A computational approach to Yorùbá morphology

Raphael Finkel
Department of Computer Science, University of Kentucky, USA
raphael@cs.uky.edu

Ọdẹ́túnjí Àjàdí, ỌDẸ́JỌBÍ
Cork Constraint Computation Center, University College Cork, Cork, Ireland.
t.odejobi@4c.ucc.ie

Abstract

We demonstrate the use of default default inheritance hierarchies to represent the morphology of Yorùbá verbs in the KATR formalism, treating inflectional exponences as markings associated with the application of rules by which complex word forms are deduced from simpler roots or stems. In particular, we suggest a scheme of slots that together make up a verb and show how each slot represents a subset of the morphosyntactic properties associated with the verb. We also show how we can account for the tonal aspects of Yorùbá, in particular, the tone associated with the emphatic ending. Our approach allows linguists to gain an appreciation for the structure of verbs, gives teachers a foundation for organizing lessons in morphology, and provides students a technique for generating forms of any verb.

1 Introduction

Recent research into the nature of morphology has demonstrated the feasibility of several approaches to the definition of a language's inflectional system. Central to these approaches is the notion of an inflectional paradigm. In general terms, the **inflectional paradigm** of a lexeme L can be regarded as a set of cells, where each cell is the pairing of L with a set of morphosyntactic properties, and each cell has a word form as its realization; for instance, the paradigm of the lexeme walk includes cells such as <WALK, {3rd singular present indicative}> and <WALK, {past}>, whose realizations are the word forms *walks* and *walked*.

Given this notion, one approach to the definition of a language's inflectional system is the **realizational** approach (Matthews 1972, Zwicky 1985, Anderson 1992, Corbett & Fraser 1993, Stump 2001); in this approach, each word form in a lexeme's paradigm is deduced from the lexical and morphosyntactic properties of the cell that it realizes by means of a system of morphological rules. For instance, the word form *walks* is deduced from the cell <WALK, {3rd singular present indicative}> by means of the rule of *-s* suffixation, which applies to the root *walk* of the lexeme WALK to express the property set {3rd singular present indicative}.

We apply the realizational approach to the study of Yorùbá verbs. Yorùbá, an Edekiri language of the Niger-Congo family (Gordon 2005), is the native language of more than 30 million people in West Africa. Although it has many dialects, all speakers can communicate effectively using Standard Yorùbá (SY), which is used in education, mass media and everyday communication (Adéwọ́lé 1988).

We represent our realizational analysis of SY in the KATR formalism (Finkel, Shen, Stump & Thesayi 2002). KATR is based on DATR, a formal language for representing lexical knowledge designed and implemented by Roger Evans and Gerald Gazdar (Evans & Gazdar 1989). Our information about SY is primarily due to the expertise of the second author.

This research is part of a larger effort aimed at elucidating the morphological structure of natural languages. In particular, we are interested in identifying the ways in which default-inheritance relations describe a language's morphology as well as the theoretical relevance of the traditional notion of principal parts. To this end, we have applied similar techniques to Hebrew (Finkel & Stump 2007), Latin (Finkel & Stump to appear, 2009b), and French (Finkel & Stump to appear, 2009a).

Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages – AfLaT 2009, pages 25–31,
Athens, Greece, 31 March 2009.

1.1 Benefits

As we demonstrate below, the realizational approach leads to a KATR theory that provides a clear picture of the morphology of SY verbs. Different audiences might find different aspects of it attractive.

- A **linguist** can peruse the theory to gain an appreciation for the structure of SY verbs, with all exceptional cases clearly marked either by morphophonological diacritics or by rules of sandhi, which are segregated from all the other rules.
- A **teacher** of the language can use the theory as a foundation for organizing lessons in morphology.
- A **student** of the language can suggest verb roots and use the theory to generate all the appropriate forms, instead of locating the right paradigm in a book and substituting consonants.

2 SY phonetics

SY has 18 consonants (*b, d, f, g, gb, h, j, k, l, m, n, p, r, s, ṣ, t, w, y*), 7 simple vowels (*a, e, ẹ, i, o, ọ, u*), 5 nasalized vowels (*an, en, in, ọn, un*), and 2 syllabic nasals (*m, n*). SY has 3 phonologically contrastive tones: High, Mid and Low. Phonetically, there are also two tone variants, rising and falling (Laniran & Clements 2003). SY orthography employs two transcription formats for these tones. In one format, the two tones are marked on one vowel. For example, the vowel *a* with a low tone followed by a high tone is written as *ǎ* and with a high tone followed by a low tone as *â*. This paper follows the alternative orthography, in which each tone is carried by exactly one vowel. We write *ǎ* as *àá* and *â* as *áà*.

3 A Realizational KATR theory for SY

The purpose of the KATR theory described here is to generate verb forms for SY, specifically, the realizations of all combinations of the morphosyntactic properties of tense (present, continuous, past, future), polarity (positive, negative), person (1, 2 older, 3 older, 2 not older, 3 not older), number (singular, plural), and strength (normal, emphatic). The combinations form a total of 160 **morphosyntactic property sets** (MPSs).

Our analysis posits that SY verbs consist of a sequence of morphological formatives, arranged in six slots:

- Person, which realizes the person and number but is also influenced by tense and polarity,
- Negator marker 1, which appears only in the negative, but is slightly influenced by person and number,
- Tense, which realizes the tense, influenced by polarity,
- Negator marker 2, which appears only in the negative, influenced by tense,
- Stem, which realizes the verb's lexeme,
- Ending, which appears only for emphatic verbs.

Unlike many other languages, SY does not distinguish conjugations of verbs, making its KATR theory simpler than ones for languages such as Latin and Hebrew. However, the tonality of SY adds a small amount of complexity.

A theory in KATR is a network of **nodes**. The network of nodes constituting SY verb morphology is very simple: every lexeme is represented by a node that specifies its stem and then refers to the node `Verb`. The node `Verb` refers to nodes for each of the slots. We use rules of Sandhi as a final step before emitting verb forms.

Each of the nodes in a theory houses a set of **rules**. We represent the verb *mún* 'take' by a node:

```
Take:
1     <stem> = m ún
2      = Verb
```

The node, named `Take`, has two rules, which we number for discussion purposes only. KATR syntax requires that a node be terminated by a single period (full stop), which we omit here. Our convention is to name the node for a lexeme by a capitalized English word (here `Take`) representing its meaning.

Rule 1 says that a query asking for the stem of this verb should produce a two-atom result containing `m` and `ún`. Rule 2 says that all other queries are to be referred to the `Verb` node, which we introduce below.

A **query** is a list of atoms, such as `<stem>` or `<normal positive past 3Older`

sg>, addressed to a node such as Take. In our theory, the atoms in queries either represent **morphological formatives** (such as stem) or **morphosyntactic properties** (such as 3Older and sg).

A query addressed to a given node is matched against all the rules housed at that node. A rule **matches** if all the atoms on its left-hand side match the atoms in the query. A rule can match even if its atoms do not exhaust the entire query. In the case of Take, the query <stem past> is matched by Rules 1 and 2; the query <positive past> is only matched by Rule 2.

Left-hand sides expressed with **path notation** (<pointed brackets>) only match if their atoms match an initial substring of the query. Left-hand sides expressed with **set notation** ({braces}) match if their atoms are all expressed, in whatever position, in the query. We usually use set notation for queries based on morphological formatives and morphosyntactic properties, where order is insignificant.

When several rules match, KATR picks the best match, that is, the one whose left-hand side "uses up" the most of the query. This choice embodies Pāṇini's principle, which entails that if two rules are applicable, the more restrictive rule applies, to the exclusion of the more general rule. We sometimes speak of a rule's **Pāṇini precedence**, which is the cardinality of its left-hand side. If a node in a KATR theory houses two applicable rules with the same Pāṇini precedence, we consider that theory malformed.

In our case, Rule 2 of Take only applies when Rule 1 does not apply, because Rule 1 is always a better match if it applies at all. Rule 2 is called a **default rule**, because it applies by default if no other rule applies. Default rules define a hierarchical relation among some of the nodes in a KATR theory.

KATR generates output based on queries directed to nodes representing individual lexemes. Since these nodes, such as Take, are not referred to by other nodes, they are called **leaves**, as opposed to nodes like Verb, which are called **internal nodes**. The KATR theory itself indicates the list of queries to be addressed to all leaves. Here is the output that KATR generates for several queries directed to the Take node.

normal,positive,present,1,sg
mo mún

normal,positive,present,1,pl
a mún

normal,positive,present,2Older,sg
ẹ mún

normal,positive,present,2Older,pl
ẹ mún

normal,positive,present,3Older,sg
wọ́n mún

normal,positive,present,3Older,pl
wọ́n mún

normal,positive,present,2NotOlder,sg
o mún

normal,positive,present,2NotOlder,pl
ẹ mún

normal,positive,present,3NotOlder,sg
ó mún

normal,positive,present,3NotOlder,pl
wọ́n mún

normal,positive,past,2NotOlder,sg
o ti mún

normal,positive,continuous,2NotOlder,sg
ò ńmún

normal,positive,future,2NotOlder,sg
o óò mún

normal,negative,present,2NotOlder,sg
o (k)ò mún

normal,negative,past,2NotOlder,sg
o (k)ò tî mún

normal,negative,continuous,2NotOlder,sg
o (k)ò mún

normal,negative,future,2NotOlder,sg
o (k)ò ní (kìóò) mún

emphatic,positive,present,2NotOlder,sg
o múnun

emphatic,positive,past,2NotOlder,sg
o ti múnun

The rule for Take illustrates the strategy we term **provisioning** (Finkel & Stump 2007): It provides information (here, the letters of the verb's stem) needed by a more general node (here, Verb).

3.1 The Verb node

We now turn to the Verb node, to which the Take node refers.

```
Verb:
1    {continuous negative} = <present
       negative>
2    {} = Person Negator1 Tense Negator2
       , "<stem>" Ending
```

Rule 1 of Verb reflects the continuous negative to the present negative, because they have identical forms.

Rule 2 is a default rule that composes the surface form by referring to a node for each slot except the stem. This rule directs an query that does not satisfy Rule 1 to each of the nodes mentioned. In this way, the theory computes values for each of the slots that represent the morphological formatives. The KATR phrase "<stem>" directs a new query to the original node (in our case, Take), which has provisioned information about the stem (in our case, m ún). The comma in the right-hand side of rule 2 is how we represent a word division; our post-processing removes ordinary spaces.

3.2 Auxiliary nodes

The Verb node invokes several auxiliary nodes to generate the surface forms for each slot.

```
Person:
1     {1 sg} = mo
2     {1 sg negative} = mi
3     {1 sg future} = m
4     {1 pl} = a
5     {2Older} = ẹ
6     {2Older continuous} = ẹ̀
7     {2Older continuous pl} = w ọ́n
8     {3Older positive !future} = w ọ́n
9     {3Older} = w ọn
10    {2NotOlder sg} = o
11    {2NotOlder pl} = ẹ
12    {2NotOlder continuous sg} = ò
13    {2NotOlder continuous pl} = ẹ̀
14    {3NotOlder} = ó
15    {3NotOlder negative sg} =
16    {3NotOlder future} = yí
17    {3NotOlder pl ++} = <3Older>
```

Generally, the Person slot depends on person and number, but it depends to a small extent on polarity and tense. For example, the exponence[1] of 1 sg is *m*, but it takes an additional vowel in the negative and the non-future positive. On the other hand, the exponence of 1 pl is always *a*. Rule 8 applies to tenses other than future, as marked by the notation !future; in the future, the more general Rule 9 applies. Rule 17 reflects any query involving 3NotOlder pl to the same node (Person) and 3Older forms, to which it is identical. The ++ notation increases the Pāṇini precedence of this rule so that it applies in preference to Rules 15 and 16, even if one of them should apply.

```
Negator1:
1     {negative} = , (k)ò
2     {negative 3NotOlder sg} = kò
3     {} =
```

The first negation slot introduces the exponence *ò* for negative forms (Rules 1 and 2) and the null exponence for positive forms. In most situations, this vowel starts a new word (represented by the comma), and careful speech may place an optional *k* before the vowel (represented by the parenthetical *k*); in 3NotOlder sg, this consonant is mandatory.

```
Tense:
1     {} =
2     {past} = , t i
3     {continuous positive} = , ń -
4     {future positive} = , óò
5     {future 1 sg positive} = , àá
6     {future 3NotOlder positive} =
         <future 3Older positive>
```

The Tense slot is usually empty, as indicated by Rule 1. However, for both negative and positive past, the word *ti* appears here. In the positive continuous, the following slot (the stem) is prefixed by *ń*. We use the hyphen (−) to remove the following word break by a spelling rule (shown later). Similarly, future positive forms have a tense marker, with a special form for 1 sg. As often happens, the 3NotOlder form reflects to the 3Older form.

```
Negator2:
1     {future negative} = , ní
2     {past negative} = ´  ì
3     {} =
```

The second negator slot adds the word *ní* in the future (Rule 1). In the past (Rule 2), it changes the tone of the tense slot from *ti* to *tî*. In all other cases, Rule 3 gives a null default. Rule 2 follows an assumption that tone and vowel can be specified independently in SY; without this assumption, this slot would be more cumbersome to specify. Such floating tones are in keeping with theories of autosegmental phonology (Goldsmith 1976) and are seen in other Niger-Congo languages, such as Bambara (Mountford 1983).

```
Ending:
1     {} =
2     {emphatic} = ↓
```

The Ending slot is generally null (Rule 1), but in emphatic forms, it reduplicates the final vowel with a mid tone, unless the vowel already has a

[1] An **exponence** is a surface form or part of a surface form, that is, the way a given lexeme appears when it is attached to morphosyntactic properties.

mid tone, in which case the tone becomes low. (We disagree slightly with Akinlabi and Liberman, who suggest that this suffix is in low tone except after a low tone, in which case it becomes mid (Akinlabi & Liberman 2000).) For this case, we introduce a *jer*[2], represented by "↓", for postprocessing in the Sandhi phase, discussed below. Such forms are important as a way to simplify presentation, covering many cases in one rule. When we tried to develop a SY KATR theory without a *jer*, we needed to separate the stem of each word into onset and coda so we could repeat the coda in emphatic forms, but we had no clear way to indicate the regular change in tone. The *jer* accomplishes both reduplication and tone change with a single, simple mechanism. It also suggests that the emphatic ending is really a matter of tone Sandhi, not a matter of default inheritance.

4 Postprocessing: Sandhi, Spelling and Alternatives

After the rules produce a surface form, we postprocess that form to account for Sandhi (language-specific rules dictating sound changes for euphony), spelling conventions, and alternative exponence. We have only one Sandhi matter to account for, the *jer* "↓". We accomplish this postprocessing with these rules:

```
1 #vars $vowel:  a e ẹ i o ọ u .
2 #vars $tone:   ´  ̀  .
3 #sandhi $vowel ↓ => $1 $1 ̀  .
4 #sandhi $vowel $tone ↓ => $1 $2 $1 .
5 #sandhi $vowel n ↓ => $1 n $1 ̀  n .
6 #sandhi $vowel $tone n ↓ => $1 $2 n $1
    n .
```

The first two lines introduce shorthands so we can match arbitrary vowels and tone marks. Sandhi rules are applied in order, although in this case, at most one of them will apply to any surface form.

Rules 3–6 represent tone Sandhi by showing how to replace intermediate surface strings with final surface strings. Each rule has a left-hand side that is to be replaced by the information on the right-hand side. Numbers like `$1` on the right-hand side refer to whatever a variable (in this case, the first variable) on the left-hand side has matched.

[2]A **jer**, also called a **morphophoneme**, is a phonological unit whose phonemic expression depends on its context. It is an intermediate surface form that is to be replaced in a context-sensitive way during postprocessing.

Rule 3 indicates that if we see a vowel without a tone mark (indicating mid tone) followed by the *jer*, we replace it with the vowel (represented by `$1`) repeated with low tone. This specification follows our assumption that tone and vowel may be treated independently. Rule 4 indicates that a vowel followed by a tone mark and the *jer* is repeated with mid tone (without a mark). Rules 5 and 6 are similar, but they deal with nasalized vowels.

There is one spelling rule to remove word breaks that would otherwise be present. We have used "–" to indicate that a word break should disappear. We use the following rule to enforce this strategy:

```
#sandhi - , => .
```

That is, a hyphen before a comma removes both.

SY allows the negative future forms *(k)ò ní* and *kò ní* to be expressed instead as *kìóò*. We provide rules of alternation for this purpose:

```
#alternative \(k\)ò , ní => kìóò .
#alternative kò , ní => kìóò .
```

These alternation rules effectively collapse the three slots, Negator1, Tense, and Negator2 into a single exponence.

5 Processing

The interested reader may see the entire SY theory and run it through our software by directing a browser to `http://www.cs.uky.edu/~raphael/KATR.html`, where theories for several other languages can also be found. Our software runs in several steps:

1. A Perl script converts the KATR theory into two files: a Prolog representation of the theory and a Perl script for post-processing.

2. A Prolog interpreter runs a query on the Prolog representation.

3. The Perl post-processing script treats the Prolog output.

4. Another Perl script either generates a textual output for direct viewing or HTML output for a browser.

This software is available from the first author under the GNU General Public License (GPL).

6 Discussion and Conclusions

This exercise demonstrates that the realizational approach to defining language morphology leads to an effective description of SY verbs. We have applied language-specific knowledge and insight to create a default inheritance hierarchy that captures the morphological structure of the language, with slots pertaining to different morphosyntactic properties. In particular, our KATR theory nicely accounts for the slot structure of SY verbs, even though most slots are dependent on multiple morphosyntactic properties, and we are easily able to deal with the tone shifts introduced by the emphatic suffix.

This work is not intended to directly address the problem of parsing, that is, converting surface forms to pairings of lexemes with morphosyntactic properties. We believe that our KATR theory for SY correctly covers all verb forms, but there may certainly be exceptional cases that do not follow the structures we have presented. Such cases are usually easy to account for by introducing information in the leaf node of such lexemes. Further, this work is not in the area of automated learning, so questions of precision and ability to deal with unseen data are not directly relevant.

We have constructed the SY theory in KATR instead of DATR for several reasons.

- We have a very fast KATR implementation, making for speedy prototyping and iterative improvement in morphological theories. This implementation is capable of taking standard DATR theories as well.

- KATR allows bracket notation ({ and }) on the left-hand side of rules, which makes it very easy to specify morphosyntactic properties for queries in any order and without mentioning those properties that are irrelevant to a given rule. Rules in DATR theories tend to have much more complicated left-hand sides, obscuring the morphological rules.

- KATR has a syntax for Sandhi that separates its computation, which we see as postprocessing of surface forms, from the application of morphological rules. It is possible to write rules for Sandhi in DATR, but the rules are both unpleasant to write and difficult to describe.

As we have noted elsewhere (Finkel & Stump 2007), writing KATR specifications requires considerable effort. Early choices color the structure of the resulting theory, and the author must often discard attempts and rethink how to represent the target morphology. The first author, along with Gregory Stump, has built KATR theories for verbs in Hebrew, Slovak, Polish, Spanish, Irish, Shughni (an Iranian language of the Pamir) and Lingala (a Bantu language of the Congo), as well as for parts of Hungarian, Sanskrit, and Pali.

Acknowledgments

We would like to thank Gregory Stump, the first author's collaborator in designing KATR and applying it to many languages. Lei Shen and Suresh Thesayi were instrumental in implementing our Java™ version of KATR. Nancy Snoke assisted in implementing our Perl/Prolog version.

Development of KATR was partially supported by the US National Science Foundation under Grants IIS-0097278 and IIS-0325063 and by the University of Kentucky Center for Computational Science. The second author is supported by Science Foundation Ireland Grant 05/IN/I886 and Marie Curie Grant MTKD-CT-2006-042563. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

Adéwọlé, L. O. (1988). *The categorical status and the function of the Yorùbá auxiliary verb with some structural analysis in GPSG*, PhD thesis, University of Edinburgh, Edinburgh.

Akinlabi, A. & Liberman, M. (2000). The tonal phonology of Yoruba clitics, *in* B. Gerlach & J. Grizenhout (eds), *Clitics in phonology, morphology and syntax*, John Benjamins Publishing Company, Amsterdam/Philadelphia, pp. 64–66.

Anderson, S. R. (1992). *A-morphous morphology*, Cambridge University Press.

Corbett, G. G. & Fraser, N. M. (1993). Network Morphology: A DATR account of Russian nominal inflection, *Journal of Linguistics* **29**: 113–142.

Evans, R. & Gazdar, G. (1989). Inference in DATR, *Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics*, Manchester, pp. 66–71.

Finkel, R., Shen, L., Stump, G. & Thesayi, S. (2002). KATR: A set-based extension of DATR, *Technical Report 346-02*, University of Kentucky Department of Computer Science, Lexington, KY. `ftp://ftp.cs.uky.edu/cs/techreports/346-02.pdf`.

Finkel, R. & Stump, G. (2007). A default inheritance hierarchy for computing Hebrew verb morphology, *Literary and Linguistic Computing* **22**(2): 117–136. `dx.doi.org/10.1093/llc/fqm004`.

Finkel, R. & Stump, G. (to appear, 2009a). Stem alternations and principal parts in French verb inflection, *Cascadilla Proceedings Project* .

Finkel, R. & Stump, G. (to appear, 2009b). What your teacher told you is true: Latin verbs have four principal parts, *Digital Humanities Quarterly* .

Goldsmith, J. A. (1976). *Autosegmental phonology*, PhD thesis, Massachusetts Institute of Technology, Boston, MA.

Gordon, R. G. (2005). *Ethnologue: Languages of the World*, 15^{th} edn, SIL International, Dallas, Texas.

Laniran, Y. O. & Clements, G. N. (2003). Downstep and high rising: interacting factors in Yorùbá tone production, *J. of Phonetics* **31**(2): 203 – 250.

Matthews, P. H. (1972). *Inflectional morphology*, Cambridge University Press.

Mountford, K. W. (1983). *Bambara declarative sentence intonation*, PhD thesis, Indiana University, Bloomington, IN.

Stump, G. T. (2001). *Inflectional morphology*, Cambridge University Press, Cambridge, England.

Zwicky, A. M. (1985). How to describe inflection, *Proceedings of the 11th annual meeting of the Berkeley Linguistics Society*, pp. 372–386.