Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA 2022), pages 108–113
Language Resources and Evaluation Conference (LREC 2022), Marseille, 25 June 2022

Handling Stress in Finite-State Morphological Analyzers for Ancient Greek and Ancient Hebrew

Daniel G. Swanson, Francis M. Tyers
Department of Linguistics,
Indiana University,
{dangswan,ftyers}@iu.edu

Abstract

Modeling stress placement has historically been a challenge for computational morphological analysis, especially in finite-state systems because lexically conditioned stress cannot be modeled using only rewrite rules on the phonological form of a word. However, these phenomena can be modeled fairly easily if the lexicon's internal representation is allowed to contain more information than the pure phonological form. In this paper we describe the stress systems of Ancient Greek and Ancient Hebrew and we present two prototype finite-state morphological analyzers, one for each language, which successfully implement these stress systems by inserting a small number of control characters into the phonological form, thus conclusively refuting the claim that finite-state systems are not powerful enough to model such stress systems and arguing in favor of the continued relevance of finite-state systems as an appropriate tool for modeling the morphology of historical languages.

Keywords: Greek, Hebrew, finite-state

1. Introduction

Morphological analysis, the identification of lexical and morphological information for a given word form, is an important step in the study of texts, most basically for the tasks of searching and indexing, particularly in more inflected languages such as Greek and Hebrew.

Computational morphological analysis, moreover, has proved itself useful in searching and indexing (Crane, 1991), pedagogy (Packard, 1973), and translation (Forcada et al., 2011), among other tasks.

One of the most common ways to implement a morphological analyser has been to use Finite-State Transducers (FSTs), which specify a mapping between two sets of strings (in this case, surface form and morphological analysis) in a compact and efficient form.

Modeling stress, however, has historically been a challenge for FSTs, to the point of being called impossible to implement as a sequence of local rewrite rules (Smith, 2016). In this paper, we demonstrate two successful approaches to stress: a full stress-placement system for Ancient Greek and a simpler stress-shifting system for Ancient Hebrew.

Section 2 discusses prior work and the capacities of finite-state systems, Section 3 describes the relevant details of the Greek and Hebrew stress systems, Section 4 describes the implementation, Section 5 provides a quantitative evaluation of the current state of development, and Section 6 concludes.

2. Finite-State Morphology

Several morphological analyzers for Ancient Greek already exist, including the mostly finite-state Morpheus (Smith, 2016), though this system required an ad-hoc extension due to difficulties in formulating the Greek stress system as a sequence of rewrite rules.

We are not aware of any prior analyzers for Ancient Hebrew, though for Modern Hebrew, which is morphologically quite similar, there are several, such as HAMSAH (Yona and Wintner, 2008).

In both of these cases, it has been concluded that finite-state transducers are not up to the task of representing all the relevant morphological alternations in a maintainable way (Smith, 2016; Wintner, 2008). However, this is due to the assumption that the only available operations when building FSTs are appending suffixes and applying rewrite rules.

In fact, there are at least three other tools available to a grammar writer which, combined, make modeling complex morphological phenomena possible and make maintaining dictionaries as they expand much easier.

The first tool is interlacing lexical entries, which is supported by the lexicon compiler Lexd (Swanson and Howell, 2021). From the perspective of the grammar writer, they make lists of affixes and where they go in relation to the root and the compiler internally expands this into a sequence of append operations, making Hebrew's templatic morphology far easier to model. An example of how this can be used is given in Figure 1.

The second tool is constraints (Karttunen, 1991). These can be written in a format almost identical to rewrite rules, but they apply in parallel so the developer does not need to carefully sequence the operations. An example of such constraints is given in Figure 2.

The final tool is intersection. A lexicon compiler can be used to generate an FST containing all forms allowed by a language's phonotactics. This can then be composed or intersected with the analyzer, leaving only valid forms.

All of these tools have compilers available which allow the rules to be written in formats which closely resemble how the processes they model would be de-

```
LEXICON VerbRoot(3)
’ m r[1’]
’ s p[1’]
b ’ {sh}[reg]
b d l[reg]
b h l[reg]

ALIAS VerbRoot C

PATTERN Pa’al
C(1) C(2) [:{˜o}{*?}] C(3)[reg]
C(1) C(2) [:{˜a}{*?}] C(3)[1’]
```

Figure 1: A fragment of the lexicon and rules for generating Hebrew verbal stems. The VerbRoot lexicon contains the tri-consonantal verb roots, which each consonant in a separate column. Each root is also tagged with features that affect verb stem formation. Here the tags are reg for “regular” and 1’ for roots where the first consonant is the glottal stop א. The ALIAS line specifies an alternate name for the VerbRoot lexicon so that the stem patterns can be written more concisely. Finally, the last two lines specify how to insert vowels between the three consonants of the root to form the Pa‘al (active) stem.

```
"schwa deletes before determiner"
@:0 <=> _ {h}: ;

"determiner before gutturals"
a:á <=> {h}: _ [ ’ | {’} ] ;

"{h} deletes after vowel"
{h}:0 <=> Vowel: _ ;
```

Figure 2: The phonological rules controlling the realization of the Hebrew definite article. These can be read like rewrite rules (the second, for instance, reads “a becomes á if and only if it is preceded by some realization of {h} and followed by either ’ (א) or {’} (ע)”). However, they are applied simultaneously, and thus the order they are written in has no effect.

scribed in theoretical linguistic analyses, which thus gives finite-state systems the advantage that the rules used to compile them are, in themselves, a form of linguistic documentation. Furthermore, since these rules have to be executed by a computer, they may well be more precise and complete than a purely linguistic description of the same phenomena.

3. Stress in Greek and Hebrew

In this section, we will summarize the relevant facts about stress and how it is marked in Greek and Hebrew.

3.1. Greek

Ancient Greek texts employ three accent marks: acute (ά), circumflex (ᾶ), and grave (ὰ).

The grave accent replaces the acute when it occurs on the final syllable in certain contexts. While handling this aspect of the Greek stress system within a single FST is possible, it results in a single entry spanning arbitrarily many words, which wouldn’t be a problem when analyzing running text, but would cause the analyzer to sometimes fail on single forms. Thus our analyzer simply accepts both forms.

When analyzing, these alternate forms never change the identification of the form and when generating, the selection of the surface form can be handled in Apertium using a second FST which is not composed and which operates on surface forms across word boundaries.

The acute and circumflex are subject to the following restrictions:

1. The circumflex may only appear on long vowels or diphthongs.
2. The circumflex may occur on the final syllable or on the penultimate syllable if the final is short.

   Thus σκηνῆς (long-long, final stress) and σωτῆρα (long-long-short, penultimate stress) are possible, but *σκῆνης (long-long, initial stress) is not.
3. The acute may appear on either of the last two syllables or the last three if the final is short.

   Thus in the five syllables of παιδευομενος, *παίδευομενος and *παιδεύομενος are impossible, but παιδευομένος and παιδευομενός are allowed, and since ο is short, so is παιδευόμενος.
4. If the accent falls on a long penultimate syllable and the final syllable is short, the accent must be a circumflex.

   So σωτήρων with long final syllable, but σωτῆρα with short.

In general, nouns have a lexically determined accented syllable and the accent will be placed as close to that syllable as possible. For example, forms of ἄνθρωπος “human” will have the stress on the initial syllable (αν) whenever the final syllable is short and on the second syllable (θρω) otherwise, such as in the genitive ἀνθρώπου. On the other hand, θεός “god”, will always have the stress on the final syllable.

Verbs, on the other hand, will place the accent on the earliest permissible syllable, so, according to the rules, παιδευομεθα “I am being taught” can have an acute accent on ο, ε, or α, so it will have it on the earliest one, giving παιδευόμεθα. Meanwhile, παιδευω “I am teaching” can have an acute on ευ or ω or a circumflex on ω, and selecting the earliest one gives παιδεύω.

Additionally, if certain vowels are adjacent, they will merge into a long vowel or diphthong. The stress, however, is placed as if they weren't merged except that an acute accent on the first vowel will become a circumflex. Thus τιμῶμαι "I am honored" has penultimate stress even though the final αι counts as short in this context because it is underlyingly τιμάομαι with antepenultimate stress (van Emde Boas et al., 2019).

3.2. Hebrew

Unlike Greek, Hebrew orthography in general does not mark the location of stress except in religious texts where diacritics called "cantillation" or "trope" are placed on stressed syllables indicating how the word is to be sung. Additionally, the different cantillation marks indicate how closely connected a word is to its neighbors, which gives some indication of the syntax (Gesenius and Kautzsch, 2006).

As a result, if identifying morphological forms is the only goal, then tracking stress is not strictly necessary. However, explicitly modeling stress makes other rules more parsimonious and allows the rules to more effectively serve as a form of documentation of the language's morphophonology.

Stress usually falls on the final syllable of a word, though some nouns have initial stress. Additionally, there are two verbal forms (one of which, the vav-consecutive construction, is the most common form in biblical narrative) which move the stress to penultimate syllable of the stem. This shift changes the final vowel and may delete the final syllable entirely, depending on the final consonant (Gesenius and Kautzsch, 2006).

4. Implementing Stress

In this section, we describe the structure of our analyzers. Both analyzers were created in the Apertium machine translation platform (Forcada et al., 2011; Khanna et al., 2021) using the lexicon compiler Lexd (Swanson and Howell, 2021) with two-level phonology (Twol) (Koskenniemi, 1983; Lindén et al., 2009) and are freely available under the GPLv3 open-source license[1].

4.1. Greek

The Greek transducer is the result of composing a lexicon transducer with five sets of rules. The process is shown in Table 1.

4.1.1. Morphophonology

The first step is the morphophonology, which takes a sequence of morphemes from the lexicon, such as φυ{ʹ}λακ+σ, and adjusts vowels and consonant clusters as required by Greek phonology and phonotactics (in this case giving φυ{ʹ}λαξ). The symbol {ʹ} indicates the lexical stress location.

```
Dental = T Δ Θ
         τ δ ϑ ;
Cx:0 <=> _ Mod:* .#. ;
         _ Mod:* [:σ|:ς|σ:|ς:] ;
         where Cx in Dental ;
```

This rule, for example, deletes dental stops (τ, δ) or fricatives (ϑ) when they occur at the end of a word (.#.) or before sigma. Mod:* indicates that the rule should still apply if there are any control characters between the two consonants.

4.1.2. Orthographic Transformations

The second step ensures that all initial vowels have breathing marks and that all final sigmas are ς rather than σ, since this turned out to be significantly easier to write than combining it with the first step.

```
σ:ς <=> _ .#. ;
```

This is the rule that ensures final sigmas are always ς.

4.1.3. Syllable Boundaries

The third step inserts a marker ({.}) after each syllable nucleus and also marks final αι and οι, since they are treated as short vowels rather than diphthongs for the purposes of stress placement if they occur word-finally.

```
0:%{.%} <=>
    Vowel: VowelMod* _
    [Consonant|.#.|NonSecondDiph] ;
```

This rule says to insert the syllable marker after a vowel, possibly accompanied by some control characters, if it is followed by a consonant, the end of the word (.#.), or a vowel which cannot be the second letter of a diphthong.

4.1.4. Stress Placement

Next the fourth step consists of a Lexd file which lists every possible combination of long and short vowels and lexical accent marks in the last three syllables of a word and which vowel should receive the stress mark.

```
Prefix LongVowel(3) Acute BD
FinalShortSyllable

LEXICON LongVowel(4)
αι:αι  αι:αί  αι:αῖ  αι:αὶ
ει:ει  ει:εί  ει:εῖ  ει:εὶ
...

PATTERN FinalShortSyllable
CC ShortVowel(1) BD CC
```

This rule matches a word consisting of arbitrarily many initial syllables (Prefix), a long vowel or diphthong (LongVowel), a stress marker (Acute), and a short syllable with no stress marker (FinalShortSyllable). The (3) indicates that the penultimate vowel should be modified based on the third column of the LongVowel lexicon (the one with circumflexes).

[1] The code can be found on Github at https://github.com/apertium/apertium-grc and https://github.com/apertium/apertium-hbo

4.1.5. Vowel Contraction

Finally, if there are any vowels separated by the contraction sign ({+}), they are merged, adjusting the accents if necessary.

```
[ ά %{%+%} [ ε ι | ῃ | α ι | ᾳ ] ] -> ᾷ
```

This rule specifies that if an alpha with an acute accent (ά) is contracted with any of the four listed diphthongs, the acute becomes a circumflex and the resulting vowel is an alpha with an iota subscript (ᾷ).

4.2. Hebrew

The Hebrew FST is likewise a lexicon followed by a cascade of five sets of rules. All steps except the final one are currently in a Latin-alphabet transliteration because rules operating on combining diacritics being hard to read and modify. However, since this issue is primarily a matter of text editor support, it should be possible to convert the process to Hebrew script. The process is shown in Table 2.

4.2.1. Morphophonology

The first step is applying morphophonological rules to the forms generated by the lexicon.

```
"feminine plural drop -áh: á"
á:0 <=> _ h: %>: w o t ;
"feminine plural drop -áh: h"
h:0 <=> á: _ %>: w o t ;
```

These two rules together indicate that when a noun ending in áh (ה◌ָ is followed by the feminine plural suffix wot (וֹת) then the áh should be deleted.

4.2.2. Stress Selection

In the lexicon, stress markers are placed both on roots and on suffixes, so the next step is to remove spurious ones leaving a single stress position.

```
Stress = %{%*%} %{%*%?%} ;
%{%*%?%}:0 <=> _ :* Stress: ;
```

This rule any stress markers for which there is another stress marker later in the word.

4.2.3. Stress Movement

In the third step, if there is a prefix containing a symbol marking that stress moves earlier in the word, the stress marker is inserted in the preceding syllable and the original one is replaced by a marker that reduction should occur if possible.

```
%{%*%}:%{%-%*%} <=> %{%$<$%*%}: :* _ ;
```

This rule replaces a stress marker ({⋆}) with a former-stress marker ({−⋆}) if there is a preceding move-stress marker ({<⋆}).

4.2.4. Stress Reduction

The fourth step applies morphophonological rules to adjust certain vowels based on the position of stress and reduction marks.

```
h:0 <=> %{%-%*%}: _ ;
```

This rule deletes h (ה) if it is immediately preceded by a former-stress marker.

4.2.5. Transliteration

Finally, the resulting form is transliterated into Hebrew script.

```
CL:CH <=> _ ( Vowel: ) .#. ;
      where CL in ( k m n p c )
            CH in ( ך ם ן ף ץ )
      matched ;
```

This rule ensures that consonants which have a distinct final form are transliterated to their final form if they are the last consonant in a word.

5. Evaluation

Development of these analyzers was originally begun as part of an experiment in processing Biblical texts in the Apertium framework and, as a result, both are currently focused on the Biblical varieties of the languages. Incorporating multiple language varieties is, however, fairly straightforward and is often done in other Apertium analyzers. We have not yet attempted such an expansion and so report results on Biblical texts only.

The Greek FST provides analyses for nearly all words in the New Testament, as shown in Table 3. The development of the Hebrew FST, on the other hand, is not as far along, and it only provides analyses for a bit less than two thirds of the book of Genesis.

Both FSTs currently overgenerate somewhat. In Greek this affects about 8% of words and is largely due to partially irregular verbs not being properly labeled in the lexicon, resulting in them having both the correct irregular form as well as an incorrect regularized form in the FST.

In Hebrew, on the other hand, various morphological processes insert different vowels in different contexts, and some of these realizations have not yet been properly constrained. This primarily affects any form involving a possessive or object pronoun. In addition, work on nominal morphology is rather incomplete, which limits the usefulness of the Hebrew FST for generating anything besides the most common verb forms.

6. Conclusion and Future Work

This paper has presented the implementation of stress in morphological analyzers for Ancient Greek and Ancient Hebrew.

In addition to the issues mentioned in Section 5, there remains a significant amount of expansion to be done

Step	Output	Output
	τιμαω<v><ind><actv><impf><pres><p1><sg>	φυλαξ<n><m><sg><nom>
Lexicon	{’}τιμα{’?}{+}o{+long}	{’}φυ{’}λαϰ{g+}{’?}ς
Morphophonology	{’}τιμα{+}ω	{’}φυ{’}λαξ
Orth. Transforms	{’}τιμα{+}ω	{’}φυ{’}λαξ
Syllable Boundaries	{’}τι{.}μα{.}{+}ω{.}	{’}φυ{’}{.}λα{.}ξ
Stress Placement	τιμά{+}ω	φύλαξ
Vowel Contraction	τιμῶ	φύλαξ

Table 1: The steps involved in generating two surface forms in the Greek transducer. Analysis follows the same process but in reverse. Since each layer is a finite-state transformation, the entire sequence can be composed to produce a single transducer, so the intermediate states are not actually present at runtime. The tags in angle brackets on the first line indicate “verb, indicative, active, imperfective, present, 1st person, singular” and “noun, masculine, singular, nominative”, respectively.

Step	Output	Hebrew Script
	w<cnjcoo>+’mr<v><actv> <impf><p3><m><sg><consec>	(ו אמר)
Lexicon	w{andc}{<*}y{i}>{paal}’m{˜a}{*?}r>	(ו י אם ר)
Morphophonology	w{andc}{<*}y.o’ma{*?}r	(ו יֹאמַ ר)
Stress Selection	w{andc}{<*}y.o’ma{*}r	(ו יֹאמַ ר)
Stress Movement	w{andc}y.o{+*}’ma{-*}r	(ו ֹּ אמַ ר)
Stress Reduction	way.o{*}’mér	וַ ֹּ אמֶר
Transliteration	וַיֹּאמֶר	—

Table 2: The steps involved in generating a surface form in the Hebrew transducer. Analysis follows the same process but in reverse. Since each layer is a finite-state transformation, the entire sequence can be composed to produce a single transducer, so the intermediate states are not actually present at runtime. The transliteration step is also applied to the analysis side, so the final transducer contains these words as ו<cnjcoo> and אמר<v>. The tags in angle brackets on the first line indicate “coordinating conjunction” and “verb, active, imperfective, 3rd person, masculine, singular, vav-consecutive form”.

	Text	Total	Known	Coverage
Greek	NT	153,665	146,265	95.2%
Hebrew	Gen	20,573	13,201	64.2%

Table 3: Naive coverage for the two analyzers. The Greek analyzer was tested on the New Testament and the Hebrew on the book of Genesis. Total is the number of tokens in the corpus and Known is the number of tokens given an analysis by the analyzer. Coverage is Known as a fraction of Total.

in the Hebrew lexicon and several morphological processes have yet to be implemented at all (adjectives and participles, for instance, currently do not appear at all). In addition, the analyzer currently only accepts text with vowels, which limits the range of texts it can be used on. Fortunately, this latter problem will be straightforward to solve once the overgeneration problem has been dealt with.

In this paper, we have shown by example that finite-state systems are sufficient to model phonological phenomena which operate on the syllable level. Given this, we commend the use of finite-state systems in building analyzers for historical languages as adequate for implementing most morphological processes and beneficial in their capacity to serve as theoretical linguistic documentation for future scholars.

7. Acknowledgements

Our thanks to Matthew Fort and Nick Howell for reading drafts of this paper.

8. Bibliographical References

Crane, G. (1991). Generating and Parsing Classical Greek. *Literary and Linguistic Computing*, 6(4):243–245, 01.

Forcada, M. L., Ginestí-Rosell, M., Nordfalk, J., O’Regan, J., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Sánchez-Martínez, F., Ramírez-Sánchez, G., and Tyers, F. M. (2011). Apertium: a free/open-source platform for rule-based machine translation. *Machine translation*, 25(2):127–144.

Gesenius, W. and Kautzsch, E. (2006). *Gesenius’ Hebrew Grammar*. Dover Publications.

Karttunen, L. (1991). Finite-state constraints. In *Proceedings of the International Conference on Current Issues in Computational Linguistics*.

Khanna, T., Washington, J., Tyers, F., Bayatlı, S., Swanson, D., Pirinen, T., Tang, I., and Alòs i Font, H. (2021). Recent advances in apertium, a free /

open-source rule-based machine translation platform for low-resource languages. *Machine Translation*.

Koskenniemi, K. (1983). *Two-level Morphology: A General Computational Model for Word-Form Recognition and Production*. phdthesis.

Lindén, K., Silfverberg, M., and Pirinen, T. (2009). Hfst tools for morphology–an efficient open-source package for construction of morphological analyzers. In *International Workshop on Systems and Frameworks for Computational Morphology*, pages 28–47. Springer.

Packard, D. W. (1973). Computer-assisted morphological analysis of Ancient Greek. In *COLING 1973 Volume 2: Computational And Mathematical Linguistics: Proceedings of the International Conference on Computational Linguistics*.

Smith, N. (2016). Morphological analysis of historical languages. *Bulletin of the Institute of Classical Studies*, 59(2):89–102.

Swanson, D. and Howell, N. (2021). Lexd: A finite-state lexicon compiler for non-suffixational morphologies. In Mika Hämäläinen, et al., editors, *Multilingual Facilitation*. University of Helsinki Library.

van Emde Boas, E., Rijksbaron, A., Huitink, L., and de Bakker, M. (2019). *The Cambridge Grammar of Classical Greek*. Cambridge University Press.

Wintner, S. (2008). Strengths and weaknesses of finite-state technology: a case study in morphological grammar development. 14(4):457–469.

Yona, S. and Wintner, S. (2008). A finite-state morphological grammar of hebrew. 14(2):173–190.