Morphological Analysis and Generation for Arabic Dialects

Nizar Habash and Owen Rambow and George Kiraz
Center for Computational Learning Systems
Columbia University
New York, NY 10115, USA
{habash,rambow}@cs.columbia.edu, gkiraz@GorgiasPress.com

Abstract

We present MAGEAD, a morphological analyzer and generator for the Arabic language family. Our work is novel in that it explicitly addresses the need for processing the morphology of the dialects. MAGEAD provides an analysis to a root+pattern representation, it has separate phonological and orthographic representations, and it allows for combining morphemes from different dialects.

1 Introduction

In this paper we present initial work on MAGEAD, a morphological analyzer and generator for the Arabic language family, by which we mean both Modern Standard Arabic (MSA) and the spoken dialects.[1] There has been much work on Arabic morphology (for an overview, see (Al-Sughaiyer and Al-Kharashi, 2004)). Our work is novel in that it explicitly addresses the need for processing the morphology of the dialects. There are several important consequences:

- First, we want to be able to exploit the existing regularities among the dialects and between the dialects and MSA, in particular systematic sound changes which operate at the level of the root consonants, and pattern changes. This requires an **explicit analysis into root and pattern**.
- Second, the dialects are mainly used in spoken communication and in the rare cases when they are written they do not have standard orthographies, and different (inconsistent) orthographies may be used even within a single written text. We thus need a representation of morphology that incorporates **models of both phonology and orthography**.
- Third, in certain contexts, speakers often create words with morphemes from more than one dialect, or from a dialect and MSA. For example, the verb stem may be from MSA while the dialectal present progressive prefix is used. This means that our analyzer needs to be able to have access to **morphological data from more than one member of the language family**.

In addition, we add two general requirements for morphological analyzers. First, we want both a morphological analyzer and a morphological generator. Second, we want to use a representation that is defined in terms of a lexeme and attribute-value pairs for morphological features such as aspect or person. This is because we want our component to be usable in natural language processing (NLP) applications such as natural language generation and machine translation, and the lexeme provides a usable lexicographic abstraction.

We tackle these requirements by implementing the multitape approach of Kiraz (2000), which we

[1] We would like to thank two anonymous reviewers for helpful comments, and Amittai Aviram for his feedback and help with the implementation. The work reported in this paper was supported by NSF Award 0329163.

Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages, pages 17–24,
Ann Arbor, June 2005.

extend by adding an additional tape for independently modeling phonology and orthography. This is the first large-scale implementation of (Kiraz, 2000). We use the AT&T finite-state toolkit (Mohri et al., 1998) for the implementation. The use of finite state technology makes MAGEAD usable as a generator as well as an analyzer, unlike some morphological analyzers which cannot be converted to generators in a straightforward manner (Buckwalter, 2004; Habash, 2004).

This paper is organized as follows. In Section 2, we discuss the linguistic situation of the Arabic-speaking world. In Section 3, we present the relevant facts about morphology in the Arabic language family. We then present our approach to morphological analysis in Section 4, and its implementation in Section 5. We conclude by sketching the planned evaluation.

2 The Arabic Dialects

The Arabic-speaking world is characterized by diglossia (Ferguson, 1959). Modern Standard Arabic (MSA) is the shared written language from Morocco to the Gulf, but it is not a native language of anyone. It is spoken only in formal, scripted contexts (news, speeches). In addition, there is a continuum of spoken dialects (varying geographically, but also by social class, gender, etc.) which are native languages, but rarely written (except in very informal contexts: blogs, email, etc). Dialects differ phonologically, lexically, morphologically, and syntactically from one another; many pairs of dialects are mutually unintelligible. In unscripted situations where spoken MSA would normally be required (such as talk shows on TV), speakers usually resort to repeated code-switching between their dialect and MSA, as nearly all native speakers of Arabic are unable to produce sustained spontaneous discourse in MSA.

3 Arabic Dialect Morphology

3.1 Types of Arabic Morphemes

Arabic morphemes fall into three categories: templatic morphemes, affixational morphemes, and non-templatic word stems (NTWSs). Affixational morphemes are concatenated to form words, while templatic morphemes are interleaved. Templatic morphemes come in three types that are equally needed to create a word stem: roots, patterns and vocalisms. Affixes can be classified into prefixes, suffixes and circumfixes, which precede, follow or surround the word stem, respectively. Finally NTWSs are word stems that are not constructed from a root/pattern/vocalism combination. The following three subsections discuss each of the morpheme categories. This is followed by a brief discussion of some morphological adjustment phenomena.

3.1.1 Roots, Patterns and Vocalism

The root morpheme is a sequence of three, four, or five consonants (termed *radicals*) that signifies some abstract meaning shared by all its derivations. For example, the words[2] كتب *katab* 'to write', كاتب *kaAtib* 'writer', and مكتوب *maktuwb* 'written' all share the root morpheme *ktb* (ك ت ب) 'writing-related'.

The pattern morpheme is an abstract template in which roots and vocalisms are inserted. We will represent the pattern as a string of letters including special symbols to mark where root radicals and vocalisms are inserted. We use numbers (i.e. 1, 2, 3, 4, or 5) to indicate radical position[3] and the symbol *V* is used to indicate the position of the vocalism. For example, the pattern *1V22V3* indicates that the second root radical is to be doubled. A pattern can include letters for additional consonants and vowels, e.g., the verbal pattern *V1tV2V3*.

The vocalism morpheme specifies which short vowels to use with a pattern.[4] A word stem is constructed by interleaving the three types of templatic morphemes. For example, the word stem كتب *katab* 'to write' is constructed from the root *ktb* (ك ت ب), the pattern *1V2V3* and the vocalism *aa*.

[2] In this paper, we use the following conventions for representing examples. All orthographic word forms are provided in undiacritized Arabic script followed by a diacritized version in the Buckwalter transliteration scheme, which is a 1-to-1 transliteration of MSA orthographic symbols using ASCII characters (Buckwalter, 2004). All morphemes are shown diacritized in the Buckwalter transliteration of a plausible standard orthographic representation, though we sometimes include an undiacritized version in Arabic script in parentheses for clarity. All phonemic sequences are written between the usual slashes, but we use the Buckwalter scheme (with obvious adjustments) rather than IPA to represent phonemes.

[3] Often in the literature, radical position is indicated with *C*.

[4] Traditional accounts of Arabic morphology collapse vocalism and pattern.

3.1.2 Affixational Morphemes

Arabic affixes can be prefixes such as *sa+* (+سـ) 'will/[future]', suffixes such as *+uwna* (+ون) '[masculine plural]' or circumfixes such as *ta++na* (تـ++ن) '[subject 2nd person feminine plural]'. Multiple affixes can appear in a word. For example, the word وسيكتبونها *wasayaktubuwnahA* 'and they will write it' has two prefixes, one circumfix and one suffixes:[5]

(1) wasayaktubuwnahA
wa+ sa+ y+ aktub +uwna +hA
and will 3person write masculine-plural it

Some of the affixes can be thought of as orthographic clitics, such as *w+* (+و) 'and' prepositions (*l+* (+ل) 'to/for', *b+* (+بـ) 'in/with' and *k+* (+كـ) 'as') or the pronominal object clitics (e.g., *+hA* (+ها) in the example above). Others are bound morphemes.

3.1.3 Non-Templatic Word Stem

NTWS are word stems that are not derivable from templatic morphemes. They tend to be foreign names and borrowed terms. For example, واشنطن *waA$inTun* 'Washington'. Word stems can still take affixational morphemes, e.g., والواشنطنيون *waAl-waA$inTuniy˜uwn* 'and the Washingtonians'.

3.1.4 Morphological Rewrite Rules

An Arabic word is constructed by first creating a word stem from templatic morphemes or by using a NTWS. Affixational morphemes are then added to this stem. The process of combining morphemes involves a number of phonological, morphemic and orthographic rules that modify the form of the created word so it is not a simple interleaving or concatenation of its morphemic components.

An example of a phonological rewrite rule is the voicing of the /t/ of the verbal pattern *V1tV2V3* (Form VIII) when the first root radical is /z/, /d/, or /*/ (ز, د, or ذ): the verbal stem *zhr+V1tV2V3+iaa* is realized phonologically as /izdahar/ (orthographically: ازدهر) 'flourish' not /iztahar/ (orthographically: ازتهر). An example of a morphemic rewrite rule is the feminine morpheme, *+p* (+ة). Phonologically, it is realized as /t/ word-internally, but it is silent at the end of a word. Orthographically, it is realized as ت *t* in word-internal position (i.e., when followed by a letter), but as ة+ *+p* word-finally. For example, *>amiyrap+nA* (أميرة+نا) is realized as أميرتنا *>amiyratnA* 'our princess' (phonologically: /'amiyratnA/)[6]. Finally, an example of an orthographic rewrite rule is the deletion of the Alif (ا) of the definite article morpheme *Al+* (+ال) in nouns when preceded by the preposition *l+* (+ل) (in both of the following examples, the Alif is silent):

(2) a. للبيت *lilbayti* /lilbayti/ 'to the house'
li+ Al+ bayt +i
to+ the+ house +[genitive]

b. بالبيت *biAlbayti* /bilbayti/ 'in the house'
bi+ Al+ bayt +i
in+ the+ house +[genitive]

3.2 Morpheme Type and Function and the Lexeme

The type of morpheme is independent of the morphological function it is used for (derivational or inflectional). Although affixational morphemes tend to be inflectional and templatic morphemes derivational, there are many exceptions. For example, the plural of كتاب *kitAb* 'book' is not formed through affixation of the inflectional plural morphemes *+At* (+ات) or *+uwn* (+ون), but rather through the use of a different pattern, resulting in كتب *kutub* 'books'. This form of plural construction is called "broken plural" in Arabic to distinguish it from the strictly affixational "sound plural". Conversely, the adjective كتبي *kutubiy˜* 'book-related' is derived from the noun كتب *kutub* 'books' using affixational morphemes. Note that approaches for Arabic stemming that are limited to handling affixational morphology will both miss related terms that are inflected templatically and conflate derived forms generated affixationally.

A common misconception about Arabic morphology concerns the regularity of derivational morphology. However, the meaning of a word cannot be predicted from the root and the pattern+vocalism pair. For example, the masculine noun مكتب *maktab* 'office/bureau/agency' and the feminine noun

[5] We analyze the imperfective word stem as including an initial short vowel, and leave a discussion of this analysis to future publications.

[6] The case markers are ignored in this example for the sake of simplicity.

مكتبة *maktabap* ‘library/bookstore’ are derived from the root كتب *ktb* ‘writing-related’ with the pattern+vocalism *ma12a3*, which indicates location. The exact type of the location is thus idiosyncratic, and it is not clear how the gender can account for the semantic difference. It is this unpredictability of derivational meaning that makes us prefer lexemes as deepest units of morphological analysis, rather than root+pattern pairs. We use the root+pattern analysis only to relate different dialects, and since it has proven useful for certain natural language processing tasks, such as IR (Abu-Salem et al., 1999). We use the lexemic representation to represent the lexicon for applications such as machine translation, including translation between dialects. We return to the definition of “lexeme” in Section 4.2.

3.3 Dialect Morphology

Arabic dialect morphology shares with MSA morphology the root-and-pattern system. Additionally, each dialect morphology shares with MSA morphology some of the morphology lexicon (inventory of morphemes), and the morphological rules. Consider the following forms by way of example:

(3) Egyptian: مبنئلهلكش mabin}ulhalak$ =
ma+ b+ n+ [’wl + V12V3 + iu] +ha +lak +$
MSA: لا نقولها لك lA naquwluha laka =
lA / n+ [qwl + V12V3 + au] +u +ha / la +ka

Here, the Egyptian stem is formed from the same pattern as the MSA stem, but the initial radical, *q* in MSA, has become ’ in Egyptian through regular sound change. The vocalism in Egyptian also differs from that in MSA. Then, we add the first person plural subject agreement marker, the prefix *n+* (which in MSA is the circumfix *n++u*) and the third person feminine singular object clitic *+ha* (same in MSA). In Egyptian, we add a second person masculine singular indirect object clitic *+lak*, the present progressive prefix *b+*, and the negation circumfix *ma++$*. None of these exist in MSA: their meaning is represented with separate words, or as a zero morpheme in the case of the present tense marker. Note that Egyptian orthography is not standardized, so that the form above could be plausibly written in any of the following orthographies, among others: مابنؤلهلكش mAbin&ulhalak$, ما بنئلهلكش mA bin}ulhAlak$, ما بنقلها لكش mA binqulhalak$, مبنقلهلكش mabinqulhalak$, ما بنقولها لكش mA binquwlhA lak$.

Within a word form, all morphemes need not be from the same dialect. Consider the following example.[7] The speaker, who is a journalist conducting an interview, switches from MSA to Egyptian (between square brackets) for a complementizer (اللي *Ailliy*) that introduces a relative clause. He then continues in Egyptian with the prefix *b+* (+بـ) ‘[present progressive]’, and then, inside the word, returns to MSA, using an MSA verb in which the passive voice is formed with MSA morphology, *-tuwaj˜ah* (-توجه) ‘be directed’.

(4) هل كانت إسرائيل هي الأولى [اللي بـ+] -توجه لها القوات المصرية أو كانت توجه ضد قواتعربية أخرى؟

hal kaAnat <isra}iyl AilmafruwD hiya Aal>uwlaY [Ailliy bi+] tuwaj˜ah laha Ailquw˜aAt AilmaSriy˜ap >aw kaAnat tuwaj˜ah Did quw˜aAt Earabiy˜ap >uxraY?
Should it have been Israel first [that] Egyptian armies were directed towards, or were they to be directed against other Arab armies?

4 Morphological Analysis of Arabic

4.1 Previous Work

Despite the complexity of Semitic root-and-pattern morphology, computational morphologists have taken up the challenge of devising tractable systems for computing it both under finite-state methods and non-finite-state methods. Kataja and Koskenniemi (1988) presented a system for handling Akkadian root-and-pattern morphology by adding a additional lexicon component to Koskenniemi’s two-level morphology (1983). The first large scale implementation of Arabic morphology within the constraints of finite-state methods was that of Beesley et al. (1989) with a ‘detouring’ mechanism for access to multiple lexica, which later gave rise to other works by Beesley (Beesley, 1998) and, independently, by Buckwalter (2004).

The now ubiquitous linguistic approach of McCarthy (1981) to describe root-and-pattern morphol-

[7] This example is a transcript of a broadcast originally taken from the Al-Jazeera web site. It can now be found at http://web.archive.org/web/20030210100557/www.aljazeera.net/programs/century_witness/articles/2003/1/1-24-1.htm .

ogy under the framework of autosegmental phonology gave rise to a number of computational proposals. Kay (1987) devised a framework with which each of the autosegmental tiers is assigned a tape in a multi-tape finite state machine, with an additional tape for the surface form. Kiraz (2000,2001) extended Kay's approach and implemented a working multi-tape system with pilot grammars for Arabic and Syriac. Other autosegmental approaches (described in more details in Kiraz 2001 (Chapter 4)) include those of Kornai (1995), Bird and Ellison (1994), Pulman and Hepple (1993), whose formalism Kiraz adopted, and others. In this work we follow the multi-tape approach, and specifically that of (Kiraz, 2000). This is the first large-scale implementation of that approach.

4.2 Our Approach: Outline

In our approach, there are three levels of representation:

Lexeme Level. Words are represented in terms of a lexeme and features. Example:

(5) Aizdaharat: Aizdahar$_1$ POS:V PER:3 GEN:F NUM:SG ASPECT:PERF

The list of features is dialect-independent. The lexeme itself can be thought of as a triple consisting of a root (or an NTWS), a meaning index, and a morphological behavior class (MBC). The MBC maps the features to morphemes. For example, [+FEM] for كاتب *kaAtib* 'writer$_{\text{MASC}}$ ' yields كاتبة *kaAtibap* 'writer$_{\text{FEM}}$ ' which is different from [+FEM] for ابيض *AabyaD* 'white$_{\text{MASC}}$ ' which yields بيضاء *bayDaA'* 'white$_{\text{FEM}}$ '. The MBCs are of course specific to the dialect in question or MSA (though conceivably some can be shared between dialects). For convenience (as in the example above), lexemes are often represented using a citation form.

Morpheme Level. Words are represented in terms of morphemes. (5) is now represented as follows:

(6) Aizdaharat: [zhr + V1tV2V3 + iaa] + at

Surface Level. Words are a string of characters. Using standard MSA orthography, our example becomes:

(7) ازدهرت Aizdaharat

Phonologically, we get:

(8) /izdaharat/

This paper focuses on the morpheme layer (morphology) and the transition between the morpheme and the surface levels. This transition draws on the following resources:

- a unified context-free grammar for morphemes (for all dialects together) which specifies the ordering of affixival morphemes.
- Morphophonemic and phonological rules that map from the morphemic representation to the phonological representation.
- Orthographic rules that map from phonology and morphology to an orthographic representation.

We will next discuss the formal representational and computational framework for these resources.

4.3 Multitape Automata

We follow (Kiraz, 2000) in using a multitape analysis. We extend that analysis by introducing a fifth tier. The five tiers are used as follows:

- Tier 1: pattern and affixival morphemes.
- Tier 2: root.
- Tier 3: vocalism.
- Tier 4: phonological representation.
- Tier 5: orthographic representation.

Tiers 1 through 3 are always input tiers. Tier 4 is first an output tier, and subsequently an input tier. Tier 5 is always an output tier. All tiers are read or written at the same time, so that the rules of the multi-tier automaton are rules which scan the input tiers and, depending on the state, write to the output tier. The introduction of two surface-like tiers is due to the fact that many dialects do not have a standard orthography, as discussed above in Section 3.3.

5 Implementing Multitape Automata

We have implemented multi-tape finite state automata as a layer on top of the AT&T two-tape finite state transducers. Conversion from this higher layer (the new **Morphtools format**) to the Lextools format (an NLP-oriented extension of the AT&T toolkit

for finite-state machines, (Sproat, 1995)) is done for different types of Lextools files such as rule files or context-free grammar files. A central concept here is that of the **multitape string** (MTS), a special representation of multiple tiers in Morphtools that gets converted to a sequence of **multi-tier tokens** (MTT) compatible with Lextools. In the next section, we discuss the conversion of MTS into MTT. Then, we discuss an example rule conversion.

5.1 The Multitape String

A multitape string (MTS) is represented as `<T,R,V,P,O>`. where:

- *T* is the template or basic pattern. The template is represented as a string indicating the position of root consonant (1,2,3,4,5 or C), vowel (V), and any consonant or vowel deemed to be part of the template but not a separate morpheme. For example, Arabic verb form II pattern is represented as `1V22V3` and form VIII is represented as `V1tV2V3`.
- *R* is the root radicals (consonants).
- *V* is the vocalism vowels.
- *P* is the phonological level.
- *O* is the orthographic level.

There are two special symbols: (1) `%` is a wild card symbol that can match anything (appropriate for that tier) and (2) `@<Letter>` (e.g., `@X`) is a variable whose type can be defined explicitly. Both symbols can appear in any tier (except that in our current implementation, `%` cannot appear in tier *T*).

The first (or template) tier (*T*) is always required. The additional tiers can be left underspecified. For example, the full MTS specification for the root `zhr` with form VIII with active vocalism is:

(9) `<V1tV2V3,zhr,iaa>`

When converting an MTS to Lextools format, the *T* tier is used to create a basic default sequence of multi tier tokens (MTTs). For our example (9), `V1tV2V3` leads to this initial MTT sequence:

(10) `[V0%00] [1%000] [t0000] [V0%00] [2%000] [V0%00] [3%000]`

When the symbol *V* appears in the template, a `0` is inserted in the radical position (since no radical can be inserted here) and a wild card is inserted in the vocalism position. The opposite is true for when radical symbol (*C,1,2,3,4,5*) appears in the template, a `0` is inserted in the vocalism tier (as no vowel from the vocalism can be inserted here) and a wild card in the radical tier. all other characters appearing in the template tier (e.g., `t` in the example above), are paired with `0`s in all other tiers.

Additional information from other tiers are then written on top of the default MTT sequence created from the template tier. The representation in (10) is transformed into (12), using the information from the root and vocalism tiers in (9):

(11) `[V0i00] [1z000] [t0000] [V0a00] [2h000] [V0a00] [3r000]`

This sequence corresponds to the form /iztahar/. After applying phonological rules, which will be discussed in the next section, the MTT sequence is as follows. Note that the fourth tier has been filled in.

(12) `[V0ii0] [1z0z0] [t00d0] [V0aa0] [2h0h0] [V0aa0] [3r0r0]`

In this fourth tier, this represents the phonological form /izdahar/. Applying orthographic rules for diacritized orthography, we write symbols into the fifth tier, which corresponds to the orthographic form ازدهر *Aizdahar*.

(13) `[0000A] [V0iii] [1z0zz] [t00dd] [V0aaa] [2h0hh] [V0aaa] [3r0rr]`

Note that the fourth tier provides the (phonemic) pronunciation for the orthography in the fifth tier.

5.2 Representing the Structure of the Word

The basic structure of the word is represented using a context-free grammar (CFG). The CFG covers all dialects and MSA, and only when they differ in terms of the morpheme sequencing does the CFG express dialect-specific rules. How exactly to write this CFG is an empirical question: for example, if frequently speakers mix MSA verb stems with ECA subject agreement suffixes, then the following grammar fragment would not be sufficient. We intend to develop probabilistic models of intra-word code switching in order to guide the morphological analysis in the presence of code switching.

The following rule is the top-level rule which

states that a word is a verb, a noun, or a particle, and it can be preceded by an optional conjunction (for example, *w+*). It holds in all dialects and MSA.

(14) `[WORD] -> [CONJ]? ([VERB]|[NOUN]|[PART])`

The following rule expands verbs to three inflectional types and adds an optional object clitic. For Egyptian (ECA) only, an indirect object clitic can also be added.

(15) `[VERB] -> ([PV_VERB]|[IV_VERB]) [OBJ_PRON]? [ECA:IOBJ_PRON]?`

The next level of expansion then introduces specific morphemes for the two classes of perfective verbs and imperfective verbs. Here, we split into separate forms for each dialect and MSA; we give examples for MSA and Egyptian.

(16) a. `[PV_VERB] -> [MSA:PV_VERB_STEM] [MSA:SUF:PVSUBJ_1S]`
b. `[PV_VERB] -> [ECA:PV_VERB_STEM] [ECA:SUF:PVSUBJ_1S]`

This list is continued (for all dialects and MSA) for all combinations of person, number, and gender. In the case of the imperfective, we get additional prefixes, and circumfixes for the subject clitics. Note that here we allow a combination of the MSA imperfective verb stem with the Egyptian prefixes, but we do not allow the MSA prefixes with the Egyptian verb stem.

(17) a. `[IV_VERB] -> ([MSA:FUT]| [MSA:RESULT]| [MSA:SUBJUNC]| [MSA:EMPHATIC]| [ECA:PRESENT] | [ECA:FUT])? [MSA:IV_VERB_CONJUG]`
b. `[IV_VERB] -> ([ECA:FUT]| [ECA:PRESENT])? [ECA:IV_VERB_CONJUG]`

We then give the verbal stem morphology for MSA (the Egyptian case is similar).

(18) `[MSA:IV_VERB_CONJUG] -> [MSA:PRE:IVSUBJ_1S] [MSA:IV_VERB_STEM] [MSA:SUF:IVSUBJ_1S]`

Again, this list is continued for all valid combinations of person, number, and gender. The verbal stems are expanded to possible forms (combination of pattern and vocalism, not specified for root), or NTWSs. Since the forms are specific to perfective or imperfective aspect, they are listed separately.

(19) `[MSA:PV_VERB_STEM] -> ([MSA:FORM_I_PV]| [MSA:FORM_II_PV]| [MSA:FORM_III_PV]| [MSA:FORM_IV_PV]| ...)`

Each form is expanded separately:

(20) a. `[MSA:FORM_I_PV] -> (<1V2V3,%,aa>| <1V2V3,%,ai>| <1V2V3,%,au>)`
b. `[MSA:FORM_II_PV] -> <1V22V3,%,aa>`

Separate rules introduce the morphemes which are represented by nonterminals such as `[MSA:PRE:IVSUBJ_1S]` or `[ECA:PRESENT]`. Such a context-free specification using MTS is then compiled into MTT sequences in the same manner as described above. The resulting specification is a valid input to Lextools, which generates the finite state machines.

5.3 Representing Rules

We now discuss the representation of rules. We start out with three default rules which are the same for all Arabic dialects and MSA (and possibly for all languages that use templatic morphology). Rule (21a) writes a letter which is in the pattern tier but which is not specified as either root or vocalism to the fourth (phonological) tier, while Rule (21b) and (21c) write a radical and a pattern vowel, respectively.

(21) a. `<@X,,,0> -> @X, @X=[LETTER]`
b. `<C,@X,,0> -> @X`
c. `<V,,@X,0> -> @X`

Phonological and morphemic rules have the same format, as they write to the fourth tier, usually overwriting a symbol placed there by the default rules. Rule (22) implements the rule mentioned in Section 3.1.4 (in Form VIII, the /t/ of the pattern changes to a /d/ if the first radical is /z/, /d/, or /*/). Rule (22) accounts for the surface phonological form in (8); without Rule (22), we would have *iztahar* instead of *izdahar*.

(22) `<t,,,t> -> d / <1,@M,,> _ , @M=[zd*]`

For the orthography we use the fifth tier. As in the case of phonology, we have default rules, which yield a simple phonemic orthography.

(23) a. `<@Y,,,@X,0> -> @X, @Y=[LETTER], @X=[LETTER]`
b. `<V,,@V,@X,0> -> @X, @X=[LETTER]`
c. `<C,@C,,@X,0> -> @X, @X=[LETTER]`
d. `<+,,,+,+> -> 0`

These default rules cover much of MSA orthography, but in addition, there are some special orthographic rules, for example:

(24) `<0V,,@X,@X,0> -> A@X, # _, @X=[LETTER]`

This rule inserts an Alif at the beginning of a word which starts with a pattern vowel.

6 Outlook

This paper describes work in progress. We are currently in the process of populating MAGEAD with morphological data and rules for MSA and Egyptian, with smaller efforts for Yemeni and Levantine. We intend to evaluate MAGEAD using a double strategy: a test suite of selected surface word/analysis pairs which tests the breadth of phenomena covered, and a test corpus, which tests the adequacy on real text. The test suite can be assembled by hand over time from individual examples and is used for regression testing during development, as well as for qualitative assessment of the analyzer or generator. The only test corpus we currently have is the Penn Arabic Treebank for MSA.

In the next phase of the development work, we will link the list of morphemes obtained during analysis to the lexeme level of representation. This will be done using a dialect-specific lexicon, but we will also develop tools to exploit the lexical similarity between the dialects and MSA (and among the dialects) by hypothesizing lexemes based on regular sound change rules.

References

Hani Abu-Salem, Mahmoud Al-Omari, and Martha W. Evens. 1999. Stemming methodologies over individual query words for an arabic information retrieval system. *J. Am. Soc. Inf. Sci.*, 50(6):524–529.

Imad A. Al-Sughaiyer and Ibrahim A. Al-Kharashi. 2004. Arabic morphological analysis techniques: A comprehensive survey. *Journal of the American Society for Information Science and Technology*, 55(3):189–213.

K. Beesley, T. Buckwalter, and S. Newton. 1989. Two-level finite-state analysis of Arabic morphology. In *Proceedings of the Seminar on Bilingual Computing in Arabic and English*, page n.p.

K. Beesley. 1998. Arabic morphology using only finite-state operations. In M. Rosner, editor, *Proceedings of the Workshop on Computational Approaches to Semitic Languages*, pages 50–7, Montereal.

S. Bird and T. Ellison. 1994. One-level phonology. *Computational Linguistics*, 20(1):55–90.

Tim Buckwalter. 2004. Buckwalter Arabic morphological analyzer version 2.0.

Charles F Ferguson. 1959. Diglossia. *Word*, 15(2):325–340.

Nizar Habash. 2004. Large scale lexeme based arabic morphological generation. In *Proceedings of Traitement Automatique du Langage Naturel (TALN-04)*. Fez, Morocco.

L. Kataja and K. Koskenniemi. 1988. Finite state description of Semitic morphology. In *COLING-88: Papers Presented to the 12th International Conference on Computational Linguistics*, volume 1, pages 313–15.

M. Kay. 1987. Nonconcatenative finite-state morphology. In *Proceedings of the Third Conference of the European Chapter of the Association for Computational Linguistics*, pages 2–10.

George Anton Kiraz. 2000. Multi-tiered nonlinear morphology using multi-tape finite automata: A case study on Syriac and Arabic. *Computational Linguistics*, 26(1):77–105.

George Kiraz. 2001. *Computational Nonlinear Morphology: With Emphasis on Semitic Languages*. Cambridge University Press.

A. Kornai. 1995. *Formal Phonology*. Garland Publishing.

K. Koskenniemi. 1983. *Two-Level Morphology*. Ph.D. thesis, University of Helsinki.

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

M. Mohri, F. Pereira, and M. Riley. 1998. A rational design for a weighted finite-state transducer library. In D. Wood and S. Yu, editors, *Automata Implementation*, Lecture Notes in Computer Science 1436, pages 144–58. Springer.

S. Pulman and M. Hepple. 1993. A feature-based formalism for two-level phonology: a description and implementation. *Computer Speech and Language*, 7:333–58.

R. Sproat. 1995. Lextools: Tools for finite-state linguistic analysis. Technical Report 11522-951108-10TM, Bell Laboratories.