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 Lex (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) [:{˜a}{*?}] C(3)[reg]
C(1) C(2) [:{˜o}{*?}] 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 .

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 <=> Vowel: _ (h): ;

"determiner before gutturals"
a:â => {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 â if and only if it is preceded by some realization of {h} and followed by either ’ (k) or {'} (y)”). 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 (˚), circumflex (ˆ), and grave (¨).

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 Aper- tium 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 σκηνής (long-long, final stress) and σωτήρα (long-long-short, penultimate stress) are possible, but σκηνῆς (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 παῖδευμενος, παῖδευμενος and παῖδευμενος are impossible, but παῖδευμενος and παῖδευμενος are allowed, and since o is short, so is παῖδευμενος.
4. If the accent falls on a long penultimate syllable and the final syllable is short, the accent must be a circumflex.

So σωτήρα 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 (έν) whenever the final syllable is short and on the second syllable (θρός) otherwise, such as in the genitive ἁν- θρώπου. 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, παῖδευμενά “I am being taught” can have an acute accent on ε, ε, or α, so it will have it on the earliest one, giving παῖδευμενά. Meanwhile, παῖδευο “I am teaching” can have an acute on εν or α or a circumflex on ω, and selecting the earliest one gives παῖδεω.
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 ζ counts as short in this context because it is underlyingly τιμόμαι with ante-penultimate 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.

#### 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.

| Pattern | Rule Description |
|---------|------------------|
| Prefix LongVowel(3) Acute BD FinalShortSyllable | Lexicon LongVowel(4) αιων· αιων· αιων· αιων· 
|        |                  | ειων· ειων· ειων· ειων· ... |
|        |                  | FinalShortSyllable CC ShortVowel(1) BD CC |
|        |                  | LongVowel(3) Acute BD FinalShortSyllable |

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

\[
\{ à:0 \%\} \{ % & + \} \{ \varepsilon \iota | \eta | \alpha | \iota \} \rightarrow à
\]

This rule specifies that if an alpha with an acute accent (à) 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 (ài).

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:0 : 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 (ות), 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.

\[
\text{Stress} = \%\{\%\} \%\{\%\} \%\{\%\} ;
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
In this paper, we have shown by example that finite-state systems can be used as a basis for building analyzers for historical languages as adequate for implementing most morphological processes and benefícial 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., Bayath, S., Swanson, D., Pirinen, T., Tang, I., and Alós 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. PhD thesis.

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 Hamalainen, 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.