Intercalation Rule

This section outlines the rules designed for the intercalation of vowels into the root consonants.

The vowel intercalation rules are employed to insert one or more vowels into the root consonants in accordance with the conjugation pattern of the verb types in Ge’ez. In Ge’ez verb intercalation, two types of vowel insertion rules are defined. One type involves the unconditional insertion of vowels into root consonants. The other type pertains to the conditional insertion of vowels into root consonants, which is influenced by the presence of specific consonants such as እ-I, ዕ-’I, ህ-h, ኅ-’h, ሕ-H, ው-w, and ይ-y in the roots (please refer to Chapter 2 for more detail). These intercalation rules are designed according to the CV template that characterizes Ge’ez verb formation. Table 6.3 presents the CV template for the intercalation of vowels into consonants, categorized according to the eight verb types.

Table 6.3: CV template

Verb Type	**1** ቀተለ **qetele**	**2** ቀደሰ **qedese**	**3** ገብረ **gebre**	**4** አእመረ **AImere**	**5** ባረከ **bareke**	**6** ሤመ **’sEme**	**7** ብህለ **bhle**	**8** ቆመ **qome**
Perfective	CVCVC	CVCVC	CVCC	CVCCVC	CV1CVC	CVCVC	CCC	CVCVC
Indicative	CVCC	CV2CC	CVCC	CVCVCC	CV1CC	CVCC	CCC	CVCC
Subjunctive	CCC	CVCC	CCVC	CVCCC	CV1CC	CCC	CCVC	CCC
Jussive	CCC	CVCC	CCVC	CVCCC	CV1CC	CCC	CCVC	CCC
Gerundive	CVCV2C	CVCC	CVCV2C	CVCCVC	CV1CVC	CVCC	CCV2C	CVCV2C
Infinitive	CVCV2C	CVCC	CVCV2C	CVCCC	CV1CC	CVCC	CCV2C	CVCV2C

V = e, V1 = a V2 = i

Based on the Ge’ez verb intercalation CV template, the defined vowel intercalation rules are described below:

1. Insertion of the vowel e between two or more constants root - C e C

   define eInsertion1 [..] →e || (“ˆ”) C_ C\Flags* “@P.eInsertion1.e@” Flags*ˆ;

2. Insertion of the vowel e between the first two consonants - C e C C

   define eInsertion2 [..]→e || (“ˆ”) C _ C C\Flags* “@P.eInsertion2.e@” Flags*ˆ;

3. Insertion of the vowel e between the last two consonants - C C e C

   define eInsertion3 [..]→e || “ˆ”C C_ C\Flags* “@P.eInsertion3.e@” Flags*ˆ;

4. Insertion of the vowel e between two or more constants root provided that the second consonant different form the consonants I, ‘I, h, ‘h and H - C e C1

   define eInsertion4 [..]→e || (“ˆ”) C _ C1\Flags* “@P.eInsertion4.e@” Flags*ˆ;

5. Insertion of the vowel e between the last two consonants provided that the second consonant is one of the consonants I, ‘I, h, ‘h and H- C C4 e C

   define eInsertion5 [..]→e || “ˆ”C C4 _ C\Flags* “@P.eInsertion5.e@” Flags*ˆ;

6. Insertion of the vowel e between the first two consonants and between the second and third consonants for a four consonants root - C e C e C C

   define eInsertion6 [..]→e || “ˆ”C _ C C C\Flags* “@P.eInsertion6.e@” Flags*ˆ.o.

[..]→e || “ˆ” C V C_ C C\Flags* “@P.eInsertion6.e@” Flags*ˆ;

7. Insertion of the vowel e between the first two consonants for a four consonants root - C e C C C

define eInsertion7 [..]→e || “ˆ”C _ C C C\Flags* “@P.eInsertion7.e@” Flags*ˆ;

8. Insertion of the vowel e between the first two consonants and a conditional insertion of the vowel e between the second and third consonants provided that third consonant is different form the consonants l, ‘l, h, ‘h and H - C e C e C1 or C e C C4

define eInsertion8 [..]→e || “ˆ”C _ C C \Flags* “@P.eInsertion8.e@” Flags*ˆ.o.

[..]→e || “ˆ” C V C_ C1 \Flags* “@P.eInsertion8.e@” Flags*ˆ;

9. Conditional insertion of the vowels e between consonants with the following condition:

- Insertion of e between the last two consonants when the second consonant is one of the consonants l, ‘l, h, ‘h and H - C C4 e C

- Insertion of e between the last two consonants when the first consonant is w and the second consonant is different from the consonants l, ‘l, h, ‘h and H - C5 C1 e C

- No insertion of vowels between the consonants if the first consonant is not w and the second consonant is different from the consonants l, ‘l, h, ‘h and H - C2 C1 C

define eInsertion9 [..]→e || “ˆ”C C4 _ C\Flags* “@P.eInsertion9.e@” Flags*ˆ.o.

[..]→e || “ˆ”C5 C1_ C\Flags* “@P.eInsertion9.e@” Flags*ˆ;

10. Insertion of the vowel e between the first two consonants provided that the second consonant is not one of the consonants l, ‘l, h, ‘h and H- C e C1 C

define eInsertion10 [..]→e || “ˆ”C _ C1 C\Flags* “@P.eInsertion10.e@”Flags*ˆ;

11. Insertion of the vowel E between two or more constants root - C E C

define EInsertion1 [..]→E || (“ˆ”) C _ C\Flags* “@P.EInsertion1.e@ ” Flags*ˆ;

12. The rule for the Insertion of the vowel a between two or more constants root - C a C

define aInsertion1 [..]→a || (“ˆ”) C_ C\Flags* “@P.aInsertion2.a@” Flags*ˆ;

13. Insertion of the vowel a between the first two consonants - C a C C

    define aInsertion2 [..]→a || “ˆ”C_ C C\Flags* “@P.aInsertion2.a@” Flags*ˆ;

14. Insertion of the vowel a between the last two consonants - C C a C

    define aInsertion3 [..]→a || “ˆ”C C_ C\Flags* “@P.aInsertion3.a@” Flags*ˆ;

15. Insertion of the vowel i between the last two consonants - C C i C

    define iInsertion1 [..]→i || “ˆ”C C_ C\Flags* “@P.iInsertion1.i@” Flags*ˆ;

16. Insertion of the vowel E between the first two consonants and the insertion of the vowel e or x between the last two consonants - C E C e C1 or C E C x C4

    define EeInsertion1 [..]→ E || “ˆ”C _ C C\Flags* “@P.EeInsertion1.Ee@” Flags*ˆ.o. [..]→ e || “ˆ”C V C _ C1\Flags* “@P.EeInsertion1.Ee@” Flags*ˆ.o. [..]→x || “ˆ”C V C_ C4\Flags* “@P.EeInsertion1.Ee@” Flags*ˆ;

17. Conditional insertion of the vowels E, e and a between consonants with the following condition:

    - Insertion of the vowel E between the first two consonants for a three consonants root provided that the last consonant is different from the consonants l, ‘l, h, ‘h and H - C E C C1
    - Insertion of the vowel E between the first two consonants and insertion of x between the last two consonants for a three consonants root provided that the last consonant is one of the consonants l, ‘l, h, ‘h and H - C E C x C4
    - Insertion of the vowel E between the second and third consonants and insertion of e between the last two consonants for a four consonants root provided that the last consonant is different from the consonants l, ‘l, h, ‘h and H - C C E C e C1
    - Insertion of the vowel E between the second and third consonants and insertion of x between the last two consonants for a four consonants root provided that the last consonant is one of the consonants l, ‘l, h, ‘h and H - C C E C x C4

    define EeaInsertion1 [..]→ E || “ˆ” C C _ C C\Flags*“@P.EeaInsertion1.Eea@” Flags*ˆ.o. [..]→ e || “ˆ” C C V C_C1\Flags*“@P.EeaInsertion1.Eea@” Flags*ˆ.o. [..]→ x || “ˆ” C C V C _ C4\Flags*“@P.EeaInsertion1.Eea@”Flags*ˆ.o. [..]→E||“ˆ”C _ C

C\Flags* “@P.EeaInsertion1.Eea@” Flags*ˆ.o. [..]→ x || “ˆ”C V C _ C \Flags* “@P.EeaInsertion1.Eea@” Flags*ˆ;

18. Insertion of the vowel a between the first two consonants and insertion of the vowel e between the last two consonants - C a C e C

define aeInsertion1 [..]→ a || “ˆ”C _ C C\Flags* “@P.aeInsertion1.ae@” Flags*ˆ.o. [..]→ e || “ˆ”C V C _ C\Flags* “@P.aeInsertion1.ae@” Flags*ˆ;

19. Conditional insertion of the vowels a between the first two consonants and the insertion of the vowel e between last two consonants provided that the last consonant is different from the consonants l, ‘l, h, ‘h and H - C a C e C1 or C a C C4

define aeInsertion2 [..]→ a || “ˆ”C _ C C\Flags* “@P.aeInsertion2.ae@” Flags*ˆ.o. [..]→ e || “ˆ”C V C _ C1\Flags* “@P.aeInsertion2.ae@” Flags*ˆ;

20. Conditional insertion of the vowels e between the first two consonants and the insertion of the vowel e or a between last two consonants provided that the last consonant based on the presence of the consonants l, ‘l, h, ‘h and H - C e C e C1 or C e C a C4

define aeInsertion3 [..]→ e || “ˆ”C _ C C\Flags* “@P.aeInsertion3.ae@” Flags*ˆ.o. [..]→ e || “ˆ”C V C _ C1\Flags* “@P.aeInsertion3.ae@” Flags*ˆ.o. [..]→a || “ˆ”C V C _ C4\Flags* “@P.aeInsertion3.ae@” Flags*ˆ;

21. Conditional insertion of the vowels e and the vowel a provided that the first consonant is not from the consonants l, ‘l, h, ‘h and H with the following additional condition:

- Insertion of the vowel e between the first two consonants provided that the last consonant is different from the consonants l, ‘l, h, ‘h and H - C1 e C C1
- Insertion of the vowel a between the last two consonants last consonant is one of the consonants l, ‘l, h, ‘h and H - C1 C a C4

define aeInsertion4 [..]→ e || “ˆ”C1 _ C C1\Flags* “@P.aeInsertion4.ae@” Flags*ˆ.o. [..]→ a || “ˆ”C1 C _ C4\Flags* “@P.aeInsertion4.ae@” Flags*ˆ;

22. Conditional insertion of the vowels e and a between consonants with the following condition:

- Insertion of e between the last two consonants when the third consonant is different from the consonants l, ‘l, h, ‘h and H - C C e C1

- Insertion of a between the last two consonants when the third consonant is one of the consonants l, ‘l, h, ‘h and H - C C a C4

define aeInsertion5 [..]→ e || “ˆ”C C _ C1\Flags* “@P.aeInsertion5.ae@” Flags*ˆ.o. [..]→ a || “ˆ”C C _ C4\Flags* “@P.aeInsertion5.ae@” Flags*ˆ;

23. Conditional insertion of the vowels e and a between consonants with the following condition:

- Insertion of e between the last first consonants - C e C C
- Insertion of a between the first two consonants when the first consonant is one of the consonants l, ‘l and the third consonant is one of the consonants h, ‘h and H - C6 a C C7

define aeInsertion6 [..]→ e || “ˆ”C _ C C \Flags* “@P.aeInsertion6.ae@” Flags*ˆ.o. e→ a || “ˆ”C6 _ C C7 \Flags* “@P.aeInsertion5.ae@” Flags*ˆ;

24. Conditional insertion of the vowels e and the consonant x provided that the first consonant is not from the consonants l, ‘l, h, ‘h and H with the following additional condition:

- Insertion of the vowel e between the last two consonants provided that the last consonant is different from the consonants l, ‘l, h, ‘h and H - C1C e C1
- Insertion of the consonant x between the last two consonants provided that the last consonant is one of the consonants l, ‘l, h, ‘h and H - C1C a C4

define aeInsertion7 [..]→ e || “ˆ”C1 C _ C1\Flags* “@P.aeInsertion7.ae@” Flags*ˆ.o. [..]→ x || “ˆ”C1 C _ C4\Flags* “@P.aeInsertion7.ae@” Flags*ˆ;

25. Conditional insertion of the vowels e and a between consonants with the following condition:

- Insertion of e between the first two consonants for a three consonants root - C e C C

- Insertion of e between the second and third consonants and the last two consonants for a four consonant root provided that the last consonant is different form the consonants l, ‘l, h, ‘h and H - C C e C e C1

- The insertion of e between the first, second and third consonants and the insertion of a between the last two consonants for a four consonant root provided that the last consonant is one of the consonants I, ‘I, h, ‘h and H - C C e C a C4

  define aeInsertion8 [..]→ e || “ˆ” C C _ C C\Flags* “@P.aeInsertion8.ae@” Flags*ˆ.o. [..]→ e || “ˆ” C C V C _ C1\Flags* “@P.aeInsertion8.ae@” Flags*ˆ.o.[..]→ a || “ˆ” C C V C _ C4\Flags* “@P.aeInsertion8.ae@” Flags*ˆ.o. [..]→ e || “ˆ” C _ C C\Flags* “@P.aeInsertion8.ae@” Flags*ˆ;

26. Conditional insertion of the vowels a and e between consonants with the following condition:

    - Insertion of the vowel a between the first two consonants for a three consonants root provided that the second consonant is one of the consonants I, ‘I, h, ‘h and H - C a C4 C

    - Insertion of the vowel a between the first two consonants and the insertion of the vowel e between the last two consonants provided that the second consonant is different from the consonants I, ‘I, h, ‘h and H - C a C1 e C

    - Insertion of the vowel a between the first two consonants and the insertion of the vowel e between the last two consonants for a four consonants root, provided that the second consonant is one of the consonants I, ‘I, h, ‘h and H - C a C4 C e C

    define aeInsertion9 [..]→ a || “ˆ”C _ C C\Flags* “@P.aeInsertion9.ae@” Flags*ˆ.o. [..]→ e || “ˆ”C V C1 _ C\Flags* “@P.aeInsertion9.ae@” Flags*ˆ.o. [..]→ e || “ˆ”C V C4 C _ C\Flags* “@P.aeInsertion9.ae@” Flags*ˆ;

27. Conditional insertion of the vowels a and e between consonants with the following condition:

    - Insertion of the vowel a between the first two consonants for a three consonants root - C a C C

    - Insertion of the vowel a between the first two consonants and the insertion of the vowel e between second and third consonants, for a four consonants root - C a C e C C

```
define aeInsertion1o [..]→ a || “ˆ”C _ C C\Flags* “@P.aeInsertion1o.ae@” Flags*ˆ.o.
[..]→ e || “ˆ”C V C _ C C\Flags* “@P.aeInsertion1o.ae@” Flags*ˆ;
```

28. Insertion of the vowel e between the first two consonants, the insertion of the vowel a between the second and third consonants and the insertion of the vowel e between the last two consonants for a four consonants root - C e C a C e C

```
define aeInsertion11 [..]→      e || “ˆ”C _ C C C\Flags* “@P.aeInsertion11.ae@”

Flags*ˆ.o. [..]→a || “ˆ”C V C _ C C\Flags* “@P.aeInsertion11.ae@” Flags*ˆ.o.

[..]→    e || “ˆ”C V C V C _ C\Flags* “@P.aeInsertion11.ae@” Flags*ˆ;
```

29. Insertion of the vowel a between the second and third consonants and the insertion of the vowel e between the last two consonants for a four consonants root - C C a C e C

```
define aeInsertion12 [..]→ a || “ˆ”C C _ C C\Flags* “@P.aeInsertion12.ae@” Flags*ˆ.o.
[..]→ e || “ˆ”C C V C _ C\Flags* “@P.aeInsertion12.ae@” Flags*ˆ;
```

30. Insertion of the vowel e between the first two consonants and the insertion of the vowel a between the second and third consonants for a four consonants root - C e C a C C

```
define aeInsertion13 [..]→ e || “ˆ”C _ C C C\Flags* “@P.aeInsertion13.ae@” Flags*ˆ.o.
[..]→ a || “ˆ”C V C _ C C\Flags* “@P.aeInsertion13.ae@” Flags*ˆ;
```

31. Conditional insertion of the vowels e and u between consonants with the following condition:

- insertion of e between the first two consonants where the second consonant is different from the consonants l, ‘l, h, ‘h and H – C e C1 C
- insertion of u between the last two consonants if the first consonant is w and the second consonant is one of the consonants l, ‘l, h, ‘h and H - C5 C4 u C
- no insertion of vowels if the first consonant is not w and the second consonant is one of the consonants l, ‘l, h, ‘h and H - C C C

```
define euInsertion1 [..]→ e || “ˆ”C _ C1 C\Flags* “@P.euInsertion1.e@” Flags*ˆ.o. [..]→
u || “ˆ”C5 C4 _ C\Flags* “@P.euInsertion1.e@” Flags* ˆ;
```

32. Insertion of the vowel e between the first two consonants and the insertion of the vowel i between the last two consonants - C e C i C

    define eiInsertion1 [..]→e || “ˆ”C _ C C\Flags* “@P.eiInsertion1.ei@” Flagsˆ*.o.

    [..]→i || “ˆ”C V C_ C\Flags* “@P.eiInsertion1.ei@” Flags*ˆ;

33. Insertion of the vowel e between the first two consonants and the insertion of the vowel i between the last two consonants for a four consonants root - C e C C i C

    define eiInsertion2 [..]→ e || “ˆ”C C C C\Flags* “@P.eiInsertion2.ei@” Flags*ˆ.o. [..]→ i || “ˆ”C V C C C\Flags* “@P.eiInsertion2.ei@” Flags*ˆ;

34. Insertion of the vowel a between the first two consonants and the insertion of the vowel i between the last two consonants - C a C i C

    define aiInsertion1 [..]→ a || “ˆ”C _ C C\Flags* “@P.aiInsertion1.ai@” Flags*ˆ.o. [..]→ i || “ˆ”C V C _ C\Flags* “@P.aiInsertion1.ai@” Flags*ˆ;

6.3.10 Alternation Rules

This section describes rules that are defined for the morphophonological alternation that occur due to concatenation of morphemes. Similar to the vowel intercalation rule, the alternation rules include rules for the changes that occur due to the presence of the consonants ኧ-*l*, ዕ-*’l*, ሀ-*h*, ኀ-*’h*, ሐ-*H*, ው-*w* and ይ-*y*.

The alternation rules for the changes that occur during affixation are described below:

1. Rule for the deletion of the last vowels (u, e, i and a):

    - define UDeletion u→0 ||Flags* “@P.Udeletion.U@” Flags*ˆ;
    - define EDeletion e→ 0 ||Flags* “@P.Edeletion.E@” Flags*ˆ;
    - define IDeletion i→ 0 ||Flags* “@P.Ideletion.I@” Flags*ˆ;
    - define ADeletion a→ 0 ||Flags* “@P.Adeletion.A@” Flags*ˆ;

2. Rules for the alternations that occur for roots containing the consonants I, ‘I, h, ‘h and H

- Rule for the alternation that occurs when the subject marker prefixes are concatenated with verbal stems beginning with A, ‘A, he, ’he or He for qetele, qedese, gebre, sEme and qome verb types (for indicative).

  - define Replace1 t “^” h e → t e “^” h e,,

```
n “^” h e → n e “^” h e,,
I “^” h e → a “^” h e,,
t “^” I e → t e “^” I e,,
n “^” I e → n e “^” I e,,
y “^” I e → y e “^” I e,,
y “^” h e → y e “^” h e,,
t “^” H e → t e “^” H e,,
n “^” H e → n e “^” H e,,
y “^” H e → y e “^” H e,,
t “^” “ ‘ ”h e → t e “^” “ ‘ ”h e,,
n “^” “ ‘ ”h e → n e “^” “ ‘ ”h e,,
y “^” “ ‘ ”h e → y e “^” “ ‘ ”h e,,
t “^” “ ‘ ”I e → t e “^” “ ‘ ”I e,,
n “^” “ ‘ ”I e → n e “^” “ ‘ ”I e,,
y “^” “ ‘ ”I e → y e “^” “ ‘ ”I e;
```

3. Rule for the alternation that occur when the subject marker prefix are concatenated with a verbal stem beginning with Ia. This rule applies to gebre type verbs

   - define Replace2 I→0 || [y | n | t| I] _ a;

4. Rule for the alternation that occurs when the root contains the letter ‘l’ at the beginning of the root, l is removed for the indicative, subjunctive and jussive verb forms. These rules apply for the llmr verb types.

    - define Replace 3 y “ˆ” l e→ y “ˆ” a, t “ˆ” l e→ t “ˆ” a, n “ˆ” l e→ n “ˆ” a, l “ˆ” l e→ l “ˆ” e;

    - define Replace 4 y “ˆ” l e l e→ y “ˆ” e l e, t “ˆ” l e l e→ t “ˆ” e l e, n “ˆ” l e l e→ n “ˆ” e l e, l “ˆ” l e l e→ l “ˆ” e l e;

5. Rule for the alternation that occurs when a verbal stem contains the consonant h. These rules apply for the bhle verb type.

    - define Replace 5 h→ 0 || [y | n | t| l] “ˆ”C _ C;

6. Rules for the alternations that occur for roots containing the consonants w

    - Rule for the alternation that occurs when a verbal stem begins with the consonant w followed by a consonant. This rule applies to qetele and gebre verb types.

    - Define Replace 6 w→ 0 || \V “ ˆ” _ C;

7. Rule for the alternation that occurs when a verbal stem contains the consonant w for three consonants roots and four consonants root. These rules apply for the qome verb type.

    - define Replace 7 e w e→ o || \C “ˆ”C _ C2;

    - define Replace 8 e w e→ o || “ˆ”C _ C2;

    - define Replace 9 e w→ o || “ˆ”C _ C C;

    - define Replace 10 w→ o || “ˆ”C _ C C;

    - define Replace 11 w→ u || “ˆ”C _ \[C6 | C5]; define Replace12 w→ u || “ˆ”C _ C6;

8. Rules for the alternations that occur for roots containing the consonants y

- Rule for the alternation that occurs when a verbal stem contains the consonant y. This rule applies for the ‘bhle verb type.

- define Replace 13 y→i || \[C |V] “ˆ”C C3 _;

9. Rule for the alternation that occurs when a verbal stem contains the consonant y for three consonants roots and four consonants root. These rules apply for the ‘sEme verb type.

    - define Replace 14 e y e →E || \C “ˆ”C _ C3; define Replace15 e y → E || “ˆ”C _ C C;

    - define Replace 16 y → E || “ˆ”CC _ C;

    - define Replace 17 y →i || “ˆ”C _ C;

10. Rules for the alternations that occur for irregular verbs

- This rule applies to the verb rIye (a troop of the bhle head verb).

    - define Replace 18 [..]→ E || [y | n | t| l] “ˆ”r _ I y;

11. Rule for the alternation that occurs for the irregular verb bEle. This rule applies for the bEle verb type.

    - define Replace 19 l → 0 || “ˆ”C E _;

12. Rules for the alternations that occur for all types of verbs

- Rule for the alternation that occurs when the verbal stem ending with the letters k, g, q or n is concatenated with a subject marker morpheme beginning with the letter k.

    - define Replace C1 k → 0 || [k | g | q] _ [C | V |.#.];

    - define Replace C2 n → 0 || n _ e.#.;

13. Rule for the alternation that occur when a verb ends with the consonant w but not a double consonant of w.

    - define Replace C3 w → u || C2 _.#.;

    - define Replace C4 e w → o || C2 _.#.;

14. Rule for the alternation that occur when a verb ends with the consonant y but not a double consonant of y.

- define Replace C5 i→ y || C _ V ([C |V]+).#.;

- define Replace C6 y → i || C3 _.#.;

15. Rule for the alternation that occurs when the derivational verb morpheme t is concatenated with verbal stem beginning with s, ’s, t, y, d, T, S or ’S.

    - define Replace C7 t→ 0 || [y | n | t| l] _ [s | ’s | t | z | d | T | S | ’S];

16. Rule for the forms of the letters I and ‘I according to the transliteration described in Chapter 2 and removal of double consonants.

    - define Replace C8 I e → A, I u → U, I i → A i, I a → A a, I E → A E, I o → O, ’I e → ’A, ’I u → ’U, ’I i → ’A i, ’I a → ’A a,’I E → ’A E,’I o→ ’O, A A → A, I I → I;

17. Rules for the alternations that occur for roots containing the consonants I, ‘I, h, ‘h and H

    - define Replace C9 x → 0 || C _ C4 ([C |V]+).#.

    - define Replace C1o x → a || C _ C4.#.

18. Rule for the removal of “ˆ”

    - define Cleanup “ˆ” →0.

By combining the alternation rule FSTs, which govern morphophonological alterations and intercalation of vowels, with the lexical FSTs, which encapsulate the lexicon and affixation rules, the final finite-state transducer emerges.

6.3.11 Composition

The composition operator '.o.' in Foma is employed to merge the lexical transducers with the rule transducers, ultimately creating the final FST.

For each Ge’ez verb type, its respective lexical transducer is combined with the rule FSTs that are applicable to it. This process results in the creation of intermediate Ge’ez verb type FSTs, each tailored to its respective verb type. These intermediate FSTs are then compiled together to form a single, comprehensive FST. As discussed in section 6.3.6, some alternation rules are universal and apply to all verb types, while others are specific to particular verb types.

Consequently, the compiled FST is further merged with the rule FSTs that apply universally to all verb types.

This combination yields the final transducer, which serves as the finite-state-based morphological analyzer for Ge’ez verbs. Figure 6.8 provides an illustration of the composition of the lexical transducers and the rule transducers..

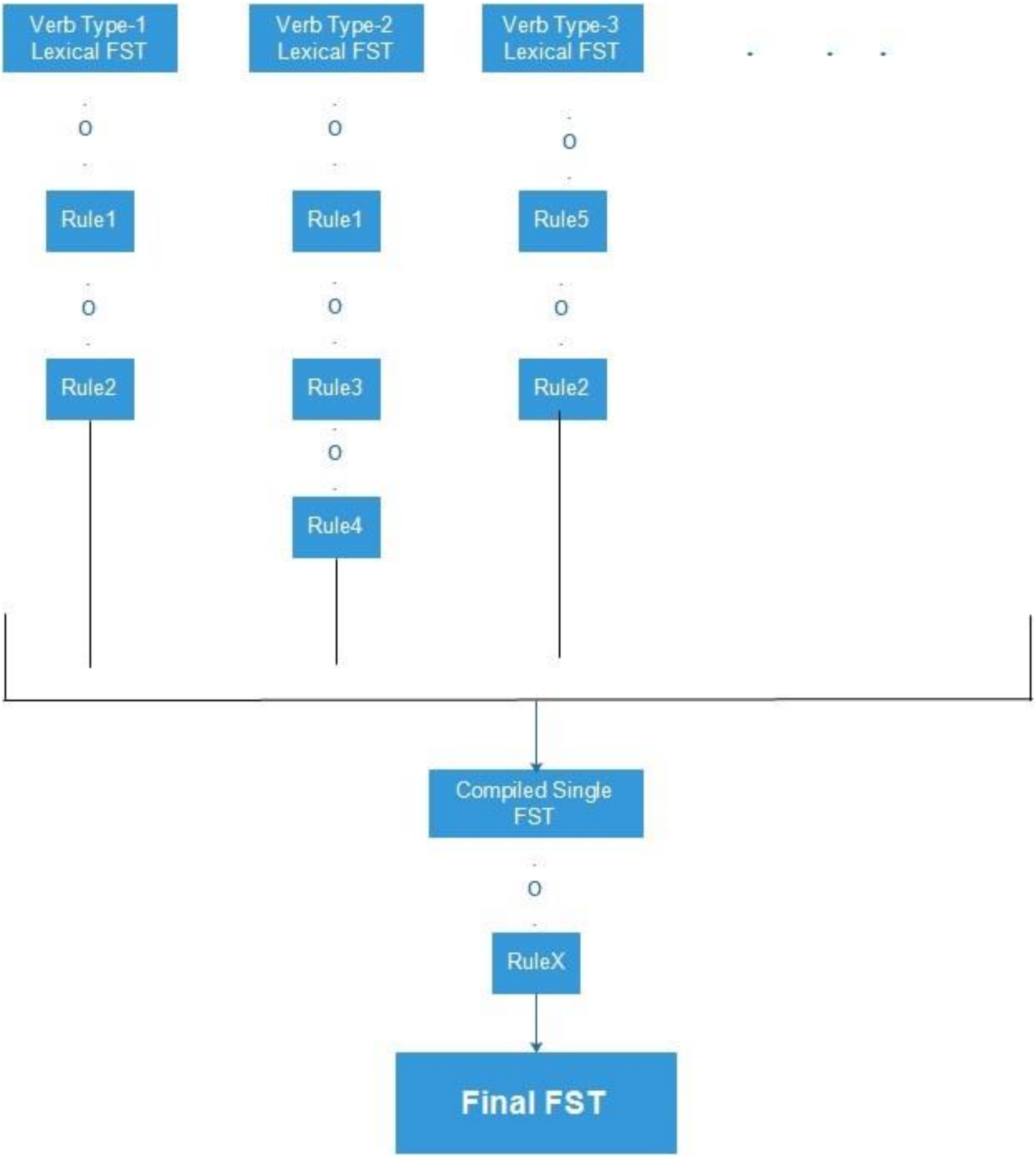


Figure 6.8: Lexical and rule FST composition

The creation of the Ge’ez morphological analyzer involves a sequence of well-defined steps, culminating in the final finite-state morphological analyzer. Below is a description of the steps:

1. Each lexical script file corresponding to the verb types is read and labeled. Following labeling, the composition operator is used to combine the intercalation and alternation rule transducers with their respective lexical transducers. As an example, the Ge’ez Foma script file contains the following entry for Ge’ez verb type -1.

    - **read lexc** VerbType1.lexc
    - **define** VerbType1;
    - **define** Verb1 VerbType1.o. eInsertion1.o. eInsertion2.o. eInsertion3.o. eInsertion9.o. eInsertion12 .o. aInsertion2 .o. iInsertion1 .o. aeInsertion1 o. euInsertion1.o. eiInsertion1 .o. aiInsertion1 .o. Replace6 .o. Replace1;

This entry reads and labels the lexical script file 'VerbType1.lexc' and then defines 'VerbType1.' Subsequently, it combines 'VerbType1' with a series of intercalation and alternation rule transducers using the composition operator.

2. The next step is to compile the transducers obtained from the composition of the lexicon and the rule. This is typically done using a regular expression that combines the different verb type, like:

    - **regex Verb1 | Verb2 |...;**
    - **define LexiconX;**

In this script, 'Verb1,' 'Verb2,' and so on represent the compiled verb transducers obtained from the previous composition step. These transducers are combined using the 'regex' operator to form 'LexiconX,' which represents the compiled single transducer.

3. Finally, the composition of the combined transducer and the alternation rules that apply to all verb types is performed. This results in the creation of the final transducer, which serves as the Ge’ez morphological analyzer. Here's the script for this composition:

    - **regex** LexiconX.o. ReplaceY.o. ReplaceYa.o.....;

In this script, 'LexiconX' represents the previously compiled lexicon transducer, and 'ReplaceY,' 'ReplaceYa,' and so on are the alternation rules that apply universally to all verb types. Combining them using the 'regex' operator yields the final transducer, which functions as the Ge’ez morphological analyzer.

In total, there are ten lexical script files corresponding to the eight head verbs and two irregular verb types. These lexical script files are named as follows: VerbType1, VerbType2, VerbType3, VerbType4, VerbType5, VerbType6, VerbType7, VerbType8, bEle, and Irregular. Additionally, there is a single Foma script file responsible for compiling the lexicon and the rule transducer into the Ge’ez finite-state morphological analyzer.

6.4 Chapter Summary

This chapter has presented the design and implementation of the Ge’ez morphological analyzer, which was developed based on the Ge’ez verb morphological properties outlined in Chapter 2. It explained the utilization of Foma, a finite-state compiler, in the development of the Ge’ez verb morphological analyzer. Additionally, the development process of the Ge’ez morphological analyzer was outlined, addressing the research sub-questions, specifically how to create finite-state transducers representing the morphotactics and orthographic rules of Ge’ez verbs, and how to use finite-state methods in developing such an analyzer.

Appendix II lists the lexical files for each verb type and the Foma file.

The next chapter will delve into data collection, evaluate the Ge’ez morphological analyzer, and present the research findings.

Chapter 7 – Data Collection & Evaluation

7.1 Introduction

The evaluation of this research serves to determine whether the study successfully achieves its objectives. In this study, the focal point of evaluation is the IT artifact, which is the finite-state morphological analyzer designed for Ge’ez verbs. This evaluation process relies on the use of gold-standard data, which are datasets that have been manually annotated with correct analyses. These gold-standard datasets are essential for assessing the accuracy and effectiveness of our morphological analyzer.

In the evaluation process, each word (surface form) in the gold-standard dataset is input into the Ge’ez morphological analyzer. The output analysis generated by our system is then compared with the gold-standard analysis to measure the system's performance.

This chapter will delve into the organization of the gold-standard data for Ge’ez verbs, explain the evaluation methodology employed to assess our system using the gold standard, and discuss evaluation of the research process itself.

7.2 Data Collection

To build a corpus of Ge'ez verbs for evaluation, we engaged the expertise of two Ge'ez language scholars affiliated with the Ethiopian Orthodox Tewahido Church. Their meticulous extraction efforts were primarily centered around two sources: the Ge'ez New Testament Bible and the Ge'ez prayer book, 'ውዳሴ ማርያም' - *'wudase maryam'.* A total of 1,519 verbs were painstakingly extracted from these sacred texts, comprising 1,350 verbs sourced from the books of Matthew, Luke, Mark, and John, and an additional 169 verbs from the *'wudase maryam'* prayer book.

It is important to note that among the collected verbs are those containing the prefixes ወ - *'wä'* and ዘ - *'zä'*. In Ge'ez grammar, the prefix ዘ - *'zä'* conveys the meaning 'the one that.' For example:

- መጽአ - *mäṣʿä* - means he came
- ዘመጽአ - *zämäṣʿä* - means the one that came

Similarly, the prefix ወ - *'wä'* is typically added to a verb when multiple verbs appear in a sentence. For instance, in the phrase 'በልዐ ወሰተየ' - *'bälʿä wäsätäyä'*, meaning 'he ate and drank,' the prefix ወ - *'wä'* is appended to the verb ሰተየ - *'sätäyä'*. In this research, these prefixes, 'ወ' (*wä*) and 'ዘ' (*zä*), were removed as part of data cleaning to ensure consistency and accuracy. Additionally, repeated words and words that are part of other parts of speech (POS) were removed from the dataset.

From the initially collected verbs, a total of 1,365 (89.86%) verbs were extracted as clean data for evaluating the Ge'ez morphological analyzer. From this manually extracted dataset, the distribution of verb types was as follows: 43.59% ቀተለ - *qätälä*, 10.70% ቀደሰ - *qäddäsä*, 22.56% ገብረ - *gäbrä*, 4.10% አእመረ - *äəmärä*, 1.03% ባረከ - *baräkä*, 1.76% ሤመ - *śemä*, 5.57% በህለ - *bəhlä*, 7.33% ቆመ - *qomä*, 1.39% ቤለ - *belä* and 1.98% irregular verbs.

The category of irregular verbs merits special attention, as it encompasses verbs that, while predominantly belonging to one type of head verb, exhibit some inflections resembling those of other head verb types.

In total, our dataset incorporates 439 unique Ge'ez root verbs, with 349 root verbs forming the basis of our test data. Table 7.1 shows the total number of words extracted for each type of verbs and Table 7.2 show the number of roots in each verb type.

Table 7.1: Number of verb types in the test data set

Verb type	**No of verbs**	**Percentage**
ቀተለ - *qätälä* (Verb 1)	595	43.59%
ቀደሰ - *qäddäsä* (Verb 2)	146	10.70%
ገብረ - *gäbrä*, (Verb 3)	308	22.56%
አእመረ - *äəmärä* (Verb 4)	56	4.10%
ባረከ - *baräkä* (Verb 5)	14	1.03%
ሤመ - *śemä* (Verb 6)	24	1.76%
በህለ - *bəhlä* (Verb 7)	76	5.57%
ቆመ - *qomä*, (Verb 8)	100	7.33%
ቤለ - *belä* (Verb 9)	19	1.39%
Irregular verbs (Verb 10)	27	1.98%
Total	**1,365**	**100%**

Table 7.2: Roots in each verb type

Verb type	Total No. of roots	No of roots in the test data
ቀተለ - *qätälä* (Verb 1)	215	178
ቀደሰ - *qäddäsä* (Verb 2)	62	44
ገብረ - *gäbrä*, (Verb 3)	76	62
አእመረ - *äəmärä* (Verb 4)	20	15
ባረከ - *baräkä* (Verb 5)	8	4
ሤመ - *śemä* (Verb 6)	11	10
በህለ - *bəhlä* (Verb 7)	19	15
ቆመ - *qomä*, (Verb 8)	22	17
ቤለ - *belä* (Verb 9)	1	1
Irregular verbs (Verb 10)	5	3
Total	439	349

7.2.1 Manual Analysis of the Verbs

For the evaluation of our Ge’ez morphological analyzer, a critical component is a list of Ge’ez verbs that have been manually annotated with their correct morphological analysis, creating what is commonly referred to as gold-standard data. To achieve this, Ge’ez language experts painstakingly annotated the collected Ge’ez verbs with the accurate morphological analysis. As outlined in Table 7.1, the test dataset encompasses verbs from all verb categories, meticulously analyzed within their respective contexts. An important consideration during this analysis was to provide only a single analysis per word within its context. This intentional limitation was imposed to render the gold-standard test dataset useful for various research purposes, including morphological disambiguation tasks.

The test set provides detailed information about each verb, including its root, the verb type to which it belongs, its tense-mood, object, subject, and whether it involves negation. For instance, let's consider the word ነበቦሙ - *näbäbomu*, which means 'he told them.' The manual analysis of this verb includes the following information:

- The verb is not negated.
- It is in the perfective form.
- It indicates a third person singular male subject.
- It also indicates a third person plural male as the object.
- The root of the verb belongs to verb type 1.
- the root itself is ንብብ ('nbb').

This rich dataset was subsequently transliterated to facilitate testing with our Ge'ez morphological analyzer. For this research study, this data serves as the gold standard against which our analyzer's accuracy is evaluated.

Following the creation of the gold-standard dataset, the next step involved extracting Ge'ez roots from this resource to construct the Ge'ez lexicon, an essential component in the development of our morphological analyzer. A total of 349 Ge'ez roots were systematically extracted from the gold-standard data. In addition, our team of Ge'ez language experts diligently organized an additional 90 Ge'ez roots, which were thoughtfully incorporated into the Ge'ez lexicon. Consequently, our Ge'ez lexicon encompasses a total of 439 roots.

This section, Section 7.2, has discussed the creation of the gold-standard test dataset and the extraction of Ge’ez roots, addressing the research sub-questions related to how to create the lexicon for Ge’ez verbs and how to generate gold-standard test data for evaluating the morphological analyzer.

The next section will delve into the evaluation of the Ge’ez morphological analyzer.

7.3 Evaluating the Ge’ez Morphological Analyzer

The primary aim of this study is to introduce a finite-state-based morphological analyzer for the Ge’ez language. In this section, we assess this artifact, the finite-state morphological analyzer for Ge’ez verbs, using gold-standard test data. The accuracy of the Ge’ez morphological analyzer was assessed by comparing its output to the gold-standard test set. In the initial stages of testing the morphological analyzer, we observed various results: some words were correctly analyzed, others produced multiple outputs, some were incorrect, and some yielded no analysis. The unexpected outputs were attributed to factors such as inaccuracies in verb transliteration, unavailability of roots in the Ge’ez lexicon, and incorrect placement of roots within their respective head verbs. To address these issues, corrections were made in consultation with Ge’ez experts. These corrections included ensuring that the roots were correctly positioned within their head verbs and incorporating all roots found in the test set into the Ge’ez lexicon. These measures significantly improved the accuracy of the Ge’ez morphological analyzer.

After implementing these corrective measures, we conducted a re-evaluation of the Ge’ez morphological analyzer. The accuracy of the morphological analyzer was measured by determining the number of correctly analyzed verbs out of the total number of verbs. In cases where the morphological analyzer produced multiple analyses for a word, we considered the word correct if at least one of the analyses matched the reference (gold standard).

The accuracy of the Ge’ez morphological analyzer was evaluated using the following measure:

Accuracy: The percentage of correctly analyzed verbs out of the total number of verbs analyze.

- Accuracy = (Total number of correctly analyzed verbs / Total number of verbs analyzed) 100%

Precision: The percentage of correctly analyzed verbs out of the total number of analysis outputs generated by the morphological analyzer.

- Precision = (Total number of correctly analyzed verbs / Total number of analysis output by the morphological analyzer) 100%

To evaluate the Ge’ez morphological analyzer, a total of 1,365 verbs were manually annotated from the Ge’ez Bible and Ge’ez prayer book. Among these manually annotated verbs, 1,328 verbs (97.29%) received correct analysis, while 37 verbs (2.71%) did not yield any analysis or output from the analyzer. For a detailed breakdown, please refer to Table 7.3 and Table 7.4, which provide the evaluation results for each type of verb.

Out of the 1,328 correctly analyzed verbs (unique verbs), the analyzer generated an additional 327 possible analyses. The precision of the Ge’ez morphological analyzer was calculated at 80.24%. This figure considers that the analyzer provided all possible analyses for each verb, irrespective of context. Table 7.6 presents the precision and accuracy of the morphological analyzer.

Table 7.3: Number of correctly evaluated verbs

Verb Type	Number of verbs	Correctly analyzed	Accuracy
ቀተለ -qetele (Verb 1)	595	577	96.97%
ቀደሰ -qedese (Verb 2)	146	142	97.26%
ገብረ -gebre (Verb 3)	308	300	97.40%
አእመረ -AImere (Verb 4)	56	54	96.43%
ባረከ -bareke (Verb 5)	14	14	100.00%
ሤመ -’sEme (Verb 6)	24	24	100.00%
ብህለ -bhle (Verb 7)	76	73	96.05%
ቆመ-qome (Verb 8)	100	98	98.00%
ቤለ -bEle (Verb 9)	19	19	100.00%
Irregular verbs (Verb 10)	27	27	100.00%
Total	**1,365**	**1,328**	**97.29%**

Table 7.4: Number of verbs with no analysis

Verb type	Number of verbs	No analysis	% of No analysis
ቀተለ -qetele (Verb 1)	595	18	3.03%
ቀደሰ -qedese (Verb 2)	146	4	2.74%
ገብረ -gebre (Verb 3)	308	8	2.60%
አእመረ -AImere (Verb 4)	56	2	3.57%
ባረከ -bareke (Verb 5)	14	0	0.00%
ሤመ -’sEme (Verb 6)	24	0	0.00%
ብህለ -bhle (Verb 7)	76	3	3.95%
ቆመ-qome (Verb 8)	100	2	2.00%
ቤለ -bEle (Verb 9)	19	0	0.00%
Irregular verbs (Verb 10)	27	0	0.00%
Total	**1,365**	**37**	**2.71%**

Table 7.5: Number of verbs analyzed correctly to the total number of analysis output by the morphological analyzer

Verb type	Correctly analyzed	Morphological analyzer output	Precision
ቀተለ -qetele (Verb 1)	577	721	80.03%
ቀደሰ -qedese (Verb 2)	142	170	83.53%
ገብረ -gebre (Verb 3)	300	368	81.52%
አእመረ -AImere (Verb 4)	54	63	85.71%
ባረከ -bareke (Verb 5)	14	24	58.33%
ሤመ -'sEme (Verb 6)	24	30	80.00%
ብህለ -bhle (Verb 7)	73	90	81.11%
ቆመ-qome (Verb 8)	98	120	81.67%
ቤለ -bEle (Verb 9)	19	39	48.72%
Irregular verbs (Verb 10)	27	30	90.00%
Total	**1,328**	**1,655**	**80.24%**

Table 7.6: Precision and accuracy

Verb Type	Accuracy	Precision
ቀተለ -qetele (Verb 1)	96.97%	80.03%
ቀደሰ -qedese (Verb 2)	97.26%	83.53%
ገብረ -gebre (Verb 3)	97.40%	81.52%
አእመረ -AImere (Verb 4)	96.43%	85.71%
ባረከ -bareke (Verb 5)	100.00%	58.33%
ሤመ -'sEme (Verb 6)	100.00%	80.00%
ብህለ -bhle (Verb 7)	96.05%	81.11%
ቆመ-qome (Verb 8)	98.00%	81.67%
ቤለ -bEle (Verb 9)	100.00%	48.72%
Irregular verbs (Verb 10)	100.00%	90.00%
Total	**97.29%**	**80.24%**

7.3.1 Causes of Errors

As described in Chapter 2, most of the morphophonological alternations occur due to the presence of certain letters in verbs, namely እ- ə, ዕ- ’ə, ህ- h, ኅ- ḫ, ሕ- ḥ, ው-w, and ይ-y. While rules have been defined to account for alternations caused by these letters, discrepancies have been observed in verbs that share the same head verbs.

During the analysis of certain ቤለ- *belä* verbs, the analyzer provided two types of structural information regarding these verbs. One type indicated that the words belonged to the ቤለ - *belä* verb category, while the other suggested membership in the ብህለ - bəhlä verb category. However, all structural information about the words remained the same, except for their categorization as verbs. This discrepancy arises because, in this study, ቤለ - *belä* is considered an irregular verb with two consonants, even though some scholars classify it as an inflectional verb of ብህለ - bəhlä. Consequently, certain inflections of ብህለ - bəhlä overlap with ቤለ - *belä*, including their meanings. It is important to note that this analysis is applicable to only certain ቤለ - *belä* verbs, while others are correctly identified as ቤለ - *belä* verbs.

This discrepancy has resulted in a relatively lower precision value of 80.24%, as indicated in Table 7.6.

7.4 Discussion

As detailed in Section 7.3, when assessed against the gold dataset, the Ge’ez morphological analyzer demonstrated a remarkable accuracy of 97.2% and a precision of 80.24%.

During the initial phases of testing, the accuracy of the morphological analyzer stood at 72%. Several steps were taken to enhance its accuracy. Initially, the manually annotated test set was meticulously reviewed in collaboration with Ge’ez experts. Subsequently, a thorough examination of the transliteration of the test set was carried out, addressing and rectifying errors that stemmed from transliteration. Another crucial investigation focused on the Ge’ez lexicon, ensuring that it encompassed all the roots found in the test set and that these roots were correctly categorized within their respective head groups. Ge’ez language experts conducted a crosscheck to confirm the inclusion of all roots in the Ge’ez lexicon and their proper placement. These meticulous corrections resulted in a significant increase in the accuracy of the Ge’ez morphological analyzer to 97.2%.

The precision of the Ge’ez morphological analyzer stands at 80.24%. This precision value is influenced by the analyzer's practice of generating all possible analyses of verbs, regardless of context. In contrast, the gold-standard test set, as discussed in Section 7.1, was intentionally designed with only one possible analysis for each word in a given context. This design choice aims to facilitate the use of the gold-standard test set by other researchers, particularly for tasks such as morphological disambiguation.

As demonstrated in Table 7.4, no analysis was found for 2.71% of the test set. This issue was primarily attributed to the presence of specific letters እ- ə, ዕ- ʼə, ህ- h, ኅ- ḫ, ሕ- ḥ, ው-w, and ይ-y in the verbs. Of the total words without analysis, 34 out of 37 contained one or more of these letters in their roots. Despite our comprehensive efforts to define all the rules governing morphophonological alternations in Ge'ez verbs due to the presence of these letters, it is evident that some words were not recognized or analyzed by the system.

Chapter 3 discusses two previous studies on Ge'ez language morphological analysis: Desta (2010) and Abate (2014). To the best of our knowledge, Desta (2010) was the first researcher to develop a Ge'ez morphological analyzer for Ge'ez verbs using a rule-based approach. However, his study was limited to a single Ge'ez verb head - ቀተለ - qätälä. Desta (2010) reported an accuracy of 73.98% at the verb level for ቀተለ - qätälä. When compared with our morphological analyzer for the same head verb, our system achieved a higher accuracy of 96.97%.

In contrast, Abate (2014) adopted a data-driven supervised approach to develop a morphological analyzer for Ge'ez verbs. Abate (2014) reported accuracies of 56.3% with the IB2 algorithm and 60.3% with the TRBIL2 algorithm. The data-driven Ge'ez morphological analyzer introduced by Abate (2014) has limitations when analyzing verbs with irregular conjugations. Furthermore, the dataset's constraints, which encompass a detailed derivation of only one head verb category and a sample of other head verb categories, hinder the analyzer's performance in predicting unknown data for verbs in other categories. Our morphological analyzer encompasses all head verbs, including irregular verbs, producing an impressive accuracy of 97.29% for all verb types.

This study entailed meticulous data collection, analysis, and collaboration with Ge'ez language experts to address issues related to transliteration, lexicon completeness, and the accurate categorization of verb roots. These efforts significantly contributed to the accuracy and reliability of your morphological analyzer and also resulted in a gold-standard dataset that can benefit other researchers. Our study culminated in the development of a finite-state-based Ge'ez verb morphological analyzer that covers all Ge'ez verb categories, achieving an accuracy of 97.29%. As a finite-based morphological analyzer, our system can also serve as a generator of Ge'ez verbs. This Ge'ez morphological analyzer holds potential for various Ge'ez language NLP applications. In summary, the Ge'ez morphological analyzer consistently delivered results with an accuracy of 97.29% for analyzed verbs and a precision of 80.24%.

7.5 Evaluation of the Research Process

In this research, the design-and-creation research methodology was employed to design and develop the IT artifact, namely, the finite-state-based morphological analyzer for Ge’ez verbs. The design-and-creation research methodology centers on developing an IT artifact to solve a research problem, thereby contributing to the field of Natural Language Processing (NLP). The methodology comprises five key steps: awareness, suggestions, development, evaluation, and conclusion, as articulated by Oates (2005). Below, we illustrate how our research seamlessly integrated with each of these steps:

1. **Awareness of the Research Problem**: we identified the need for a Ge'ez morphological analyzer and recognized the research problem in the context of NLP for this language.
2. **Suggestions for a Solution**: we proposed a solution to the research problem, which involved developing a finite-state based morphological analyzer using a rule-based approach, with a focus on Ge'ez verbs.
3. **Development of the Solution**: To bring our proposed morphological analyzer to life, we rigorously applied the principles of adaptive software development. Key facets of this approach included modularization, unit testing, composability, system testing, and iterative refinement. The choice of the freely available Foma served as our finite-state compiler, facilitating the development process.
4. **Evaluation**: Our research involved comprehensive evaluation of the morphological analyzer using a gold-standard test data set. This step ensured that the artifact met its intended goals and provided reliable results.
5. **Conclusion and Contribution to Knowledge**: Our research has made a significant contribution to the field of NLP by creating a Ge'ez morphological analyzer and generator for all Ge'ez verbs. Additionally, you generated a gold-standard test data set that can be valuable for evaluating other Ge'ez language morphological analyzers

6. **Alignment with Research Questions:** Our research methodology aligned well with our research questions, demonstrating how we followed the design-and-creation research approach to address the challenges associated with Ge'ez language processing.

7.6 Chapter Summary

This chapter has presented the evaluation of the Ge’ez morphological analyzer using the gold-standard dataset and outlines the process of creating this dataset. The researcher enlisted the expertise of Ge’ez language specialists for the collection and manual analysis of Ge’ez verbs, drawn from the Ge’ez New Testament Bible and the Ge’ez prayer book 'ውዳሴ ማርያም' - wudase maryam. The accuracy of the Ge’ez morphological analyzer was assessed against this gold-standard data.

Current studies in Ge’ez language morphology have reported accuracies of less than 75%. In contrast, our Ge’ez morphological analyzer achieved an outstanding accuracy of 97.29% for the analyzed verbs, along with a precision of 80.24%. Furthermore, our Ge’ez morphological analyzer encompasses all Ge’ez verb categories and not only provides analysis but also generates verbs.

This chapter also evaluated the application of the research methodology employed in this study. The research questions were addressed as follows:

1. Chapter 2 tackled the research sub-question concerning the classification of Ge’ez verbs by justifying the selection of the washära Ge’ez verb classification.

2. Chapter 5 addressed the research sub-question by elucidating how Finite State Transducers (FST) are used to address the non-concatenative morphology of Ge’ez.

3. Chapter 6 addressed the research sub-questions by designing and developing a finite-state-based morphological analyzer for Ge’ez verbs using an adaptive development method, emphasizing modularization, unit testing, composition, system testing, and iteration.

4. Chapter 7 addressed the research sub-questions by creating a gold-standard test dataset and organizing a Ge’ez lexicon for the development of the Ge’ez analyzer.

The subsequent chapter will present the research conclusions and provide recommendations for future work.

Chapter 8 – Conclusions and Recommendations

8.1 Conclusions

The primary objective of this study was to develop a morphological analyzer for Ge’ez verbs. To achieve this, we identified the morphological properties of the eight Ge’ez head verbs and thoroughly studied their inflections. We explored two approaches to morphological analysis: rule-based and data-driven. Given the highly inflectional nature of the Ge’ez language, we opted for a rule-based approach to morphological analysis, as it is well-suited for such complex languages. Ge’ez, being a Semitic language with extensive inflection, can be efficiently represented using rule-based methods.

In designing and implementing the Ge’ez verb morphological analyzer, we opted to employ finite-state tools and techniques. This choice was primarily driven by the advantages of bidirectionality (allowing for both analysis and generation) and the simplicity in representing morphological rules, which align well with the complex nature of the Ge’ez language.

Ge’ez verbs are categorized as head and troops, with head verbs capable of representing others in their category, while troops follow the inflection and derivation patterns of head verbs. Therefore, our morphological analyzer was designed and developed for all eight head verbs. The process of Ge’ez verb formation involves roots, intercalation, morphophonological alternations, and suffixation to produce the surface form of the verb. We utilized CV-templates, intercalations, alternation rules, and affixations within the finite-state framework to develop the morphological analyzer.

To ensure the accuracy and completeness of the rules, we consulted with Ge’ez language experts who helped identify Ge’ez verb formation rules, including morphophonological alternation rules. These rules were then implemented using finite-state tools in the development of the Ge’ez morphological analyzer. Additionally, a test data set was organized in collaboration with Ge’ez language experts to facilitate the development of the lexicon.

The resulting Ge’ez morphological analyzer is capable of both Ge’ez verb morphological analysis and generation. It was evaluated using manually extracted verbs from the Ge’ez Bible (specifically, the books of Matthew, Mark, Luke, and John) and the Ge’ez prayer book 'ውዳሴ ማርያም' - ‘*wudase maryam’*. These verbs were meticulously annotated with structural information. Subsequently, the annotated verbs were compared with the output of the analyzer.

The evaluation involved 1,365 manually annotated verbs, and our Ge’ez verb morphological analyzer achieved an impressive accuracy rate of 97.29% and a precision rate of 80.24%.

Previous research on Ge’ez morphological analyzers had explored both rule-based and data-driven approaches, with varying degrees of success. Desta (2010) used a rule-based approach for one of the Ge’ez head verb types, while Abate (2014) employed a data-driven approach for all Ge’ez verb types, excluding irregular verbs. However, these efforts yielded accuracy rates below 75%. In contrast, our Ge’ez morphological analyzer achieved an impressive accuracy of 97.29% for the analyzed verbs, along with a precision rate of 80.24%.

Notably, both previous researchers focused solely on the morphological analysis of Ge’ez verbs, omitting morphological generation. This study builds upon their work by extending the research in several significant ways:

- We applied the rule-based approach to develop a morphological analyzer for all Ge’ez verb categories.
- We leveraged finite-state tools and techniques for the development of the Ge’ez verbs morphological analyzer.
- Our Ge’ez morphological analyzer encompasses all verb types, including irregular verbs
- Our Ge’ez morphological analyzer performs both the analysis and generation of Ge’ez verbs

In addition to achieving its primary objectives, this research has contributed to the field by providing an annotated test dataset, which can be valuable for evaluating other Ge’ez morphological analyzers. Moreover, the potential for enhancing the Ge’ez morphological analyzer through the addition of new roots to the lexicon is a promising avenue for future research. Furthermore, the potential applications of this research extend to various domains:

- It can serve as a fundamental component in the development of a comprehensive morphological analyzer for the Ge’ez language.
- It can contribute to a range of natural language processing applications, including spell checking and machine translation, for Ge’ez.
- Ge’ez language learners can benefit significantly from this resource
- Availability as a valuable resource for researchers working with Ge’ez language data

As the Ge’ez language gains renewed interest, particularly in educational settings, this research serves as a crucial step toward enhancing the accessibility and understanding of this ancient language.

8.2 Recommendations

In general, this research contributes its part to the development of a full-fledged morphological analyzer for the Ge’ez language by focusing on the development of a rule-based morphological analyzer for Ge’ez verbs. The study demonstrates that for a highly inflected Semitic language like Ge’ez, a rule-based methodology offers a more accurate analysis of verbs.

One of the primary challenges encountered in this research was the scarcity of documented full inflectional forms for head verbs, along with irregularities stemming from the presence of specific letters like ኧ-I, እ-’I, ሀ-h, ኅ-’h, ሕ-H, ው-w, and ይ-y in verbs. This scarcity is primarily due to the oral transmission of Ge’ez language teachings, notably within the Ethiopian Orthodox Tewahido Church. To further enhance the accuracy of the morphological analyzer, refining alternation rules may be necessary.

This research provides a solid foundation for future endeavors in Ge’ez language processing and natural language applications. Potential extensions include:

- Developing a morphological analyzer that encompasses other parts of speech (POS) within the Ge’ez language.
- Creating a disambiguation module to refine the Ge’ez morphological analyzer's output, particularly in handling context-dependent analyses.
- Exploring the development of Ge’ez language natural language processing applications, such as machine translation, to unlock the wealth of knowledge encoded in Ge’ez texts.

This study has not only deepened my appreciation for the richness of the Ge’ez language but has also underscored the significance of traditional teachings in the Ethiopian Orthodox Tewahido Church. These teachings encompass not only spiritual knowledge but also worldly wisdom, including philosophy, law, and arithmetic. The development of a natural language application would facilitate access to this wealth of knowledge for interested individuals.

Therefore, the finite-state-based Ge’ez verb morphological analyzer represents a crucial component in the development of Ge’ez language natural language applications, ensuring the preservation and accessibility of this valuable linguistic and cultural heritage.

REFERENCES

Abate, M., & Assabie, Y. (2014). Development of Amharic morphological analyzer using memory-based learning. In Advances in natural language processing. (pp. 1–13). Springer.

Abate, Y. (2014). *Morphological analysis of Ge'ez verbs using memory based learning.* Addis Ababa University.

Adihana (Memhir), Z. (2015). *Meriho Sewasew Zelisan Ge'ez.* Akotet Printing Press.

Alpaydin, E. (2010). *Introduction to machine learning.* (2nd ed.). MIT Press.

Amsalu, S., & Demeke, G. A. (2006). *Nonconcatenative finite-state morphotactics of Amharic simple verbs.* Citeseer.

Amsalu, S., & Gibbon, D. (2005). Finite state morphology of Amharic. *Proceedings of Recent Advances in Natural Language Processing*. RANLP2005

Andualem, M. (2007). *Ge'ez verb classification in the three traditional schools of Qene.* Addis Ababa University.

Argaw, A. A., & Asker, L. (2007). An Amharic stemmer: Reducing words to their citation forms. *Proceedings of the 2007 workshop on computational approaches to semitic languages: Common issues and resources*. (pp. 104–110). Association for Computational Linguistics.

Bayu, T. (2002). *Automatic morphological analyzer for Amharic: An experiment employing unsupervised learning and auto-segmental analysis approaches.* (Master's thesis, Addis Ababa University.) http://etd.aau.edu.et/bitstream/handle/123456789/12037/Meaza%20Demissie.pdf?sequence=1&isAllowed=y

Beesley, K. R. (1996). Arabic finite-state morphological analysis and generation. *Proceedings of the 16th conference on Computational linguistics - Volume 1,* (pp. 89–94). Association for Computational Linguistics.

Beesley, K. R. (1998). Arabic morphology using only finite-state operations. *Proceedings of the Workshop on Computational Approaches to Semitic languages*. (pp. 50–57). Association for Computational Linguistics.

Beesley, K. R. (2004). Morphological analysis and generation: A first step in natural language processing. First Steps in language documentation for minority languages: Computational linguistic tools for morphology, lexicon and corpus compilation. *Proceedings of the SALTMIL Workshop at LREC.* (pp. 1–8). Retrieved from: http://www.lrec-conf.org/proceedings/lrec2004/ws/ws2.pdf

Beesley, K. R., & Karttunen, L. (2000). Finite-state non-concatenative morphotactics. *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*. (pp. 191–198). Association for Computational Linguistics.

Beesley, K. R., & Karttunen, L. (2003). *Finite-state morphology: Xerox tools and techniques.* CSLI.

Berhanu, L. (2006). *Yege'ez Memaria.* Signature Book Printing.

Book of Enoch. (2019). Retrieved from https://www.en.wikipedia.org/wiki/Book

Challiot, C. (2009). Traditional teaching in the Ethiopian Orthodox Church: Yesterday, today and tomorrow. *Proceedings of the 16th International Conference of Ethiopian Studies.* Chittick.

Clark, A., & Lappin, S. (2010). Unsupervised learning and grammar induction. In S. Lappin, A. Clark, & C. Fox (Eds.), *The handbook of computational linguistics and natural language processing.* (p. 57). Wiley.

Cohen-Sygal, Y., & Wintner, S. (2006). Finite-state registered automata for non-concatenative morphology. *Computational Linguistics, 32*(1), 49–82.

Creswell, J. W. (2003). *Research design.* SAGE.

Dale, R. (2010). Classical approaches to natural language processing. In N. Indurkhya, & F. J. Damerau (Eds.), *Handbook of natural language processing,* (2nd ed.). (pp. 3–7). CRC Press.

Desta, B. W. (2010). *Design and implementation of automatic morphological analyzer for Ge'ez verbs*. (Doctoral dissertation, Addis Ababa University). http://thesisbank.jhia.ac.ke/id/eprint/5981

Dillmann, A., Bezold, C., & Crichton, J. A. (2003). *Ethiopic grammar.* Wipf and Stock.

Faaß G., Heid, U., & Schmid, H. (2010). Design and application of a gold standard for morphological analysis: SMOR as an example of morphological evaluation. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), Valletta, Malta*. European Language Resources Association (ELRA).

Foma*.* (2011). *Regular expression operators.* Retrieved from: https://fomafst.github.io/regexreference.html

Gasser, M. (2011). HornMorpho: a system for morphological processing of Amharic, Oromo, and Tigrinya. *Conference on Human Language Technology for Development,* Alexandria, Egypt. Retrieved from https://cgi.luddy.indiana.edu/~gasser/Papers/hltd11.pdf

Hulden, M. (2009). Foma: a finite-state compiler and library. *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Demonstrations Session*. (pp. 29–32). Association for Computational Linguistics.

Hulden, M. (2011a). *Morphological analysis tutorial. A self-contained tutorial for building morphological analyzers.* Retrieved from: https://fomafst.github.io/morphtut.html

Hulden, M. (2011b). *The foma FST compiler.* Retrieved from: https://github.com/mhulden/foma/blob/master/foma/docs/simpleintro.md

Jurafsky, D., & Martin, J. H. (2008). *Speech and language processing. An introduction to speech recognition, natural language processing, and computational linguistics.* (2nd ed.). Prentice-Hall.

Kataja, L., & Koskenniemi, K. (1988). Finite-state description of Semitic morphology: A case study of Ancient Akkadian. *Proceedings of the 12th conference on*

Computational Linguistics - Volume 1. (pp. 313–315). Association for Computational Linguistics.

Kay, M. (1987, April). Nonconcatenative finite-state morphology. In *Proceedings of the Third Conference of the European Chapter of the Association for Computational Linguistics, Copenhagen, Denmark,* April 1 - 3, 1987 (pp. 2–10). Association for Computational Linguistics.

Kazakov, D., & Manandhar, S. (2001). Unsupervised learning of word segmentation rules with genetic algorithms and inductive logic programming. *Machine Learning, 43*(1–2), 121–162.

Keleb (Memhir), D. (2010). *Tinsae Ge'ez.* Mahbere Kidusan Printing Press.

Kifle (Aleka), K. (1956). Metsihafe Sewasew wegis wemezgebe kalat. Artistic Printing Press.

Koskenniemi, K. (1983). *Two-level morphology. A general computational model for word-form recognition and production*. University of Helsinki.

Lambdin, T. O. (1978). *Introduction to Classical Ethiopic (Ge'ez).* Scholars Press.

Liddy, E. D. 2001. Natural language processing. In *Encyclopedia of library and information science*. (2nd ed.). Marcel Decker.

Martin, H. (2002). *Understanding morphology.* Oxford University Press Inc.

McCarthy, J. J. (1981). A prosodic theory of nonconcatenative morphology. *Linguistic Inquiry, 12*(3), 373–418.

Mercer, S. A. B. (1961). *Ethiopic grammar: with chrestomathy and glossary*. (2nd ed.). Ungar.

Mulugeta, W., & Gasser, M. (2012, May). Learning morphological rules for Amharic verbs using inductive logic programming. In *Proceedings of the Workshop on Language Technology for Normalisation of Less-Resourced Languages* (SaLTMiL 8 - AfLaT 2012). LREC.

Oates, B. J. (2005). *Researching information systems and computing*. SAGE.

Metsehefe Sewasew. (1985). Tensae Printing Press.

Sergew H. & Ṗāwlos Ṗāwlos. (1997, 1970). The church of Ethiopia : a panorama of history and spiritual life (Second printing). A Publication of the Ethiopian Orthodox Church.

Tachbelie, M. Y. (2010). *Morphology-based language modelling for Amharic*. (Doctoral dissertation: Hamburg University). https://ediss.sub.uni-hamburg.de/bitstream/ediss/3816/1/TachbelieDissertation.pdf

Trost, H. (2003). Computational morphology. In R. Mitkov (editor), *The Oxford handbook of computational linguistics*. (pp. 25–47). Oxford University Press.

Vaishnavi, V., & Kuechler, W. (2004). Design research in information systems. Retrieved from www.isworld.org/Researchdesign/drisISworld.htm

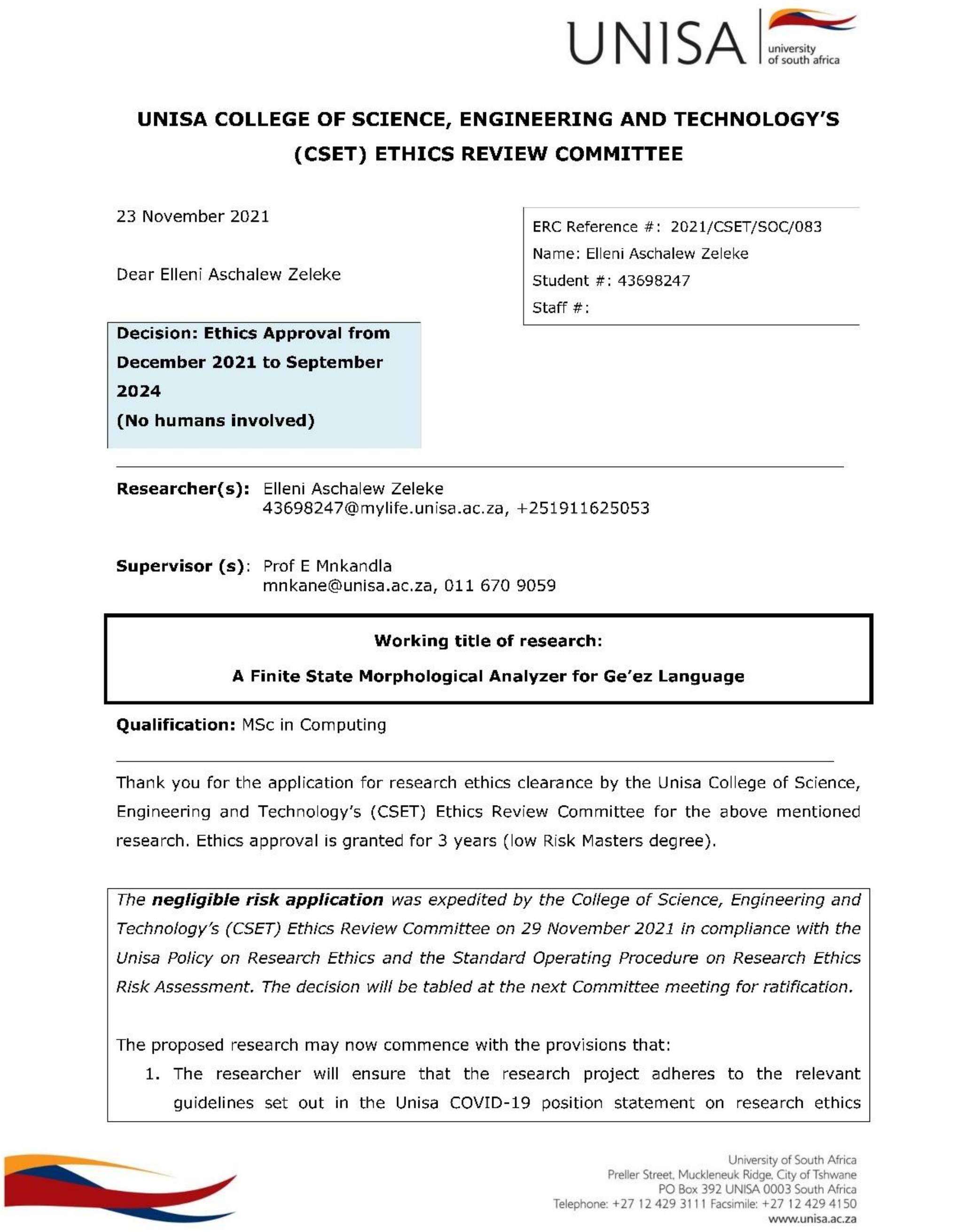
Wikipedia: AT & T FSM and Library. (2016). *Finite-state machine.* Retrieved from: https://en.wikipedia.org/wiki/AT% 26T FSM Library

Yacob, D. & Firdyiwek, Y (1997). *The system for Ethiopic representation in ASCII (SERA) 1997 standard.* Retrieved from :**Error! Hyperlink reference not valid.**https://www.researchgate.net/publication/2682324_The_System_for_Ethiopic_Representation_in_ASCII

Yona, S., & Wintner, S. (2008). A finite-state morphological grammar of Hebrew. *Natural Language Engineering, 14*(02), 173–190.

APPENDICES

I. Ethical clearance



UNISA university of south africa

UNISA COLLEGE OF SCIENCE, ENGINEERING AND TECHNOLOGY'S (CSET) ETHICS REVIEW COMMITTEE

23 November 2021

Dear Elleni Aschalew Zeleke

ERC Reference #: 2021/CSET/SOC/083
Name: Elleni Aschalew Zeleke
Student #: 43698247
Staff #:

Decision: Ethics Approval from December 2021 to September 2024
(No humans involved)

Researcher(s): Elleni Aschalew Zeleke
43698247@mylife.unisa.ac.za, +251911625053

Supervisor (s): Prof E Mnkandla
mnkane@unisa.ac.za, 011 670 9059

Working title of research:

A Finite State Morphological Analyzer for Ge'ez Language

Qualification: MSc in Computing

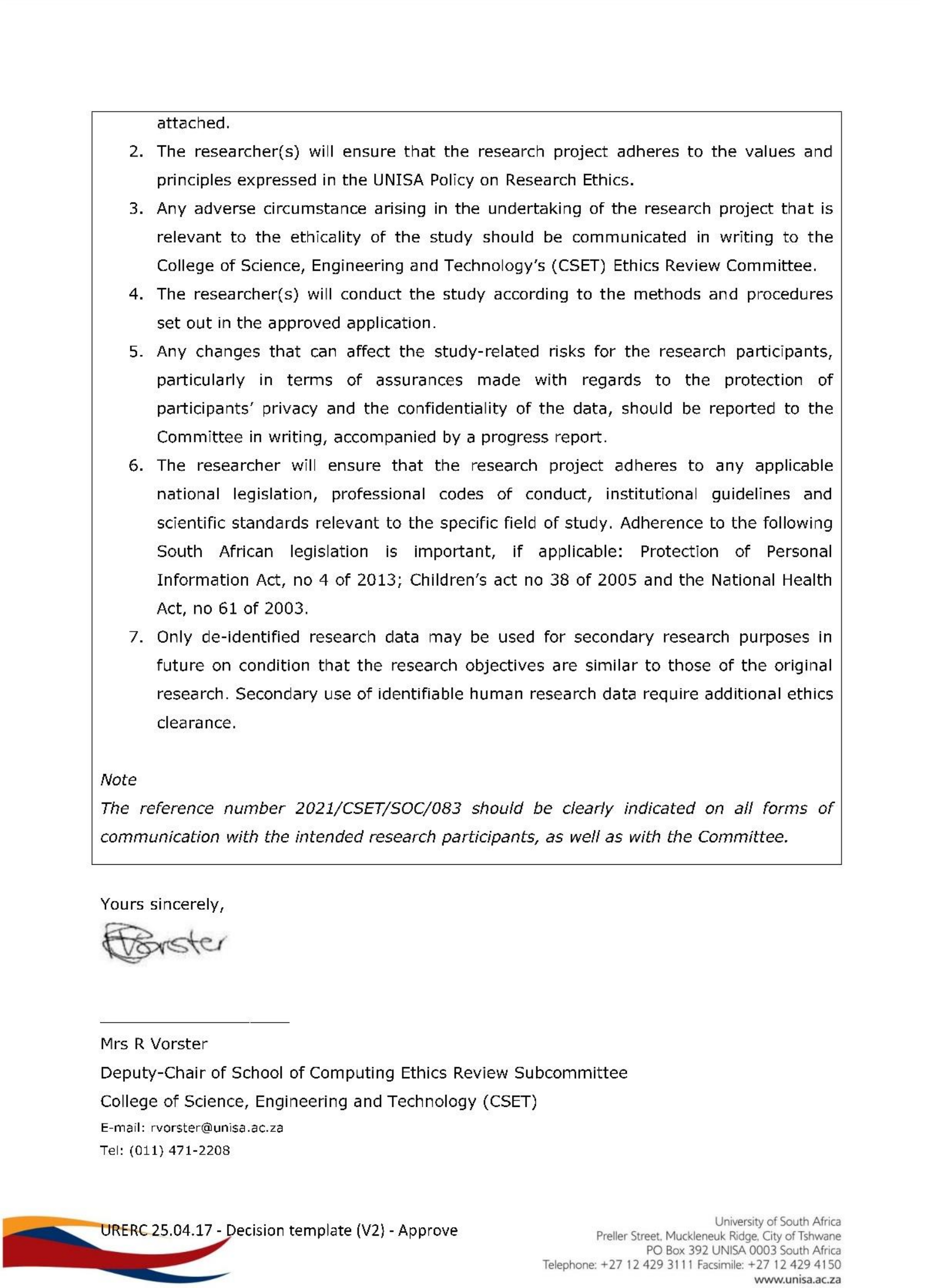
Thank you for the application for research ethics clearance by the Unisa College of Science, Engineering and Technology's (CSET) Ethics Review Committee for the above mentioned research. Ethics approval is granted for 3 years (low Risk Masters degree).

*The **negligible risk application** was expedited by the College of Science, Engineering and Technology's (CSET) Ethics Review Committee on 29 November 2021 in compliance with the Unisa Policy on Research Ethics and the Standard Operating Procedure on Research Ethics Risk Assessment. The decision will be tabled at the next Committee meeting for ratification.*

The proposed research may now commence with the provisions that:

1. The researcher will ensure that the research project adheres to the relevant guidelines set out in the Unisa COVID-19 position statement on research ethics

University of South Africa
Preller Street, Muckleneuk Ridge, City of Tshwane
PO Box 392 UNISA 0003 South Africa
Telephone: +27 12 429 3111 Facsimile: +27 12 429 4150
www.unisa.ac.za



attached.

2. The researcher(s) will ensure that the research project adheres to the values and principles expressed in the UNISA Policy on Research Ethics.
3. Any adverse circumstance arising in the undertaking of the research project that is relevant to the ethicality of the study should be communicated in writing to the College of Science, Engineering and Technology's (CSET) Ethics Review Committee.
4. The researcher(s) will conduct the study according to the methods and procedures set out in the approved application.
5. Any changes that can affect the study-related risks for the research participants, particularly in terms of assurances made with regards to the protection of participants' privacy and the confidentiality of the data, should be reported to the Committee in writing, accompanied by a progress report.
6. The researcher will ensure that the research project adheres to any applicable national legislation, professional codes of conduct, institutional guidelines and scientific standards relevant to the specific field of study. Adherence to the following South African legislation is important, if applicable: Protection of Personal Information Act, no 4 of 2013; Children's act no 38 of 2005 and the National Health Act, no 61 of 2003.
7. Only de-identified research data may be used for secondary research purposes in future on condition that the research objectives are similar to those of the original research. Secondary use of identifiable human research data require additional ethics clearance.

Note

The reference number 2021/CSET/SOC/083 should be clearly indicated on all forms of communication with the intended research participants, as well as with the Committee.

Yours sincerely,

RVorster

Mrs R Vorster
Deputy-Chair of School of Computing Ethics Review Subcommittee
College of Science, Engineering and Technology (CSET)
E-mail: rvorster@unisa.ac.za
Tel: (011) 471-2208

Dr T Masombuka
COD: Department Computer Science
College of Science Engineering and
Technology (CSET)
E-mail: masomkt@unisa.ac.za
Tel: (011) 670 9123

pp

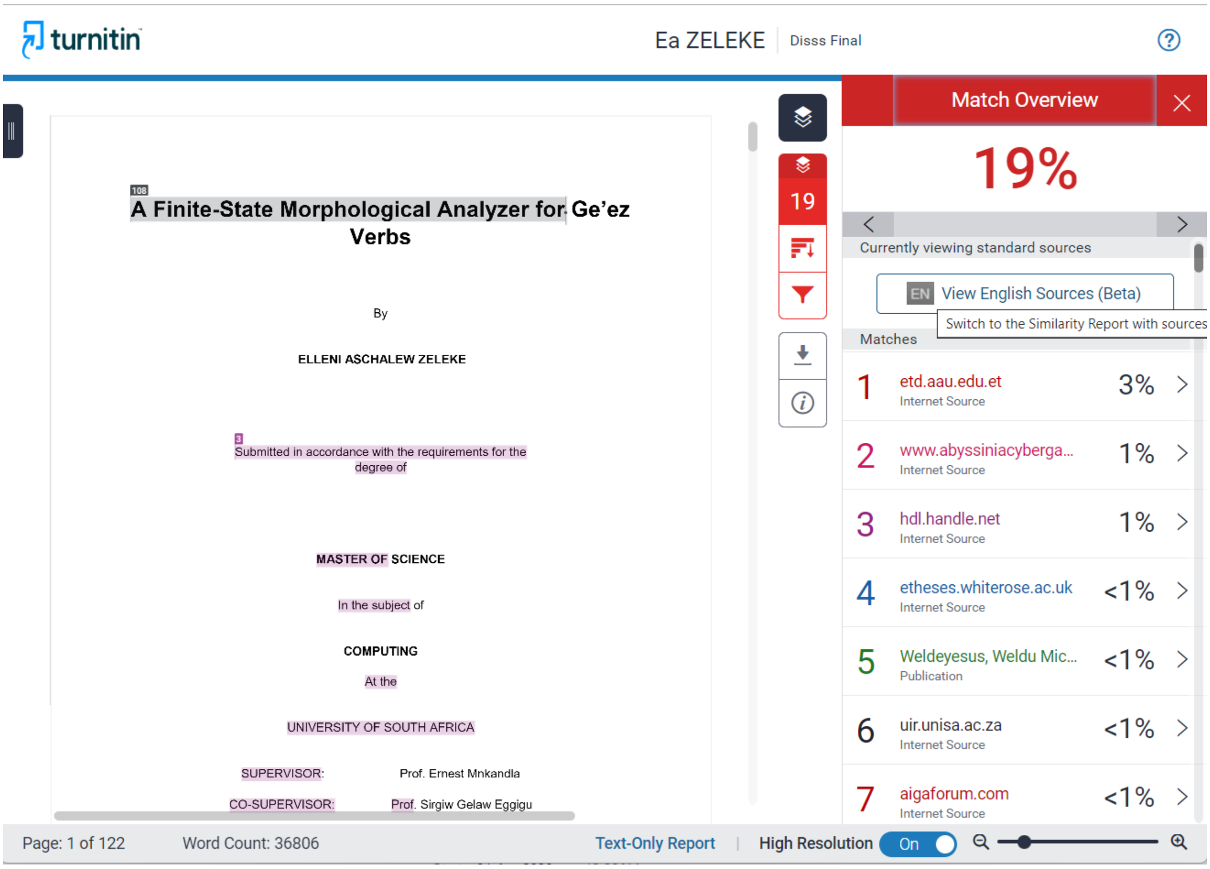
Prof. B Mamba
Executive Dean
College of Science Engineering and
Technology (CSET)
E-mail: mambabb@unisa.ac.za
Tel: (011) 670 9230

II. Lexical and Foma files

The Ge'ez Morphological analyzer is built using Foma, Version 0.9.18alpha (svn r241). The analyzer consists of 10 lexc files coressponding to eight verb types, bEle verb and Irregular verbs and a foma script file. The lexical files describes verb formation rules and contains a list of root verbs in that particular verb type. The foma script file describes the intercalations of vowels into consonants and morphophonological rules. The Geez Finite-State Transducer files are:

- VerbType1 - qetele verb type
- VerbType2 - qedese verb type
- VerbType3 - gebre verb type
- VerbType4 - AImere verb type
- VerbType5 - bareke verb type
- VerbType6 - 'sEme verb type
- VerbType7 - bhle verb type
- VerbType8 - qome verb type
- bEle - bEle verb type
- irregular - irregular verb type
- geez.foma - Foma file

III. Turnitin Report



A Finite-State Morphological Analyzer for Ge’ez Verbs

By

ELLENI ASCHALEW ZELEKE

Submitted in accordance with the requirements for the degree of

MASTER OF SCIENCE

In the subject of

COMPUTING

At the

UNIVERSITY OF SOUTH AFRICA

SUPERVISOR: Prof. Ernest Mnkandla

CO-SUPERVISOR: Prof. Sirgiw Gelaw Eggigu

19%

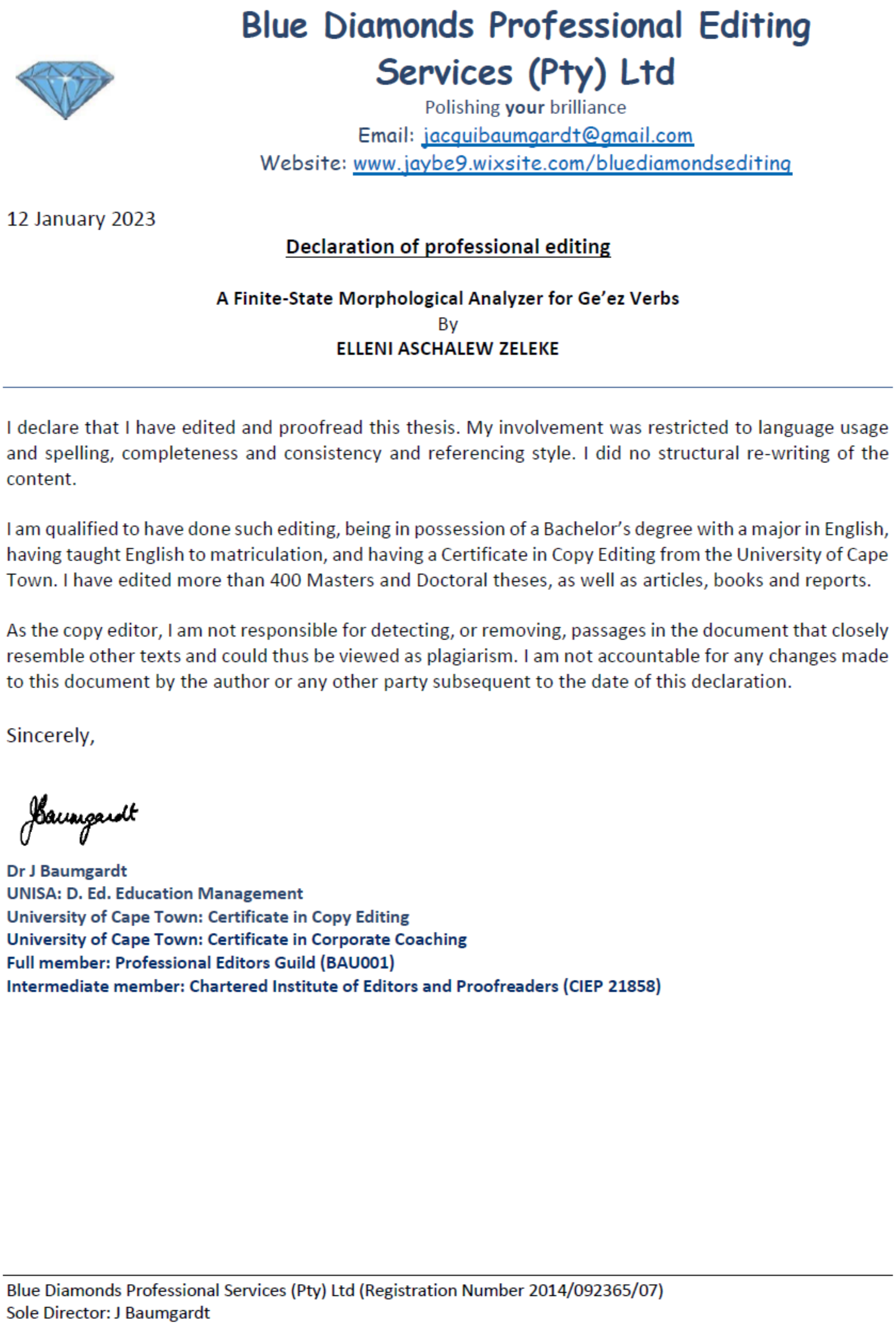
Currently viewing standard sources

Matches

1	etd.aau.edu.et Internet Source	3%
2	www.abyssiniacyberga... Internet Source	1%
3	hdl.handle.net Internet Source	1%
4	etheses.whiterose.ac.uk Internet Source	<1%
5	Weldeyesus, Weldu Mic... Publication	<1%
6	uir.unisa.ac.za Internet Source	<1%
7	aigaforum.com Internet Source	<1%

Page: 1 of 122 Word Count: 36806

IV. Confirmation of Professional Editing



Blue Diamonds Professional Editing Services (Pty) Ltd

Polishing **your** brilliance

Email: jacquibaumgardt@gmail.com

Website: www.jaybe9.wixsite.com/bluediamondsediting

12 January 2023

Declaration of professional editing

A Finite-State Morphological Analyzer for Ge'ez Verbs

By

ELLENI ASCHALEW ZELEKE

I declare that I have edited and proofread this thesis. My involvement was restricted to language usage and spelling, completeness and consistency and referencing style. I did no structural re-writing of the content.

I am qualified to have done such editing, being in possession of a Bachelor's degree with a major in English, having taught English to matriculation, and having a Certificate in Copy Editing from the University of Cape Town. I have edited more than 400 Masters and Doctoral theses, as well as articles, books and reports.

As the copy editor, I am not responsible for detecting, or removing, passages in the document that closely resemble other texts and could thus be viewed as plagiarism. I am not accountable for any changes made to this document by the author or any other party subsequent to the date of this declaration.

Sincerely,

JBaumgardt

Dr J Baumgardt
UNISA: D. Ed. Education Management
University of Cape Town: Certificate in Copy Editing
University of Cape Town: Certificate in Corporate Coaching
Full member: Professional Editors Guild (BAU001)
Intermediate member: Chartered Institute of Editors and Proofreaders (CIEP 21858)

Blue Diamonds Professional Services (Pty) Ltd (Registration Number 2014/092365/07)
Sole Director: J Baumgardt