A Syllable-based approach to verbal morphology in Arabic

Lynne Cahill
University of Brighton
NLTG, Watts Building, Lewes Rd, Brighton BN2 4GJ, UK
E-mail: L.Cahill@brighton.ac.uk

Abstract

The syllable-based approach to morphological representation (Cahill, 2007) involves defining fully inflected morphological forms according to their syllabic structure. This permits the definition, for example, of distinct vowel constituents for inflected forms where an ablaut process operates. Cahill (2007) demonstrated that this framework was capable of defining standard Arabic templatic morphology, without the need for different techniques. In this paper we describe a further development of this lexicon which includes a larger number of verbs, a complete account of the agreement inflections and accounts for one of the oft-cited problems for Arabic morphology, the weak forms. Further, we explain how the use of this particular lexical framework permits the development of lexicons for the Semitic languages that are easily maintainable, extendable and can represent dialectal variation.

1. Introduction

The Semitic languages are linguistically interesting for a number of reasons. One of the most widely discussed aspects of these languages is the so-called templatic morphology with the typical triliteral verbal (and nominal) roots and their vocalic inflections. In the 1980s a rash of studies emerged discussing ways of describing this morphology and associated problems such as spreading (where only two consonants are specified in the root) and the weak verbs, where one of the consonants in the root is one of the "weak" consonants or glides, *waw* (/w/) or *yaa* (/j/).

Cahill (2007) presented an alternative to these approaches which made use of a framework developed to describe European languages which is based on defining the syllabic structure for each word form. The lexicon is defined as a complex inheritance hierarchy. The fundamental assumption behind this work is that the vocalic inflections can be defined in exactly the same way as an ablaut process commonly seen in European languages. Even the less obviously similar derivations which involve "moving around" of the root consonants (for the different binyan[1] derivations) can be dealt with using the same apparatus as required for consonant adaptations in European languages.

The account in Cahill (2007) describes the basic lexical hierarchy for triliteral verbal roots in MSA with a single verb root being used to demonstrate the ability to generate the full (potential) range of forms with the framework. The account does not cover the agreement inflections (the prefixes and suffixes), nor does it cover anything other than verbs with triliteral strong roots. In this paper we present the latest extensions to this work, which aims ultimately to provide a complete account of the verbal and nominal morphology of Modern Standard Arabic (MSA).

The key developments we report here are:

1. the addition of the agreement inflections;
2. the addition of the apparatus required for handling non-standard roots.

The first of these does not amount to anything very different from a large number of accounts of affixal morphology within an inheritance framework. The second is more interesting, but turns out to be no more challenging for the framework than various types of phonological conditioning in the morphological systems of many European languages. We illustrate our approach to the weak roots with an analysis of one particular weak root, the defective root *r-m-j*, "throw", which has a weak final consonant.

Finally, we discuss the ways in which the framework presented allows for easier extension of the lexicons to enable the development of large-scale lexical resources for the Arabic languages, and how the lexicon structure will permit the definition of dialects in addition to the current account of MSA.

2. MSA verbal morphology

The verbal morphology of the semitic languages has attracted plenty of attention in both the theoretical and computational linguistics communities. What makes it interesting, particularly from the perspective of those exposed only to European languages, is the structure of the stems, involving consonantal roots, vocalic inflections and templates or patterns defining how the consonants and vowels are ordered. Several approaches to the task have been implemented, most based to some degree on the two-level morphology of Koskenniemi (1983), although once adapted to allow for the formation of semitic roots, it ended up being four-level morphology

[1] We use the Hebrew term "binyan" to refer to the different derived forms, also known as "measures" or "forms".

(see e.g. Kiraz (2000)).

The stem formation has already been shown (Cahill, 2007) to be elegantly definable using an approach which was developed mainly for defining European languages such as English and German. We will describe this technique in the next section. However, semitic morphology, and specifically the morphology of MSA, involves other word formation and inflection processes. One of the areas that has attracted a good deal of attention is the issue of what happens when the verb root, traditionally assumed to consist of three consonants, does not fit this pattern. The three principal situations where this happens are in the case of biliteral or quadriliteral roots, where there are either two or four consonants instead of the expected three, and the weak roots, where one of the consonants is a "weak" glide, i.e. either /w/ or /j/.

Where a root has only two consonants, one or other of those consonants is used as the third (middle) consonant, which one depending on the stem shape. Where a root has four consonants, the possible forms are restricted to forms where there are at least four consonant "slots". Early accounts of these types of root include a range of means of "spreading" where post lexical processes have to be invoked to copy one or other of the consonants (see, e.g. Yip (1988)).

The issue of bi- and quadri-literal roots is relatively simply handled within the syllable-based framework, as described in section 4 below. The weak roots are slightly more complex, but nevertheless amenable to definition in a similar way to the kind of phonological conditioning seen, for example, in German final consonant devoicing, where the realisation of the final consonant of a stem depends on whether it is followed by a suffix beginning with a vowel or not. The Syllable-based Morphology framework has been developed to allow for the realisation of fully inflected forms to be determined in part by phonological characteristics of the root or stem in question. This means that, while Arabic weak roots are often cited as behaving differently **morphologically**, we argue that they behave entirely regularly morphologically, but their behaviour is determined by their phonology.

3. Syllable-based morphology

The theory of syllable-based morphology (SBM) can trace its roots back to the early work of Cahill (1990). The initial aim was to develop an approach to describing morphological alternation that could be used for all languages and all types of morphology. Cahill's doctoral work included a very small indicative example of how the proposed approach could describe Arabic verbal stem formation. The basic idea behind syllable-based morphology is simply that one can use syllable structure to define all types of stem alternation, including simple vowel alternations such as ablauts. All stems are defined by default as consisting of a string of tree-structured syllables. Each syllable consists of an onset and a rhyme and each rhyme of a peak and a coda[2]. The simplest situation is where all wordform stems of a particular lexeme are the same. In this case, we can simply specify the onsets, peaks and codas for all of the syllables. For example, the English word "pit" has the root /pIt/ and this is also its stem for all forms (singular, plural and possessive). The phonological structure of this word in an SBM lexicon would therefore by defined as follows[3]:

```
<phn syl1 onset> == p
<phn syl1 peak> == I
<phn syl1 coda> == t
```

This example is monosyllabic, but polysyllabic roots involve identifying individual syllables by counting from either the left or right of a root. For suffixing languages, the root's syllables are counted from the right, while for prefixing languages, they are counted from the left. For Arabic, although both pre- and suffixing processes occur, the decision has been made to count from the right, as there is more suffixation. However, as the roots in Arabic, to all intents and purposes, always have the same number of syllables, it is not important whether we choose to call the initial syllable syl1 or syl2.

In the case of simple stem alternations such as ablaut, the peak of a specified syllable is defined as distinct for the different wordforms. That is, the realisation of the peak is determined by the morphosyntactic features of the form. To use a simple example, for an English word *man*, which has the plural *men*, we can specify in its lexical entry:

```
<phn syl1 peak sing> == a
<phn syl1 peak plur> == E.
```

As the individual consonants and vowels are defined separately for any stem, the situation for Arabic is actually quite straightforward. For each verb form, inflected or derived, the consonants and vowels are defined, not in terms of their position in a string or template, but in terms of their position in the syllable trees. Thus, Cahill (2007) describes how the three consonants can be positioned as the onset or coda of different syllables. The vowels are defined in terms of tense/aspect.

Figure 1 shows how the (underspecified) root structure for the root *katab* looks. This is defined in DATR as follows[4]:

```
<phn syl2 onset> == Qpath:<c1>
```

[2] The term "peak" is used to refer to the vowel portion of the syllable, rather than the sometimes used "nucleus". The syllable structure is relatively uncontroversial, having been first proposed by Pike and Pike (1947).

[3] We use the lexical representation language DATR (Evans and Gazdar, 1996) to represent the inheritance network and use SAMPA (Wells, 1989) to represent phonemic representations.

[4] This is specified at the node for verbs, which defines all of the information that is shared, by default, by all verbs in Arabic.

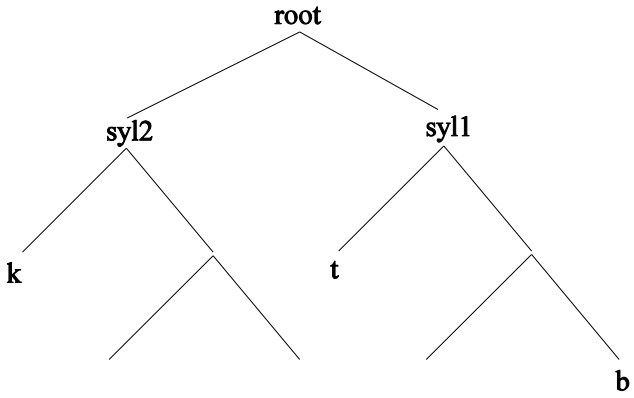

Figure 1: the structure of /katab/

```
<phn syl1 onset> == Qpath:<c2>
<phn syl1 coda>  == Qpath:<c3>
```

These equations simply say that (by default) the onset of the initial syllable is filled by the first consonant (*c1*), the onset of the second syllable is filled by the second consonant (*c2*) and the coda of the second syllable is filled by the third consonant (*c3*). The precise position of the consonants depends not only on the binyan, but also on tense. By default, the past tense has the structure in figure 1, but the present tense has that in figure 2.

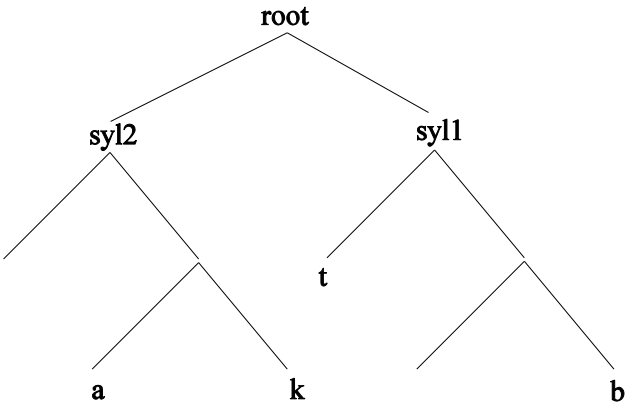

Figure 2: the structure of /aktub/

Affixation is handled as simple concatenation, such that (syllable-structured) affixes concatenate with (syllable-structured) stems to make longer strings of structured syllables. For a simple case such as English noun plural suffixation, for example, we need to specify that a noun consists of a stem and a suffix. We then need to state that, by default, the suffix is null, and that in the case of the plural form, a suffix is added.

```
<mor word form> ==
       "<phn root>" "<mor suffix>"
<mor suffix> == Null
<mor suffix plur> ==
       Suffix_S:<phn form>
```

As we are dealing with phonological forms, we also need to specify how the suffix is realised, which is defined at the separate "Suffix_S" node[5].

One of the key aspects of SBM is that all forms are defined in terms of their syllable structure. This does lead to a slight complication with affixes which consist of a single consonant, for example. The SBM approach to this is to say that there is a necessary post-lexical resyllabification process which takes place after all affixes have been added and so it is not a problem to define affixes as (at least) single syllables, even if they are syllables with no peaks. Although this may seem a little counter-intuitive, the issue of resyllabification is clearly one which must be addressed. If we affix *–ed* (/Id/) to an English verb stem which ends in a consonant, it is almost always the case that that consonant becomes the onset of the suffix syllable, while it is the coda of the final syllable of the stem if no affix is added. Indeed, in most languages it is even the case that resyllabification takes place across word boundaries in normal speech.

4. Extensions to the framework

Cahill's (2007) account of Arabic morphology only covered the stem formation, and did not attempt to cover anything other than straightforward triliteral strong verb roots. In fact, the fragment published in the appendix of that paper includes a single example verb entry, an example of a standard strong triliteral verb. In this section we discuss the three ways in which we have, to date, extended the lexicon.

4.1 Adding more lexemes

We have extended the lexicon initially to include a larger number of strong, triliteral verbs. This is an extremely simple process in the lexicon structure provided as all that needs to be specified are the three consonants in the root. This does result in overgeneration, as all possible stems, for all binyanim, are generated. However, it is a simple process to block possible forms, and there is a genuine linguistic validity to the forms, such that, if a particular verb has a Binyan 9 form, then we know what form it will take.

The issue of how many binyanim to define is an interesting one, and one we will come back to in the discussion of extending coverage to dialects of Arabic. Classical Arabic has a total of fifteen possible binyanim, while MSA makes use of ten of these standardly and two more in a handful of cases.

4.2 Agreement inflections

The next extension to the existing lexicon was to add the agreement inflections. These include prefixes and suffixes and mark the person, number and gender of the form. As noted above, the affixal inflections do not pose any particular difficulties for the syllable-based framework.

[5] For more detail of this type of SBM definition for German, English and Dutch, see Cahill and Gazdar (1999a, 1999b).

The "slots" for the affixes were already defined in the original account, so it was simply a case of specifying the realisations. The exact equations required for this will not be covered in detail here, but we note that the affixes display typical levels of syncretism and default behaviour so that, for example, we can specify that the default present tense prefix is *t-* as this is the most frequent realisation, but the third person present tense prefix is *y-* while the third person feminine prefix is *t-*. This kind of situation occurs often in defining default inheritance accounts of inflection and is handled by means of the following three equations[6]:

```
<agr prefix pres> == t
<agr prefix pres third> == y
<agr prefix pres third femn> == t
```

4.3 Non-standard verb roots

The final extension which we report in this paper is the adaptation of the framework for stem structure to take account of the different types of verb root, as discussed in section 2.

Dealing with biliteral roots involves specifying for each consonant (i.e. onset or coda) defined in the stem structure whether it should take the first or third consonant value, **if the second consonant is unspecified**. Thus, biliteral roots have their second consonants defined thus:

```
<c2> == Undef
```

Then, an example of defining the correct consonant involves a simple conditional statement:

```
<phn syl2 onset> ==
    IF:<EQUAL:<"<c2>" Undef>
    THEN "<c1>"
    ELSE "<c2>"
```

This simply states that, if the second consonant is unspecified, then the first consonant takes this particular position, but if not, then the second consonant will take its normal place. In positions where the absent second consonant is represented by the third consonant simply require the third line above to give *c3* rather than *c1* as the value.

In order to handle quadrilateral roots, we need a separate node for these verbs which defines which of the consonants occupies each consonant slot in the syllable trees. In many cases these are inherited from the Verb node, for example, the first consonant behaves the same in these roots. Typically, where a triliteral root uses *c1*, a quadriliteral root will use *c1*; where a triliteral root uses *c3*, the quadriliteral root will use *c4* in most cases, but *c3* in others; where a triliteral root uses *c2*, the quadriliteral root will either use *c2* or *c3*, so these equations have to be specified.

The weak roots have a glide in one of the consonant positions. This leads to phonologically conditioned variation from the standard stem forms. For example, the hollow verb *zawar* ("visit") has a glide in second consonant position. This leads to stem forms with no middle consonant, and a *u* in place of the two *a*s. In order to allow for this variation, we need to check whether the second consonant is a glide and this will determine the realisations. This check must be done for each onset, peak and coda that is defined as having possible variation, and involves a simple check whether the second consonant is a /w/ or a /j/. In each case the behaviour is the same for the consonant itself, i.e. it is omitted, but different for the vowel. With /w/, the vowel is /u/ but with /y/ it is /i/, in the second vowel position.

There are two possible approaches we could take to defining the different behaviour of weak verbs. The first is to specify a finite state transducer to run over the default forms. For example, we could state that if a verb root has the sequence /awa then this becomes /u:/. The second approach is to define the elements of the syllable structure according to the phonological context in which they occur. We opt for the second of these approaches for a number of reasons. The first is that we wish to minimise the different technologies used in our lexicon. Although FSTs are very simple to implement, we want to resist using them if possible, in order to make use only of the default inheritance mechanisms available to us. The second is that we are not yet at a stage in the project where we have enough varied data for all of the different verb and noun forms to be certain that any transducer we devise will not over apply, whereas we can be more confident of the specific generation behaviour of the inheritance mechanisms we are employing in the lexicon structure as a whole.

One disadvantage of the approach we have chosen to take is that is does result in somewhat more complex definitions in our lexical hierarchy. For example, if we only define strong triliteral verb roots, then our lexical hierarchy can include statements like:

```
<phn syl1 onset> == Qpath:<c1>
```

which are very simple. If we include all of the variation in this hierarchy then we need more statements (to distinguish between, for example, past and present tense behaviour) and those statements are more complex. This is because, even for the standard strong triliteral roots, we need to check for each consonant whether or not it is weak and for each vowel, we need to check whether it is adjacent to a weak consonant . For this reason, we do not include the DATR code which defines the weak verb forms, but rather describe the checks needed.

The approach we take involves two levels of specification. At the first level, each equation defining a consonant or a vowel calls on a simple checking function to determine

[6] We have specified the present tense prefixes without the /a/, as this is present in all forms. We therefore consider that this segment is part of the present tense stem.

whether the realisation is the default one or something different. These calls to checking functions may take different arguments. Thus, the simplest type just needs to be passed the root consonant in question and will determine whether it is realised (if it is strong) or not (if it is weak). In more complex situations, e.g. where a weak root has /u:/ where it would by default have /awa/, we need to pass both the consonant and at least one of the vowels.

The checking nodes are each very simple. The simplest just state that the consonant is realised if it is strong but not if it is weak:

```
Check_ajwaf_cons:
    <$weak> ==
    <$strong> == $strong.
```

We add similar checks to the equations for vowels so that, instead of the default stem form of /zawar/ we get the correct (first and second person[7]) stem of /zur/. The other weak forms involve similar checks for the other consonants.

These checks are very similar to the checks we can see in the syllable-based accounts of, for example, German (Cahill and Gazdar, 1999a). The realisation of the final consonant in any stem in German is dependent on whether or not there is a suffix which begins with a vowel. Therefore, the equation specifying that consonant checks for the beginning of the suffix (if there is one) and for underlying voiced consonants returns the voiced variant only if there is a vowel following, and returns the voiceless variant otherwise.

To clarify the entire process involved in generating a verb form from our lexicon, we shall now describe the derivation of the present tense active third person plural masculine form of the verb "throw" (they(m) throw). This is a weak (defective) verb, with a root of *r-m-j*. The first thing we do is look for the agreement prefix. Our Verb node tells us that this is /j/. Next we need to determine the stem for this form. The stem is defined as having /a/ as the peak of the first syllable (the default value for all present tense forms) and the first consonant of the root, i.e. /r/ as the coda of the first syllable. We determine this by checking whether it is weak or not. Once we have determined that it is a strong consonant, it takes its place in the syllable structure. The onset of the second syllable is the second consonant, in this case /m/, just as it is in most stems. Once again, we check that this is not a weak consonant before placing it in its position. At this point we start to find different behaviour. If the final consonant was strong then we would get a /u/ as the peak of the second syllable. However, as the final consonant, /j/ is weak, the peak is null. Similarly, the final consonant is not realised, because it is weak. So, our stem is fully realised as /arm/. Finally, we look for the agreement suffix, which is defined as /u:na/. So, our fully inflected form is /jarmu:na/. The syllable structure of the stem is shown in figure 3.

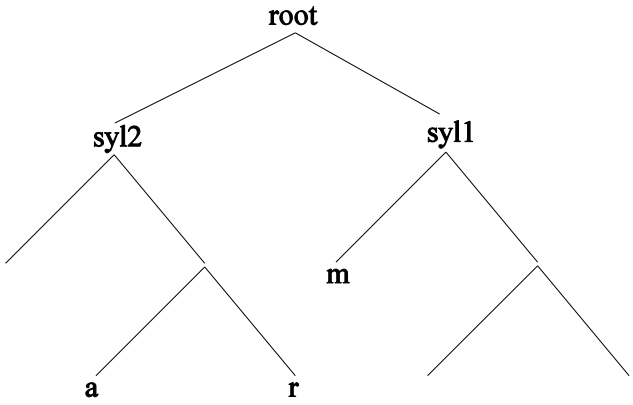

Figure 3: the structure of /arm/

5. Future directions

The extensions we report on here are only the start of a program of research which will add nouns and other non-regular morphological forms (e.g. the broken plural). The project is also going to add orthographic forms, derived from the phonological and morphological information, and supplement these with information about the relative frequency of the ambiguous forms in unvocalised script.

5.1 Extension of the lexicon

The DATR implementation of the lexicon is based on the lexicon structure of PolyLex (Cahill and Gazdar, 1999b). This gives the lexicon two big advantages over other lexicon structures. The first is the default inheritance machinery, which allows very easy extension. It is extremely easy to add large numbers of new lexemes automatically, as long as the hierarchy defines all of the variation. The task is simply to add basic root information (the consonants and the meaning and any irregularities peculiar to that lexeme – although there should not be many irregularities in new additions, as the most frequent words will have been added, and it is usually the more frequent words which are irregular) and choose the node in the hierarchy from which it should inherit. The PolyLex project developed tools to allow the generation of large numbers of additional lexical entries from a simple database format which includes the important information.

Crucially, the use of default inheritance means that, even if we do not have all of the information available to determine the exact morphological behaviour of a particular lexeme, we can assign sensible default values. For example, if we wanted to add a new English noun to our lexicon, and we have not seen an example of that noun

[7] The discussion here has been simplified for the sake of brevity. The first and second person stem forms are the same, and are defined here, but the third person stems are different. This is not a problem for our account, as the framework is specifically designed to allow both morphosyntactic and phonological information to be used in determining the correct form.

in its plural form, we can add it as a regular noun, and generate a plural form which adds the *–s* suffix. This may not be correct, but it is a reasonable guess, and the kind of behaviour we would expect from a human learning a language. This is useful if the data we use to extend our lexicons comes, for example, from corpora – often a necessity for languages which do not have large established resources.

In terms of the Arabic lexicon we describe here, the forms of verbs, even those with weak roots, do not need any further specification, as the lexical hierarchy defines the alternation in terms of the phonological structure of the root. Therefore, if a newly added root has a weak consonant, the correct behaviour will automatically be instigated by the recognition of that weak consonant.

This process has already been tested with a random selection of 50 additional strong verbs, two weak verbs for each of the consonant positions (i.e. two with weak initial consonants, two with weak medial consonants and two with weak final consonants) and one with two weak consonants. The resulting forms for some of these verbs are included in Appendix 2.

5.2 Adding more dialects

Another issue which causes much concern in the representation and processing of Arabic is the question of the different varieties or dialects. Buckwalter (2007) says "... the issue of how to integrate morphological analysis of the dialects into the existing morphological analysis of Modern Standard Arabic is identified as the primary challenge of the next decade." (p. 23). Until relatively recently, the issue of dialects in Arabic was only relevant for phonological processing, as dialects did not tend to be written. However, the rapid expansion of the Internet, amongst other developments, means that written versions of the various dialects are increasingly used, and processing of these is becoming more important.

The PolyLex architecture was developed as a multilingual representation framework, particularly aimed at representing closely related languages (the PolyLex lexicons themselves include English, German and Dutch). The framework involves making use of extended default inheritance to specify information which is shared, by default, by more than one language, with overrides being used to specify differences between languages as well as variant behaviour within a single language (such as irregular or sub-regular inflectional forms). In the case of English, German and Dutch, for example, it is possible to state that, by default, nouns form their plural by adding an *–s* suffix. This is true of all regular nouns in English and of one class of nouns in both Dutch and German. Importantly, those classes in Dutch and German are the classes that new nouns tend to belong to, so assuming that class to be the default works well.

One of the great advantages of such a framework is that, being designed to work for closely related languages, it is also appropriate for dialects of a single language. We can map the situation for MSA[8] and the dialects onto this directly, with MSA taking the place of the multilingual hierarchy and the dialects taking the place of the separate languages here. The assumption is that, by default, the dialects inherit information (about morphology, phonology, orthography, syntax and semantics) from the MSA hierarchy, but any part of that information can be overridden lower down for individual dialects. There is nothing to prevent a more complex inheritance system, for example, to allow two dialects to share information below the level of the MSA hierarchy, but to also specify some distinct bits of information.

6. Conclusions

The approach to Arabic morphology presented here is still in the early stages of development. It does, nevertheless, demonstrate a number of crucial points. First, it backs Cahill (2007) in showing that the SBM approach appears to be adequate to define those aspects of Arabic morphology that have frequently been cited as problematic. It is important to establish proof of concept in employing a new approach to specifying the morphology of any language, and the (admittedly small) lexicon does demonstrate the possibility of handling bi- and quadriliteral roots as well as weak verb roots within the SBM framework. Although not all of the details for all of the verbal morphology have yet been implemented, nothing has been shown to cause any significant difficulties that cannot be overcome in the framework.

Secondly, having established that the approach appears to be feasible for the complexities of Arabic morphology, it follows that the implementation of the morphology in the form of a PolyLex-style lexicon will permit the definition of dialectal variation, thus allowing the development of a full lexicon structure defining MSA, Classical Arabic as well as regional variants in an efficient and practically maintainable way. Although the details remain to be worked out, the assumed structure would involve a core lexicon which defines, for example, all fifteen of the Classical Arabic binyanim, with each of the lexicons for a "real" language specifying which of those are employed within that language or dialect.

The PolyLex lexicon structure allows the definition of defaults, which can be overridden at any of a number of levels. It is possible to override some pieces of information for an entire language or dialect, for a word-class such as nouns, for a sub-class of nouns or verbs or for an individual lexeme. This makes it very efficient at representing lexical information which tends

[8] It may prove more accurate and useful to have Classical Arabic in the multilingual position, as this probably includes more of the range of forms that the different dialects would need to inherit.

to be very largely regular. It also makes it very easy to add new lexemes, even if it has not been wholly established what all of the correct forms of that lexeme are. To use an analogy from child language acquisition, a child hearing an English noun, will assume that its plural is *–s* unless and until they hear an irregular plural form for it. Similarly, a child learning Arabic will assume that a new verb it hears follows the default, regular patterns unless and until they hear non-regular forms. That is the kind of behaviour that our default inheritance lexicon models when adding new lexemes.

7. Acknowledgements

The work reported here was undertaken as part of the ESRC (UK) funded project *Orthography, phonology and morphology in the Arabic lexicon*, grant number RES-000-22-3868. Their support is gratefully acknowledged. I am also grateful to the anonymous reviewers for their constructive comments.

8. References

Buckwalter, Tim. (2007) Issues in Arabic Morphological Analysis. In Abdelhadi Soudi, Antal van der Bosch and Günther Neumann (eds.) *Arabic Computational Morphology* Dordrecht : Springer. pp. 23-41.

Cahill, Lynne. (2007) A Syllable-based Account of Arabic Morphology. In Abdelhadi Soudi, Antal van der Bosch and G ü nther Neumann (eds.) *Arabic Computational Morphology* Dordrecht : Springer. pp. 45-66.

Cahill, Lynne. (1990) Syllable-based Morphology. *COLING-90*, Vol 3, pp. 48-53, Helsinki.

Cahill, Lynne and Gazdar, Gerald. (1999a) German noun inflection. *Journal of Linguistics*, 35 :1, pp. 211-245.

Cahill, Lynne and Gazdar, Gerald. (1999b) The PolyLex architecture : multilingual lexicons for related languages. *Traitement Automatique des Langues*, 40 :2, pp. 5-23.

Evans, Roger and Gazdar, Gerald. (1996) DATR : a language for lexical knowledge representation. *Computational Linguistics*, 22 :2, pp. 167-216.

Kiraz, George. (2000) A Multi-tiered Non-linear Morphology using Multi-tape Finite State Automata : A Case Study on Syriac and Arabic. *Computational Linguistics*, 26 :1, pp. 77-105.

Koskenniemi, Kimmo. (1983) *Two-level Morphology : A General Computational Model for Word-form Recognition and Production.* PhD Dissertation University of Helsinki.

Pike, Kenneth L. and Pike, Eunice V. (1947) Immediate constituents of Mazateco syllables. *International Journal of American Linguistics*, 13, pp. 78-91.

Wells, John. (1989) Computer-coded phonemic notation of individual languages of the European Community. *Journal of the International Phonetic Association*, 19 :1, pp. 31-54.

Yip, Moira. (1988) Template Morphology and the Direction of Association. *Natural Language and Linguistic Theory*, 6.4. pp. 551-577.

Appendix: Sample output

The DATR-implemented lexicon can be compiled and queried. In this appendix, we include the full lexical dumps for three lexemes: the fully regular strong triliteral, *k-t-b*, "write"; the weak (defective) verb *r-m-y*, "throw"; and the "doubly" weak verb *T-w-y*, "fold". The dumps give the present and past active forms for the first binyan.

```
Write:<bin1 mor word past act first sing>
        = k a t a b t u.
Write:<bin1 mor word past act first plur>
        = k a t a b n a:.
Write:<bin1 mor word past act secnd sing
        masc> = k a t a b t a.
Write:<bin1 mor word past act secnd sing
        femn> = k a t a b t i.
Write:<bin1 mor word past act secnd plur
        masc> = k a t a b t u m.
Write:<bin1 mor word past act secnd plur
        femn> = k a t a b t u n n a.
Write:<bin1 mor word past act third sing
        masc> = k a t a b a.
Write:<bin1 mor word past act third sing
        femn> = k a t a b a t.
Write:<bin1 mor word past act third plur
        masc> = k a t a b u:.
Write:<bin1 mor word past act third plur
        femn> = k a t a b n a.
Write:<bin1 mor word pres act first sing>
        = a k t u b u.
Write:<bin1 mor word pres act first plur>
        = n a k t u b u.
Write:<bin1 mor word pres act secnd sing
        masc> = t a k t u b u.
Write:<bin1 mor word pres act secnd sing
        femn> = t a k t u b i: n a.
Write:<bin1 mor word pres act secnd plur
        masc> = t a k t u b u: n a.
Write:<bin1 mor word pres act secnd plur
        femn> = t a k t u b n a.
Write:<bin1 mor word pres act third sing
        masc> = j a k t u b u.
Write:<bin1 mor word pres act third sing
        femn> = t a k t u b u.
Write:<bin1 mor word pres act third plur
        masc> = j a k t u b u: n a.
Write:<bin1 mor word pres act third plur
        femn> = j a k t u b n a.

Throw:<bin1 mor word past act first sing>
        = r a m a j t u.
Throw:<bin1 mor word past act first plur>
        = r a m a j n a:.
Throw:<bin1 mor word past act secnd sing
        masc> = r a m a j t a.
Throw:<bin1 mor word past act secnd sing
        femn> = r a m a j t i.
Throw:<bin1 mor word past act secnd plur
        masc> = r a m a j t u m.
```

```
Throw:<bin1 mor word past act secnd plur
       femn> = r a m a j t u n n a.
Throw:<bin1 mor word past act third sing
       masc> = r a m a a.
Throw:<bin1 mor word past act third sing
       femn> = r a m a t.
Throw:<bin1 mor word past act third plur
       masc> = r a m a w.
Throw:<bin1 mor word past act third plur
       femn> = r a m a j n a.
Throw:<bin1 mor word pres act first sing>
       = a r m i:.
Throw:<bin1 mor word pres act first plur>
       = n a r m i:.
Throw:<bin1 mor word pres act secnd sing
       masc> = t a r m i:.
Throw:<bin1 mor word pres act secnd sing
       femn> = t a r m i: n a.
Throw:<bin1 mor word pres act secnd plur
       masc> = t a r m u: n a.
Throw:<bin1 mor word pres act secnd plur
       femn> = t a r m i: n a.
Throw:<bin1 mor word pres act third sing
       masc> = j a r m i:.
Throw:<bin1 mor word pres act third sing
       femn> = t a r m i:.
Throw:<bin1 mor word pres act third plur
       masc> = j a r m u: n a.
Throw:<bin1 mor word pres act third plur
       femn> = j a r m i: n a.

Fold:<bin1 mor word past act first sing>
       = T a w a j t u.
Fold:<bin1 mor word past act first plur>
       = T a w a j n a:.
Fold:<bin1 mor word past act secnd sing
       masc> = T a w a j t a.
Fold:<bin1 mor word past act secnd sing
       femn> = T a w a j t i.
Fold:<bin1 mor word past act secnd plur
       masc> = T a w a j t u m.
Fold:<bin1 mor word past act secnd plur
       femn> = T a w a j t u n n a.
Fold:<bin1 mor word past act third sing
       masc> = T a w a:.
Fold:<bin1 mor word past act third sing
       femn> = T a w a t.
Fold:<bin1 mor word past act third plur
       masc> = T a w a w.
Fold:<bin1 mor word past act third plur
       femn> = T a w a j n a.
Fold:<bin1 mor word pres act first sing>
       = a T w i:.
Fold:<bin1 mor word pres act first plur>
       = n a T w i:.
Fold:<bin1 mor word pres act secnd sing
       masc> = t a T w i:.
Fold:<bin1 mor word pres act secnd sing
       femn> = t a T w i: n a.
Fold:<bin1 mor word pres act secnd plur
       masc> = t a T w u: n a.
Fold:<bin1 mor word pres act secnd plur
       femn> = t a T w i: n a.
Fold:<bin1 mor word pres act third sing
       masc> = j a T w i:.
Fold:<bin1 mor word pres act third sing
       femn> = t a T w i:.
Fold:<bin1 mor word pres act third plur
       masc> = j a T u: n a.
Fold:<bin1 mor word pres act third plur
       femn> = j a T w i: n a.
```