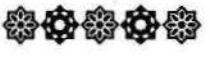
A Finite-State Approach to Arabic Verbal Inflection*

Salah Alnajem
Department of Arabic, College of Arts
Kuwait University

Abstract

This paper introduces a finite-state computational approach to Arabic verbal inflection. This approach handles the non-concatenative discontinuous nature of Arabic verbal inflection, and captures generalizations, sub-generalizations, and syncretisms governing Arabic verbal inflection. It also handles phonological and orthographic variations in Arabic. These variations are caused by phonological and orthographic processes which are applied to inflected verbs. This application yields surface inflected verb forms which are different than the abstract lexical forms. The approach has been implemented using Xerox Finite-State System.



* *I would like to thank Dr Louisa Sadler for here review and comments.*

1. Introduction

Arabic morphology has overt dependencies between canonical forms in its derivational system. Handling such dependencies in a computational approach saves such approach from the redundancy caused by ignoring those dependencies. Besides, Arabic morphology has overt generalizations, sub-generalizations, and syncretisms governing its inflectional system. Redundancy in computational approaches can be avoided if such approaches considered these generalizations, sub-generalizations, and syncretisms.

A previous paper introduced a finite-state computational approach to Arabic verbal derivation, which focused on capturing dependencies in the derivational system. This papers introduces a finite-state computational approach to Arabic verbal inflection, which focuses on capturing generalizations, sub-generalizations, and syncretisms governing Arabic verbal inflection. In addition, this paper considers handling variations which are associated with some of the inflected verbs. Some inflected verbs have a lexical representation which is different than the surface representation. This is caused by specific phonological and orthographic processes which are applied to these inflected verbs. The application of such processes yields a surface inflected verb form which is different than the lexical one. Variations in verb structure will be handled in our finite-state computational approach using Replace Rules and Composition.

2. Linguistic Data

2.1 Derivation

Verbs in Arabic are formed from roots which consist of three or four radicals. From theses roots, verbal stems are formed using a number of canonical forms known as Measures. The measures may also contain stem derivational affixes. Each measure is normally associated with perfective

(active and passive), imperfective (active and passive), and imperative patterns. These inflectional patterns are used to form perfective, imperfective, and imperative verbal stems. The perfective verbs indicate a completed act, while imperfective verbs denote an unfinished act which is just commencing or in progress. Some measures which are intransitive or express a state do not normally associate with passive voice inflection. The stems formed using the above patterns are used to construct verbs through prefixing and/or suffixing inflectional prefixes and/or suffixes.

Verbs in Arabic are either triliteral (having three radicals) or quadriliteral (having four). The triliteral verbal stems are formed using fifteen verbal measures while the quadriliteral verbal stems use four measures. Table 1 shows three triliteral measures and two quadriliteral measures with examples which demonstrate how verbal stems are constructed using those verbal measures. The stems are given in this table *without inflectional prefixes and suffixes*. For the triliteral verbs, the root *ktb* (writing) will be used as an example root; while the root *dhrj* (rolling) will be used for the quadriliteral verbs. Roman numbers are used for reference, with the prefix Q denoting a quadriliteral verbal measures. Not all the measures occur with every root. The root *ktb*, for example, does not occur with some measures. For instance, **nkatab* (Measure VII) is unacceptable in Standard Arabic while *katab* (Measure form I) is acceptable. It should also be noted that some stems undergo certain additional phonological processes. For example, the stem *staktab* (Measure form X) has the surface form *?istaktab* with the epenthetic *?i* prefixed to it. In addition, in this table I will use CV arrays to represent the perfective, imperfective, and imperative stems (patterns) of each measure.

Table 1 The verbal measures

No	Measure	Active Perfective	Passive Perfective	Active Imperfective	Passive Imperfective and Imperative
I	fa9al	$C_1aC_2aC_3$ katab	$C_1uC_2iC_3$ kutib	$C_1C_2iC_3$[1] ktib **OR**	$C_1C_2aC_3$ ktab $C_1C_2uC_3$ ktub **OR** $C_1C_2aC_3$ ktab
II	fa99al	$C_1aC_2C_2aC_3$ kattab	$C_1uC_2C_2iC_3$ kuttib	$C_1aC_2C_2iC_3$ kattib	$C_1aC_2C_2aC_3$ kattab
III	faa9al	$C_1aaC_2aC_3$ kaatab	$C_1uuC_2iC_3$ kuutib	$C_1aaC_2iC_3$ kaatib	$C_1aaC_2aC_3$ kaatab
QI	fa9lal	$C_1aC_2C_3aC_4$ dahraj	$C_1uC_2C_3iC_4$ duhrij	$C_1aC_2C_3iC_4$ dahrij	$C_1aC_2C_3aC_4$ dahraj
QII	tafa9lal	$taC_1aC_2C_3aC_4$ tadahraj	$tuC_1uC_2C_3iC_4$ tuduhrij	$taC_1aC_2C_3aC_4$ tadahraj	$taC_1aC_2C_3aC_4$ tadahraj

Besides, there are variants of measure I with different vocalism for the perfective and imperfective stems. Table 2 shows these variants using the roots *hsb* and *krm*.

Table 2 The variants of measure I (*fa9al*)

Measure	Active Perfective	Passive Perfective	Active Imperfective	Passive Imperfective and Imperative
fa9il	$C_1aC_2iC_3$ hasib			$C_1C_2iC_3$[2] hsib **OR** $C_1C_2aC_3$ hsab
fa9ul	$C_1aC_2uC_3$ karum			$C_1C_2uC_3$ krum

2.2 Inflection

The previous section covered the derivation of verbs in Arabic. We turn now to Arabic verb inflection. The inflection of verbs in Arabic is mainly achieved through the use of prefixes and suffixes denoting person, number, and gender. These non-stem prefixes and suffixes are affixed to the perfective, imperfective, and imperative stems which are produced using verbal measures from roots as shown in Table 1. The reader should note that further changes in stem structure like rejection or transformation of radicals are also applied in the inflection process when we deal with *Weak roots*. Weak roots are roots which contain among their radicals a weak letter or more than one weak letter. In addition, in verb inflection, there is a process of vowel rejection which is applied when we deal with triliteral roots that have their last two letters identical. Such changes will be considered in future research.

The perfective, imperfective, and imperative stems differ in their inflection for person, number, and gender. This difference appears in the inflectional non-stem affixes. The following tables are paradigms illustrating verb inflection in Arabic. These paradigms are illustrated with stems formed using measure I from the root *ktb*:

Table 3 The perfective active paradigm

	Mas	**Fem**
Singular		
1st	katab-tu	katab -tu
2nd	katab -ta	katab -ti
3rd	katab -a	katab -at
Dual		
1st	katab -naa	katab -naa
2nd	katab -tumaa	katab -tumaa
3rd	katab -aa	katab -ataa
Plural		
1st	katab -naa	katab -naa
2nd	katab -tum	katab -tunna
3rd	katab -uu	katab -na

Imperfective inflection is normally achieved by prefixing and suffixing non-stem imperfective prefixes and suffixes, which indicate person, number, and gender, to imperfective (active and passive) stems that are produced using verbal measures as shown in Table 1[3]. The perfective and imperfective suffixes are unmarked for voice (i.e. they can be used for both the active and passive inflection). The imperfective active paradigm is shown in the following table:

Table 4 The imperfective active paradigm

	Mas	Fem
Singular		
1st	?a-staktib-u	?a-staktib-u
2nd	ta-staktib-u	ta-staktib-iin
3rd	ya-staktib-u	ta-staktib-u
Dual		
1st	na-staktib-u	na-staktib-u
2nd	ta-staktib-aan	ta-staktib-aan
3rd	ya-staktib-aan	ta-staktib-aan
Plural		
1st	na-staktib-u	na-staktib-u
2nd	ta-staktib-uun	ta-staktib-na
3rd	ya-staktib-uun	ya-staktib-na

Lastly, we turn to the imperative inflection that normally takes the form of imperative suffixes, indicating person, number, and gender, which are suffixed to imperative stems. These imperative stems are produced using verbal measures from roots as shown in Table 1. The reader should note that there is a phonological process normally applied in the imperative inflection which prefixes the epenthetic syllable *?V* to the imperative stems of some measures. This epenthetic prefixation is

governed by a phonological generalization, and it will be considered later.

Table 5 The imperative paradigm

	Mas	Fem
Singular		
2nd	kattib	kattib-ii
Dual		
2nd	kattib-aa	kattib-aa
Plural		
2nd	kattib-uu	kattib-na

3. Previous Computational Approaches to Arabic Morphology

Since Arabic morphology has a special non-concatenative nature, special computational approaches were suggested in the literature to handle it computationally. In this respect, (Kay, 1987) introduced a finite-state model for the generation and analysis of Arabic morphology. This finite-state model was inspired by the theory of Autosegmental Phonology, and by the Templatic Analysis of Arabic morphology which was introduced by (McCarthy, 1981; McCarthy, 1982). In his proposal, Kay uses Finite-State Transducers which work with four tapes (instead of the ordinary two). These tapes convey the root tier, template tier, vocalism tier, and the surface string respectively. These four tapes are read and combined by Kay's finite state transducers at the same time instead of reading two tapes, as normally happens.

(Beesley, 1991) introduced a large computational system for Arabic morphological analysis based on the Two-Level Morphology approach. This model uses two lexicons: one for the roots and another for the patterns precompiled with their vocalisms. Root and pattern

interdigitation is done using an algorithm called Detouring which consists of programs that allow the system to follow, at the same time, a lexical path through two lexicons, which are the root and pattern lexicons. This process is done at run time. Two-level rules are used to deal with morphological and orthographic variations. Feature unification is used to deal with inflection. In 1996 and 1998 ((Beesley, 1996; Beesley 1998)), this system was rebuilt and enhanced using Xerox Finite-State Tools. Root, pattern, and vocalism interdigitation is done through intersection at compile time. Finite-state two-level rules are used to map abstract lexical strings to surface string. These rules are used to handle deletion, assimilation, and other variations. The rules are intersected into a single rule finite state transducer. The application of these two level rules is done through composing the rule transducer to the main lexicon transducer. Composition is also used to deal with long-distance morphotactic restrictions and other phenomena.

(Kiraz, 1994; Kiraz, 2001) introduced a multi-tape two-level computational model of Arabic morphology. This model is built on a developed version of the two-level morphology technique introduced by Pulman and Hepple (Pulman and Hepple, 1993) and others. In this developed version, the two-level rules map between *strings* of lexical and surface symbols. This is different from what we have in the conventional two-level rules where we deal with one *symbol* at a time.

Kiraz adopts Kay's (Kay, 1987) idea of using four tapes to deal with Arabic morphology. Three tapes are used to represent the lexical level and one tape to represent the surface level. Beside the previous conventions, Kiraz introduced extensions related to the expressions in the lexical side of the two-level rules. Kiraz applied this computational model using three linguistic theories: the Templatic Analysis (McCarthy, 1981; McCarthy, 1982), the Moraic Analysis (McCarthy and Prince, 1990), and the Affixational Analysis (McCarthy, 1992).

In relation to the inflectional prefixes and suffixes, there is an apparent phenomenon in Arabic morphology represented by the syncretisms in inflectional paradigms among inflectional prefixes and suffixes. The syncretism is a phenomenon in which two or more morphemes which are morphosyntactically distinct appear identical in form. Beesley in his proposals did not discuss such syncretisms nor how they could be dealt with. These syncretisms have also been ignored in Kay's and Kiraz' proposals. In contrast, and in my finite state approach, syncretisms will be handled.

Another interesting point which was not discussed in Beesley's models (nor in Kay's and Kiraz' models) is the regularity of Arabic inflectional morphology. An example of such regularities is represented in the regularity which states that the imperative verbal stems are identical to their corresponding active voice imperfective stems. Such regularities are handled straightforwardly in our finite state approach to Arabic morphology as will be shown later.

`Moreover, focusing on capturing the dependencies[4] between measures, generalizations[5], and sub-generalizations in Arabic morphology is absent in Kay's, Beesley's, and Kiraz' models. Kiraz (1994, 2001) captured to some extent the dependencies between a number of measures through applying McCarthy's (1992) Affixational analysis, but he did not capture the generalizations and syncretisms in the inflectional system. In his research, he did not focus on capturing and formalizing dependencies compared to our finite state approach. Capturing such dependencies, generalizations, and sub-generalizations yields a linguistically motivated computational formalization for Arabic morphology. It also saves this formalization from the redundancy caused by ignoring such dependencies, generalizations, and sub-generalizations; which are overt in Arabic morphology.

In addition to these finite state approaches, other non-finite-state approaches to Arabic morphology have been suggested. Cahill (Cahill,

1990; Cahill, 1991), Gibbon (Gibbon, 1990), Reinhard and Gibbon (Reinhard and Gibbon, 1991) suggested three models of Arabic morphology using the Default Inheritance approach.

The models introduced by Cahill , Gibbon, and Reinhard and Gibbon did not deal with syncretisms in inflectional paradigms. In contrast, our approach captures syncretisms in inflectional paradigms in a compact non-redundant manner as will be seen later. Furthermore, and compared with the models of Gibbon (Gibbon, 1990) and Reinhard and Gibbon (Reinhard and Gibbon, 1991), our finite state approach have captured dependencies between verbal measures in a compact non-redundant manner. Gibbon (Gibbon, 1990) and Reinhard and Gibbon (Reinhard and Gibbon, 1991) did not deal properly with such dependencies between verbal measures, and did not discuss how such dependencies could be handled. What we found is a limited handling of some instances of dependency which generally involve sharing substructures of templates and template prefixing as in *kattab* (II, Perf Act) and *takattab* (V, Perf Act) without handling other dependencies which require further changes inside the stem structure such as what we have in the dependency between *katab* (I, Perf Act) and *kattab* (II, Perf Act). In contrast, our approach has presented a more extensive coverage of the dependencies between verbal measures, and have dealt with dependencies requiring change inside stem structure such as the dependency which requires the gemination of the medial radical of the derived stem as in *katab* (I, Perf Act) and *kattab* (II, Perf Act).

4. Capturing Generalizations and Syncretisms

4.1 Generalizations about Verb Inflection

There is a number of generalizations in Arabic morphology which can be captured in relation to the verbal inflectional system. Considering the

perfective, imperfective, and imperative verbal paradigms (see tables one), the following generalizations emerge:

(1) Generalizations about Verbal Inflectional Paradigms:

a- Perfective active and passive inflectional paradigms are constructed by suffixing a perfective inflectional suffix (like: {*-tu*}) to a perfective stem (like *katab*). There are no perfective inflectional prefixes.

b- Imperfective active and passive inflectional paradigms are constructed by prefixing an imperfective inflectional prefix (like:{*ya-*}) to an imperfective stem (like: *staktib*) and suffixing an imperfective inflectional suffix (like {*-u*}).

c- Imperative inflectional paradigms are constructed by suffixing an imperative inflectional suffix (like: {*-aa*}) to an imperative stem (like: *staktib*). There are no imperative inflectional prefixes.

d- The imperative stems are the same as the imperfective active stems.[6]

e- The imperfective active prefixes end with the vowel '*a*'. The imperfective passive prefixes are the same as the imperfective active prefixes but with changing the prefix vowel to '*u*'.

Beside the previous generalizations relating to the formation of verbal paradigms, there is a sub-generalization related to the inflectional paradigms which correspond to the verbal measures II,III,IV, and QI. The inflectional paradigms which correspond to these measures are constructed using the same way of constructing verbal inflectional paradigms which is mentioned in (1), except that the imperfective active prefix will end with the vowel '*u*' instead of '*a*'[7]. In other words, the vowel '*a*' of the imperfective active prefixes is changed to '*u*' when these prefixes are used with the imperfective active stems that consist of exactly two heavy syllables. These dual heavy syllable stems are those imperfective active stems produced using measures II,III,IV, and QI such as the stems *kattib* (Measure II) and *kaatib* (Measure III).

There are some exceptions relating to the passive voice inflection which are as follows:

- Measures IX , XI , XII , XIII , XIV, and XV do not normally have passive voice stems.
- The measures $C_1aC_2uC_3$ and $C_1aC_2iC_3$ (the variant of measure I) normally lack passive voice stems (see Table 2).
- The active voice stems of measure I ($C_1aC_2aC_3$) which do not express an act but rather a state or condition (such as being) do not normally associate with passive voice stems. An example of this is *fasad* (to be bad, Perf Act).

4.2 Syncretisms in the Inflectional Paradigms

The verbal inflectional paradigms exhibit a number of synretisms and partial syncretisms. An instance of such syncretisms is represented by the prefix {*ta-*} in the imperfective active paradigm which indicates the second person. Another example of syncretisms is represented by the suffix {*-naa*} which indicates perfective first person. An example of the partial syncretisms is illustrated by the incorporated (connected) pronoun '*aa*' (dual) which is used as a complete suffix as in {*-aa*} (Imperfective 2nd Dual Mas/Fem), and as a *part* of a suffix as in the suffix {*-tumaa*} (Perfective 2nd Dual Mas/Fem) and {*-aan*} (Imperfective 2nd Dual Mas/Fem).

5. A Finite-State Approach to Arabic Verbal Inflection

This section introduces the inflectional part of a computational morphological analysis and generation approach to Arabic verbal morphology[8], in which I attempt to capture dependencies in verb derivation; and generalizations, sub-generalizations, and syncretisms governing verb inflection. The derivational part of the approach is

covered in (Alnajem, 2004a). This analysis and generation approach will be expanded later to handle Arabic nouns and capture dependencies, generalizations and syncretisms existing in Arabic noun derivation and inflection. The approach is implemented using Xerox Finite-State Tools and it adopts Xerox Finite-State Calculus.

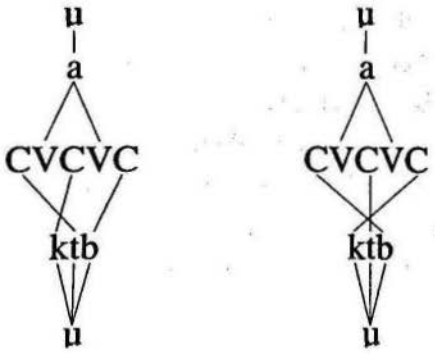
This finite state approach adopts McCarthy's Templatic Analysis (McCarthy, 1981; McCarthy, 1982). McCarthy's Templatic Analysis builds on the theory of Autosegmental Phonology. Before this, there was no theoretically elegant way of describing the method of word formation found in non-concatenative morphological systems (Katamba,1993). The templatic analysis represents a modern morphophonological formalization of the traditional morphological account for Arabic.

The templatic analysis provides a multilinear approach to Arabic morphology. This analysis splits word morphemes into separate layers (root tier, vocalism tier[9], skeletal tier, and affixal tiers). The use of separate autosegmental tiers is motivated by the fact that non-concatenative morphological systems have discontinuous morphemes. In other words, a word (stem) in such morphological systems is not normally constructed from a sequence of morphemes which are adjacent to each other in a linear order. The sounds which correspond to a single morpheme are normally *not adjacent* to each other and are *interrupted* by sounds of other morphemes. So, since we have separate autosegmental tiers which isolate each morpheme exclusively, we can deal with the sounds of discontinuous morphemes without losing the structure of each of these morphemes

Another linguistic motivation behind the use of multiple autosegmental tiers appears when we consider specifying a separate tier for the root. By dealing with the root as a single morpheme placed on a separate autosegmental tier, and by using what McCarthy calls the Well-

formedness Condition, we can prevents measures from re-ordering their radicals to give unacceptable forms like, for example, **9afal* ($C_2aC_1aC_3$) or **la9af* ($C_3aC_2aC_1$) instead of *fa9al* ($C_1aC_2aC_3$). This means that we can predict that there will be only a limited number of measures available due to the well-formedness condition and the linear ordering of the root consonantism which is placed on a separate autosegmental tier. To clarify the point, consider the following example. In the templatic analysis, it is unacceptable to have, from a root like *ktb*, a vocalism like *a*, and a measure like CVCVC (measure I) a stem like *takab* or *batak*. The only possible stem is *katab*, because the well-formedness condition and the Universal Association Conventions prevent the CV-template (measure) from re-ordering its radicals since this will yield to the crossing of association lines:

(2)



takab *batak*

The well-formedness condition states that when associating the melodic elements to the appropriate melody-bearing elements, the association lines *must not cross*. The well-formedness condition will be represented in our finite state approach by a numbering convention used in representing the radicals.

In contrast, in the traditional morphological account for Arabic there is no real means of preventing measures from re-ordering their radicals. There is no constraint in the system which prevents changing the measure *fa9al* ($C_1aC_2aC_3$), for example, to **9afal** ($C_2aC_1aC_3$), **fala9** ($C_1aC_3aC_2$), or **9alaf** ($C_2aC_3aC_1$). In addition, there is no way which allows us to predict that only a limited number of measures will be available as is the case in the templatic analysis.

5.1 Inflectional Stems

As mentioned in a previous section, the inflection of verbs in Arabic is mainly achieved through the use of inflectional prefixes and suffixes. These non-stem prefixes and suffixes are affixed to the perfective, imperfective, and imperative stems which are produced using verbal measures from roots. The formation of inflectional stems in our finite-state approach was covered in (Alnajem, 2004a). In our finite-state approach, the formation of the inflectional stems of each measure from different roots is achieved thorough applying the union, subtraction, and composition operations to finite-state transducers. Replace rules are also used beside these operations. This yields a lexicon transducer whose upper language is a set of morphological feature tags, and its lower language is a set of (abstract) surface inflectional stems. This lexicon transducer accepts surface stems and returns lexical morphological feature

tags. This bidirectional network can also generate surface stems given morphological feature tags.

5.2 Inflectional Affixation

Inflectional affixation in Arabic is achieved through suffixation or circumfixation (prefixation and suffixation). Suffixation is used to form the perfective and imperative verbs. Circumfixation is used to form the imperfective verbs. We start now with suffixation which is straightforward. In this finite-state approach, suffixation is achieved through applying the *concatenation* operation to concatenate suffix transducers (representing the perfective, imperfective, or imperative suffixes) to the lexicon transducer. The upper language of each of these suffix transducers is a set of morphosyntactic feature strings indicating the tense/mood (perfective, imperfective, or imperative) , number, person, and gender. A typical string which represents the upper language is:

```
+PERF+FIRST+SING+MAS
```

The lower language is a set of inflectional suffixes. For instance, the transducer of the perfective suffixes contains, in addition to others, the following morohosyntactic feature strings:

```
+SECOND+SING+MAS
+SECOND+PL+MAS
+THIRD+SING+FEM
```

which correspond to the following set of suffixes in the lower side respectively:

```
- ta
- tum
- at
```

On the other hand, imperfective prefixation is achieved through applying the concatenation operation to concatenate the lexicon transducer to the prefix transducer representing the imperfective prefixes

To apply suffixation, we firstly define and compose a transducer for the perfective suffixes, one for the imperfective suffixes, and another for the imperative suffixes. The following is a definition of the transducer holding to the perfective suffixes:

```
define SUFFPERF

      [ [ [%+FIRST %+SING %+MAS] | [%+FIRST %+SING %+FEM] ]:[%- t
   u] ] |
         [ [%+SECOND %+SING %+MAS]:[%- t a] ] |
         [ [%+SECOND %+SING %+FEM]:[%- t i] ] |
         [ [ [%+FIRST %+DUAL %+MAS] | [%+FIRST %+DUAL %+FEM] |
         [%+FIRST %+PL %+MAS] | [%+FIRST %+PL %+FEM] ]:[%- n aa] ]
         |
         [ [ [%+SECOND %+DUAL %+MAS] | [%+SECOND %+DUAL %+FEM]
         ]:[%- t u m aa] ] |
         [ [%+SECOND %+PL %+MAS]:[%- t u m] ] |
         [ [%+SECOND %+PL %+FEM]:[%- t u n n a] ] |
         [ [%+THIRD %+SING %+MAS]:[%- a] ] |
         [ [%+THIRD %+SING %+FEM]:[%- a t] ] |
         [ [%+THIRD %+DUAL %+MAS]:[%- aa] ] |
         [ [%+THIRD %+DUAL %+FEM]:[%- a t aa] ] |
         [ [%+THIRD %+PL %+MAS]:[%- uu] ] |
         [ [%+THIRD %+PL %+FEM]:[%- n a] ] ;
read regex SUFFPERF;
save stack suffperf.fst
```

After defining and compiling the transducer, we concatenate this transducer to a transducer holding the perfective stems. The perfective stems are extracted from the lexicon transducer which holds all the inflectional stems through a filter composed to the upper side of the lexicon transducer, so that we are left with a transducer holding the perfective stems only:

```
define  PERfilter $[?* %+PERF ?*];
read regex PERfilter .o. @"stems.fst";
save stack PERF.fst;
```

Now we concatenate to the transducer holding the perfective stems the transducer holding the perfective suffixes:

```
read regex @"PERF.fst" @"suffperf.fst";
save stack "perfective.fst"
```

The result transducer is saved for later use. Notice that concatenation is indicated by a space in Xerox Finite-State tools. The suffixation of the other inflectional suffixes is done in a similar way.

Having considered suffixation, we turn now to circumfixation which is more complicated. Circumfixation is used with the imperfective stems. As mentioned before, imperfective verbs are formed by prefixing an imperfective inflectional prefix to an imperfective stem and suffixing an imperfective inflectional suffix. In this circumfixation process, the prefix should unify with the suffix in their morphosyntactic features indicating person, number, and gender. So, a verb like (3-a) is unacceptable, since there is a contradiction in the person feature between the prefix and suffix. The verb in (3-b) is acceptable because there is no contradiction:

(3)

a-

* ?a -	staktib	-iin
[+ *First* + *Sing* + *Mas / Fem*]		[+ *Second* + *Sing* + *Fem*]

b-

ta -	staktib	-iin
[+ *Second* + *Sing / Dual / Plural* + *Mas / Fem*]		[+ *Second* + *Sing* + *Fem*]

This long-distance dependency in verb structure between the inflectional prefixes and suffixes of the imperfective verbs represents a challenge facing the computational approaches to Arabic morphology. In the literature, Beesley (Beesley, 1991) used feature unification to deal with this long-distance dependency in his two-level approach. In (Alnajem, 1998) I used Default Inference in my Default Inheritance computational approach to Arabic morphology. In this finite-state approach, I use filters and composition to handle this long-distance dependency.

Firstly we define filters to extract the imperfective stems from the lexicon transducer. Two classes of imperfective active stems should be extracted since each class is preceded by a different active voice prefix vowel. As we mentioned in the sub-generalization of Section 4.1, the verbal inflectional paradigms which correspond to measures II,III,IV, and QI are constructed using the same way of constructing the verbal inflectional paradigms of the other measures, except that the imperfective active prefix will end with the vowel '*u*' instead of '*a*'. This means that

we have two classes of imperfective active stems. The first is the default class which contains the majority of the imperfective active stems that are preceded by a prefix with the default vowel '*a*'. The second exceptional class contains the imperfective active stems of measures II,III,IV, and QI which are preceded by a prefix with the vowel '*u*'. Similar to the imperfective active stems of measures II,III,IV, and QI , the imperfective passive stems are preceded with the imperfective prefixes which end with the vowel '*u*'.

A filter is defined to extract the imperfective active stems of all the measures from the lexicon transducer:

```
define  IMPERACTfilter $[?* %+IMPER %+ACT ?*];
```

Another filter is defined to extract the exceptional imperfective active stems of measures II,III,IV, and QI:

```
define  IMPERACTfilter234q1 $[?* %+FORM2 %+IMPER %+ACT ?*]
| $[?* %+FORM3 %+IMPER %+ACT ?*] | $[?* %+FORM4 %+IMPER
%+ACT ?*] | $[?* %+FORMQ1 %+IMPER %+ACT ?*]     ;
```

Notice the use of the union operator (|) to union the filters of measures II,III,IV, and QI. Finally a filter is defined to extract the imperfective passive stems:

```
define  IMPERPASSfilter $[?* %+IMPER %+PASS ?*];
```

After defining the imperfective filters, we start extracting the required stems from the lexicon transducer through the use of composition. Firstly we extract a general transducer holding the active voice imperfective stems of all measures:

```
read regex IMPERACTfilter  .o. @"stems.fst";
save stack IMPERFACT1.fst;
clear stack;
```

Then we extract the exceptional transducer which holds the exceptional imperfective active stems of measures II,III,IV, and QI:

```
read regex IMPERACTfilter234q1  .o. @"stems.fst";
save stack IMPERFACT2.fst;
clear stack;
```

Then we subtract from the general imperfective active transducer the exceptional imperfective active traducer to get a default transducer holding the imperfective active stems of all measures *except* the exceptional imperfective active stems of measures II,III,IV, and QI:

```
read regex @"IMPERFACT1.fst"  - @"IMPERFACT2.fst";
save stack IMPERFACT1.fst;
clear stack;
```

The imperfective passive stems are extracted in the normal way using composition:

```
read regex IMPERPASSfilter  .o. @"stems.fst";
save stack IMPERFPASS.fst;
clear stack;
```

Now we turn to the definition of the transducers holding the imperfective active prefixes and suffixes. The following is the definition of the imperfective suffixes:

```
define SUFFIMPER    [ [ [%+FIRST %+SING %+MAS] | [%+FIRST
%+SING %+FEM]| [%+FIRST %+DUAL %+MAS] | [%+FIRST %+DUAL
%+FEM] | [%+FIRST %+PL %+MAS] | [%+FIRST %+PL %+FEM] |
[%+SECOND %+SING %+MAS]|
[%+THIRD %+SING %+MAS] | [%+THIRD %+SING %+FEM]  ]:[%- u] ]
|
[ [%+SECOND %+SING %+FEM]:[%- ii n] ] |[ [ [%+SECOND %+DUAL
%+MAS] | [%+SECOND %+DUAL %+FEM] | [%+THIRD %+DUAL   %+MAS]
|
[%+THIRD %+DUAL %+FEM]  ]:[%- aa n] ] | [ [ [%+SECOND %+PL
%+MAS] | [%+THIRD %+PL %+MAS]  ]:[%- uu n] ] | [          [
[%+SECOND %+PL %+FEM] | [%+THIRD %+PL %+FEM] ]:[%- n a] ];

read regex SUFFIMPER;
save stack suffimper.fst
```

The transducer holding the imperfective active prefixes is defined with an upper language which consists of the union of sets, each of the sets contains one feature tag only (PNG) where P stands for person, N for number, and G for gender. This feature tag will be used later to handle the long-distance dependency between imperfective active prefixes and suffixes. The following is the definition of the default imperfective active prefixes with the default vowel '*a*':

```
define IMPERPRE1 [[[%+FSM] | [%+FSF]] : [%@a %-]] |
[[[%+FDM] | [%+FDF] | [%+FPM] | [%+FPF]] : [na %-]] |
[[[%+SSM] | [%+SSF] | [%+SDM] | [%+SDF] | [%+SPM] | [%+SPF] | [%+TSF] |
[%+TDF]] : [ta %-]] |
[[[%+TSM] | [%+TDM] | [%+TPM] | [%+TPF]] : [ya %-]];

read regex IMPERPRE1;
save stack preimper1.fst
```

The transducer holding the exceptional imperfective active prefixes with the exceptional vowel '*u*' is simply the transducer holding the default prefixes but with substituting the default vowel '*a*' with '*u*'. This substitution is achieved through `substitute symbol` command, which changes all the occurrences of '*a*' to '*u*' in the prefix transducer. The changed transducer is saved as a different finite-state transducer (`preimper2.fst`):

```
substitute symbol u for a;
save stack preimper2.fst
```

So far, we extracted the imperfective active and passive stems, and defined the imperfective circumfixes (prefixes and suffixes). Now we affix the imperfective circumfixes taking care of the long-distance dependency existing between each prefix and its corresponding suffix, and taking care of the sub-generalization mentioned in Section 4.1. To handle the long-distance dependency, special filters are defined. Each filter states that the feature tag (PNG) should be followed be a compatible morphosyntactic feature string. For example the feature tag FSM (First

Singular Masculine) should be followed by the morphosyntactic feature string +FIRST+SING+MAS. The following is a partial definition of the filters corresponding the first person circumfixes:

```
define LONGDfilter1st $[%+FSM ?* %+FIRST %+SING %+MAS] |
          $[%+FDM ?* %+FIRST %+DUAL %+MAS] |
          $[%+FPM ?* %+FIRST %+PL %+MAS] |
          $[%+FSF ?* %+FIRST %+SING %+FEM] |
          $[%+FDF ?* %+FIRST %+DUAL %+FEM] |
          $[%+FPF ?* %+FIRST %+PL %+FEM] ;
```

The regular expression ?* indicates zero or more symbols of any type. In similar way, the filters corresponding to the second and third person circumfixes are defined.

Then the default prefixes (with the vowel '*a*') are prefixed to the default imperfective active stems, and the imperfective suffixes are suffixed to these default stems. This affixation is achieved through concatenation. The long-distance dependency is handled through composing the long-distance dependency filters to the upper (lexical) side of the transducer resulted from the concatenation operation. The concatenation is achieved through the following lines:

```
read regex @"preimper1.fst" @"IMPERFACT1.fst" @"suffimper.fst";
save stack imperstems.fst
clear stack;
```

Then, the composition of the long-distance dependency filters to the upper side of the transducer resulted from the concatenation operation is applied:

```
read regex [LONGDfilter1st .o. @"imperstems.fst"] |
[LONGDfilter2nd .o. @"imperstems.fst"] | [LONGDfilter3rd .o.
@"imperstems.fst"] ;
```

Notice that the union operation is used to union the three transducers resulted from composition. The three transducers hold the first, second,

and third person imperfective verbs respectively. The long-distance dependency filters are composed to the upper side of the transducer because the upper (lexical) side language of the traducer is the language in which the morphosyntactic feature strings are encoded.

In a similar way, the exceptional imperfective active prefixes (with the vowel *'u'*) are prefixed to the exceptional imperfective active stems (the imperfective active stems of measures II,III,IV, and QI), and the imperfective suffixes are suffixed to these exceptional imperfective active stems. According to the generalization (1) of Section 4.1, the prefixes of the imperfective passive stems are the same as the prefixes of the exceptional imperfective active stems ({Cu-}). This allows us to union the transducer holding the exceptional imperfective active stems with the transducer holding the imperfective passive stems before prefixing the exceptional prefix ({Cu-}) and suffixing the default suffixes which are suffixed without change to all the imperfective stems (active and passive stems).

```
read regex @"preimper2.fst" [@"IMPERFPASS.fst"|@"IMPERFACT2.fst"]
@"suffimper.fst";
save stack imperstems.fst
clear stack;
read regex [LONGDfilter1st .o. @"imperstems.fst"] |
[LONGDfilter2nd .o. @"imperstems.fst"] | [LONGDfilter3rd .o.
@"imperstems.fst"] ;.
```

Syncretisms are handled in this finite-state approach using the union operation. For example, the syncretism expressed by the suffix {*-naa*} which corresponds to the morphosyntactic feature lists :

Perf 1st Dual Mas, Perf 1st Plural Mas, Perf 1st Dual Fem, and Perf 1st Plural Fem

is handled through mapping the union of the multiple upper side morphosyntactic feature lists (strings):

```
[%+FIRST %+DUAL %+MAS] | [%+FIRST %+DUAL %+FEM] | [%+FIRST
%+PL %+MAS] | [%+FIRST %+PL %+FEM]
```

to one lower side string which is the suffix {*-naa*}:

```
[ [ [%+FIRST %+DUAL %+MAS] | [%+FIRST %+DUAL %+FEM] | [%+FIRST
%+PL %+MAS] | [%+FIRST %+PL %+FEM] ]:[%- n aa] ]
```

Finally, a lexicon transducer containing the inflected perfective, imperfective, and imperative verb forms is constructed by the union of the transducers holding the perfective, imperfective, and imperative inflectional stems affixed to their corresponding inflectional affixes. A replace rule is composed to the upper (lexical) side of the lexicon transducer to map the PNG feature tag used to handle long-distance-dependency to epsilon.

The upper (lexical) language of the produced lexicon transducer is a set of morphosyntactic feature strings which take a form like the following:

```
+KTB+FORM1+PERF+ACT+THIRD+SING+MAS
+KTB+FORM1+PERF+ACT+THIRD+SING+FEM
```

The lower (surface) language of the lexicon transducer is a set of inflected perfective, imperfective, and imperative verbs formed using verbal measures from roots. The lower side verbs which correspond to the upper side strings mentioned above are the following verbs respectively:

```
kataba
katabat
```

This transducer generates an inflected verb when it is supplied with a morphosyntactic feature string:

```
apply up +KTB+FORM1+PERF+ACT+THIRD+SING+FEM
```

The command `apply up` checks the upper (lexical) side language of the transducer to find the input feature string +KTB+FORM1+PERF+ACT+THIRD+SING+FEM. If this input is found, the corresponding lower (surface) side string is returned. On the other hand, when we use the command `apply down` with an inflected verb as an input, it checks the lower (surface) side language of the transducer to find the input verb. If this input is found, the corresponding upper (lexical) side string is returned:

```
apply down katabat
```

So, this bi-directional transducer can be used for generation and analysis.

What should be mentioned here is that some of the inflected verbs in the lower (surface) side language of the lexicon transducer are still abstract and they are not actually surface forms. They should undergo further phonological processes like rejection or transformation of radicals. These processes are normally applied when we deal with week roots. Besides, verbs which begin with a consonantal CC cluster undergo epenthesis. Such variations in word structure are handled using replace rules and composition.

5.3 Variation Rules

The variation rules are encoded using replace rules. A typical variation rule takes the following form:

```
A -> B || L _ R
```

This rule states that any upper sided string containing a string from A is related to a string or strings on the lower side containing a string from

B. This mapping is applied when the left context ends with L and the right context begins with R.

These variation replace rules are analogous to the phonological re-writing rules widely adopted in phonology. A replace rule (or more) is defined in order to handle each variation. The replace rules can be applied either in parallel or in a cascade sequential method. In a cascade application method, the output of a rule *feeds* another rule as its input. These rule are used to handle phonological and orthographic variations like epenthesis, transformation of radicals, and deletion of radicals. Handling such phonological and orthographic variations using our approach has been covered in three separate papers (Alnajem, 2004b; Alnajem, 2004c; Alnajem 2004d).

An example variation rule is the epenthetic rule which inserts the short syllable /?V/ before the beginning of a stem which begins with a CC consonantal cluster:

(4)

a- staktab (Measure X, Perf Act) → ***?i***staktab

b- staktib (Measure X, Imper Act/Imp) → ***?i***staktib

c- ktub (Measure I, Imper Act/Imp) → ***?u***ktub

You can notice that the vowel of the epenthetic syllable is either '*u*' or '*i*'. The quality of this vowel is governed by the nucleus of the next syllable. If the nucleus of the next syllable is the vowel '*u*', then the vowel of the epenthetic syllable will be '*u*'. Otherwise, it will be '*i*'. The epenthetic rule is defined in our approach using two replace rules as follows:

```
[..]  -> %@ u || .#. _ C C u  ;
[..]  -> %@ i || .#. _ C C [i|a]  ;
```

These rules prefix an epenthetic /?V/ syllable to the beginning of a stem. The beginning of the stem is indicated by the string boundary symbol '#' which is a special symbol indicating the beginning or the end of a string. This stem boundary is the left context of the rule. The CC consonant cluster and the vowel of the second syllable are in the right context of the rule. The rule states that a null symbol [..] at the beginning of the string is changed to an epenthetic syllable (/?V/) in the provided context. The symbol [..] is a special symbol used to indicate a null symbol within a string. The symbol '@' is an ASCII character representing the glottal stop '?'.

Thus, replace rules are used to handle phonological and orthographic variations in Arabic morphology. These replace rules are applied in cascade sequential method and/or in parallel, and they are composed to the upper side of the lexicon transducer. These rules are either obligatory or optional, which allows us to deal with optional variations like partial or absent vocalism.

6. Conclusion

This paper shows how finite-state technique can be used to handle the non-concatenative discontinuous nature of Arabic verbal inflection. Long-distance dependency between (discontinuous) inflectional affixes have been handled through the use of filters and composition. In the approach presented in this paper, the use of filters and composition to handle discontinuous morphemes stands for the use of feature unification applied in (Beesley, 1991). It also stands for the use of default inference applied in (Alnajem, 1998).

In addition, the paper shows how finite-state technique can be used to capture generalizations, sub-generalizations, and syncretisms governing Arabic verbal inflection. Such generalizations, sub-generalizations, and syncretisms have been handled using mathematical operations applied to finite-state transducers namely union, composition, and concatenation.

Capturing such generalizations, sub-generalizations, and syncretisms governing Arabic verbal inflection saves the computational approach from redundancy caused by ignoring such generalizations, sub-generalizations, and syncretisms. It also proves that Arabic verb inflection has a high level of regularity.

It has also been shown how finite-state technique can be used to handle phonological and orthographic variations in Arabic. These variations are caused by phonological and orthographic processes which are applied to lexical inflected verbs. This application yields surface inflected verb forms which are different than the lexical abstract forms. Such variations have been handled through the use of replace rules and composition.

References

Alnajem, Salah (1998). *Computational Approaches to Arabic Morphology*. PhD thesis, Essex University.

Alnajem, Salah (2004a). *A Finite-State Approach to Arabic Verbal Derivation*, Manuscript.

Alnajem, Salah, (2004b). A Computational Approach to the Variations in Arabic Verbal Orthography. Manuscript.

Alnajem, Salah. (2004c). Manhaj haasoubi lit taʕaamul maʕa ?isnaad al ?afʕaal ?ilad damaa?ir. Paper accepted for publication and to appear in the *Journal of the Gulf and Arabian Peninsula Studies*.

Alnajem, Salah. (2004d). A Computational Approach to Arabic Orthographic Relaxation. Paper accepted for publication and to appear in the *Arab Journal for the Humanities*.

Beesley, K. (1991). Computer Analysis of Arabic Morphology: A Two-Level Approach with Detours. In Comrie, B. and M. Eid (eds.), *Perspectives on Arabic Linguistics III: Papers from the Third Annual Symposium on Arabic Linguistics*, pp. 15572. Amsterdam, John Benjamin's Publishing Company.

Beesley, K. (1996). Arabic finite-state morphological analysis and generation, *COLING'96*, 1: 8994, Copenhagen.

Beesley, K. (1998). Arabic morphology using only finite-state operations. In Michael Rosner (ed.), *Computational Approaches to Semitic Languages: Proceedings of the Workshop*, pp. 507. Montreal, University of Montreal.

Bird, S and T. Ellison (1992). *One-level phonology: Autosegmental representations and rules as finite-state automata*. Technical report, Research Paper EUCCS/RP-51, University of Edinburgh.

Bird, S. and T. Ellison (1994). One-level phonology, *Computational Linguistics*, 20.1: 5590.

Cahill, Lynne (1991). *Syllable-based Morphology for Natural Language Processing*. A Doctoral Thesis, CSRP 181, University of Sussex.

Cahill, Lynne and Roger Evans (1990). An Application of DATR: The TIC Lexicon. In *Proceedings of the Ninth European Conference on Artificial Intelligence*, pp. 1205.

Gibbon, Dafydd (1990). Prosodic Association by Template Inheritance. In Daelemans, Walter and Gerald Gazdar (eds.), *Proceedings of the International Workshop on Inheritance and Natural Language Processing*, pp. 6581. Tilburg, Institute for Language Technology.

Katamba, Francis (1993). *Morphology*. Hampshire, Macmillan.

Kay, M. (1987). Nonconcatenative Finite-State Morphology. In *Proceedings of the Third Conference of the European Chapter of the Association for Computational Linguistics*, pp. 210, Copenhagen.

Kiraz, George (1994). Multi-tape Two-level Morphology: A Case Study in Semitic Non-linear Morphology. In *Proceedings of Coling 94*, pp. 1806.

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

Kornai, A. (1991). *Formal Phonology*. PhD thesis, Stanford University.

McCarthy, John (1981). A Prosodic Theory of Nonconcatenative Morphology, *Linguistic Inquiry*, 12: 373418.

McCarthy, John (1982). *Formal Problems in Semitic Phonology and Morphology*. A PhD thesis, Reproduced by the Indiana University Linguistics Club.

McCarthy, John (1992). Template Form in Prosodic Morphology. In Laurel Stvan et al. (eds.), *Papers from the Third Annual Meeting of the Formal Linguistics Society of Midamerica*, pp. 187218. Bloomington, Indiana University Linguistics Club.

McCarthy, John and Alan Prince (1990). Foot and Word in Prosodic Morphology: The Arabic Broken Plural, *Natural Language and Linguistic Theory*, 8: 20983.

Narayanan, A. and L. Hashem (1993). On abstract finite-state morphology. In *Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics*, pp. 297304.

Pulman, S. and M. Hepple (1993). A feature-based formalism for two-level phonology: a description and implementation, *Computer Speech and Language*, 7: 33358.

Reinhard, Sabine and Dafydd Gibbon (1991). Prosodic Inheritance and Morphological Generalisations. In *Proceedings of the Fifth Conference of the European Chapter of the Association for Computational Linguistics*, pp. 1316, Berlin.

Roche, E. and Yves Schabes (1997). Introduction. In Roche, E. and Yves Schabes (eds.), *Finite-State Language Processing*, pp. 166. Cambridge, MIT Press.

Trask, R. L (1993). *A Dictionary of Grammatical Terms in Linguistics*. London, Routledge.

Endnotes

1 The verbal measure I is associated with three active voice imperfective / imperative stems (patterns). Each active voice imperfective / imperative stem corresponds to specific perfective stems, and this correspondence is governed by some semantic and phonological conditions which will not be discussed here.

2 Similar to *fa9al*, the measure *fa9il* is associated with more than one active voice imperfective / imperative stem.

3 Thus, the imperfective inflection is achieved by circumfixation.

4 In such dependencies a (dependent) form is derived form another original form through applying changes to that original form.

5 The generalization is a statement about the facts of a language which holds true in all cases or in nearly all cases (Trask, 1993).

6 Analyzing the vowel of the imperfective prefix as a part of the imperfective pattern will not allow us to have one pattern (stem) for both the imperfective and imperative stems. Having one stem for both the imperfective and imperative stems is a universal generalization (Example: We *play* football (Present) / *Play* football (Imperative)).

7 This means that the vowel of the imperfective active prefix is *not constant*. It is either '*u*' or '*a*' according to the weight of the syllables of the imperfective active stem.

8 This approach is partially inspired by Beesley (1996, 1998).

9 In our computational approach, the vocalism tier is integrated in the skeletal tier (CV template).