Procesamiento del Lenguaje Natural, nº39 (2007), pp. 21-26 recibido 18-05-2007; aceptado 22-06-2007

A simple formalism for capturing order and co-occurrence in computational morphology

Mans Hulden
University of Arizona
Department of Linguistics
P.O. BOX 210028
Tucson AZ 85721-0028
USA
mhulden@email.arizona.edu

Shannon T. Bischoff
University of Arizona
Department of Linguistics
P.O. BOX 210028
Tucson AZ 85721-0028
USA
bischoff@email.arizona.edu

Resumen: Tradicionalmente, modelos computacionales de morfología y fonología han venido asumiendo, como punto de partida, un modelo morfotáctico donde los morfemas se extraen de subléxicos y se van concatenando de izquierda a derecha. El modelo de 'clase de continuación' se ha venido utilizando como el sistema estándar de facto en la creación de diferentes cajas de herramientas de software. Tras estudiar lenguas de tipología diversa, proponemos aquí un modelo de rasgos ampliado. Nuestro modelo consta de varias operaciones diseñadas con el fin de que un buen número de restrictiones de co-ocurrencia local y global puedan ser descritas de manera concisa. Aparte también sugerimos ciertas formas de implementar estos operadores en modelos de morfología basados en transductores de estado finito. Palabras clave: morfología computacional; morfotáctica, unificación de rasgos.

Palabras clave: morfología computacional, morfotáctica, unificación de rasgos.

Abstract: Computational models of morphology and phonology have traditionally assumed as a starting point a morphotactic model where morpehemes are drawn from sublexicons and concatenated left-to-right. In defining the lexicon-morphotactic level of a system, this 'continuation-class' model has been the de facto standard implementation in various software toolkits. From surveying of a number of typologically different languages, we propose a more comprehensive feature-driven model of morphotactics that provides the linguist with various operations that are designed to concisely define a variety of local and global co-occurrence restrictions. We also sketch ways to implement these operators in finite-state-transducer-based models of morphology.

Keywords: computational morphology, morphotactics, feature unification.

1. *Introduction*

Morphotactics—how morphemes combine together to make for well-formed words in languages—can, and is, often treated as an isolated problem in computational morphological analysis and generation. This has been particularly true of two-level and finite-state morphological models, where grammars describe a mapping from an abstract morphotactic level to a surface level. In such models, the topmost level is often described not only as a mapping to some lower level of representation, but is also separately constrained to reflect only legal combinations of morphemes in a language.

Insofar as morphotactics is seen to be a problem of expressing combinatorial constraints, it would be desirable to develop a formalism that would allow for simple descriptions of such constraints on combinations of morphemes as frequently occur in various natural languages. Such models have indeed been proposed. By far the most popular model in computational morphology has been the 'continuation class' model (Koskenniemi, 1983; Beesley and Karttunen, 2003) and variants thereof. The underlying assumption—and the reason for its popularity—is that a majority of languages exhibit the kind of morphotactics that is easily expressed through such systems: left-to-right concatenative models where the allowability of a morpheme is primarily conditioned by the preceding morpheme. This assumption does not always hold, however, which has led to many proposals and implementations that augment this model with extensions that provide for expressive power to include some phenomenon otherwise not capturable.

While a variety of such extensions to the continuation-class model have been proposed—some quite comprehensive—we depart entirely from the continuation-class model in this proposal, and instead propose a formalism that is based on declarative constraints over both the order and co-occurrence of individual morphemes.[1] This approach to restricting morphotactics takes advantage of a fairly restricted set of operations on feature-value combinations in morphemes. The formalism allows us express a variety of non-concatenative phenomena—complex co-occurrence patterns, free morpheme ordering, circumfixation, among others—concisely with a small number of statements.

2. Nonconcatenative phenomena

In the following, we give a few examples of nonconcatenative morphotactic phenomena that are difficult to capture with only a continuation-class model of morphotactics in order to motivate particular features of the notation we propose.[2]

2.1. Slot-and-filler morphotactics

The so-called slot-and-filler morphologies (also called templatic morphologies) tend to differ from concatenative processes or left-to-right agglutinative morphologies in that they feature abundant, often long-distance, restrictions on the co-occurrence of morphemes. An example of this type of language is *Navajo* (and other Athabaskan languages) where a strict template guides the order of morphemes. Some templatic slots may be empty, while others are obligatorily filled:

1	2	3	4	5	6	7	8
O	P	Obj.	In	Fut	S	Cl	Stem
ha	da	j	$\emptyset$	$\emptyset$	íí	$\emptyset$	geed
	Pl.	4p			4p		Imp.
‘out’							‘dig’

hadajííííged

‘Those guys dug them up’

In the above example, we have a template consisting of eight slots, where certain classes of morphemes are allowed to appear—slot 1 for ‘outer’ lexical prefixes, slot 2 for marking distributive plurals, etc.[3]

What is noteworthy is the complex co-occurrence constraints that govern the legal formation of *Navajo* verbs. To give a few examples with respect to the above templatic derivation: 1) the ‘outer’ prefix *ha* is allowed with stems that conjugate according to a certain pattern (the so-called yi-perfective), which *geed* fulfils; 2) the allomorph of the 4th person subject pronoun *íí* is selected on the basis of what slots 1 and 2 contain; 3) the 4th person subject pronoun is discontinous in that a *j* must also appear in slot 3—without this, the *íí* in slot 6 signals 3rd person; 4) the ‘classifier’ in slot 7 has four possibilities which *together* with the stem mode and prefixes in slots 1 and 2 determine what the subject allomorph can be.

Navajo is an extreme example of long-distance systematic patterns of co-occurrence restrictions. Some languages, such as the American Indian language *Koasati*, which features around 30 slots for its verbs, allow almost any co-occurrence pattern (Kimball, 1991). Nevertheless, a consise formalism for defining morphotactics needs to include the possibility of capturing easily the type of patterns *Navajo* and other similar languages have.

2.2. Free morpheme ordering

Although less documented among the world’s major languages, there also exists languages where certain classes of morphemes can appear in free relative order without affecting the semantics of a word. Recent examples of this include *Aymara*, an American Indian language spoken in the Andean

[1] The Xerox xfst/lexc (Beesley and Karttunen, 2003) toolkit is a particularly versatile toolkit that offers a variety of notational devices to capture the same phenomena we document here.

[2] We exclude two common patterns from this discussion: that of templatic root-and-pattern morphology (as seen in Arabic), as well as reduplication phenomena. These have been extensively treated in the literature and the most efficient solutions seem to treat these more as phonological phenomena not specified in the most abstract level of morphotactic description.

[3] This simplified model follows Faltz (1998); the majority of analyses for *Navajo* assume 16 slots or more. See Young (2000) for details.

region,[4] and *Chintang*, a Tibeto-Burman language, from which the following example is drawn:

(1) u-kha-ma-cop-yokt-e
3nsA-1nsP-NEG-see-NEG-PST
(2) u-ma-kha-cop-yokt-e
(3) kha-u-ma-cop-yokt-e
(4) ma-u-kha-cop-yokt-e
(5) kha-ma-u-cop-yokt-e
(6) ma-kha-u-cop-yokt-e
'They didn't see us'

(from Bickel et al. (2007))

Here, examples (1) through (6) are interchangeable and equally grammatical.

A concatenative model where order *must* be declared would require extra machinery to capture this phenomenon.[5] As will be seen below, we will want to capture this phenomenon by simply leaving certain order constraints *undeclared*, from which the free order falls out naturally.

3. Constraining morphotactics

Given these phenomena, we now propose a simple formalism to capture morphotactics. First, we assume the existence of labeled sublexicons containing various morphemes in a given class. Also, we assume that each morpheme can be associated with feature-value combinations:

Class$_1$	...	Class$_n$
Morpheme$_1$	...	Morpheme$_1$
{Subclass}	...	{Subclass}
OP Feat Value	...	OP Feat Value
⋮	⋮	⋮
Morpheme$_i$	...	Morpheme$_j$
OP Feat Value	...	OP Feat Value

That is, we assume that a complete lexicon is a collection of sublexicons (or classes) that contain morphemes. These morphemes may carry any number of feature-value pairs, to which an operator is associated, and may be a member of a subclass as well.

3.1. Order

In a fashion similar to that of the continuation-class model, we propose that morphemes are drawn out of this finite number of sublexicons (classes) one at a time. However, instead of each sublexicon consisting of a statement guiding the choice of the next sublexicon, the order is to be governed by a number of statements over the sublexicons using two operators: $>$ and $\gg$.

The operator $C_1 > C_2$ defines the patterns (languages) where each morpheme drawn out of the sublexicon named C_1 must immediately precede each morpheme drawn out of C_2. Likewise $C_1 \gg C_2$ illustrates the constraint that morphemes drawn from C_1 must precede (not necessarily immediately) those from C_2. For the sake of completeness, we can also assume the existence of the reverse variants $<$ and $\ll$.

In a templatic morphology, order constraints could simply be a single transitive statement $C_1 \gg \ldots \gg C_n$, and the majority of the grammar would consist of feature-based constraints regarding the possible *co-occurrence* of morphemes.

Likewise, the examples of free morpheme order are now easy to capture: let us suppose that there exists a number of prefixes that have free internal order (such as in the Chintang example above), C_1 to C_n, followed by a number of morphemes with strict internal ordering, $C_x \ldots C_y$. This could now be captured by the statements:

$$C_1 \ll C_x$$

$$\ldots$$

$$C_n \ll C_x$$

$$C_x \ll \ldots \ll C_y$$

When modeled in this fashion there need not be any separate statements saying that C_1 to C_n occur in free internal order—rather, this falls out of simply not specifying an order constraint for those morpheme classes, other than that they must occur before C_x.

3.2. Co-occurrence

For defining the possible co-occurrence of morphemes, we take advantage of the basic idea of features and feature unification. We do not assume elaborate feature structures to

[4] See Hardman (2001) for examples of the free morpheme ordering in Aymara. Thanks to Ken Beesley and Mike Maxwell for pointing out these resources and the phenomenon.

[5] Beesley and Karttunen (2003) hint at a solution that first declares a strict order with contination classes and subsequently 'shuffle' the morphemes freely with a regular expression operator that is composed after the output of the strictly ordered morphotactic level.

exist, rather we take unification to be an operator associated with features in the morpheme lexicon, such that conflicting feature-value pairs may not exist in the same word.

As mentioned, every morpheme in every sublexicon can carry OP [Feature Value] combinations, where OP is one of $\sqcup$, $+$, or $-$.

3.2.1. Unification

The 'unification' operator $\sqcup$ has the following semantics: a morpheme associated with $\sqcup[FX]$ disallows the presence of any other morpheme in the same word carrying a feature F and a value other than X.

3.2.2. Coercion

The operator $+$ control for co-ocurrence as follows: an $+[FX]$ combination associated with a morpheme requires that there be another $[FX]$ combination in the word somewhere else for the word to be legal.

3.2.3. Exclusion

Similarly, $-[FV]$ requires that any $[FV]$ combination be absent from the word in question.

For the sake of transparency, it is assumed that a $+[FV]$ statement can be satisfied by $\sqcup[FV]$.

3.3. Examples

With these tools of defining morphotactics, we can now outline an example from English derivational morphology using order constraints and the feature-related operators.

3.3.1. Order constraints

A well-known generalization of English is that derivational suffixes often change parts of speech, and so must attach to the proper part of speech that the preceding morpheme 'produces.' Also, prefixes and suffixes are seen to fall into two strata: an inner stratum of (mostly) latinate affixes (such as *ic* and *ity*, which attach closest to the stem, and an outer stratum of (mostly native) affixes (such as *ness* and *less*) (Mohanan, 1986). Assuming the stem *atom*, and a vocabulary of suffixes *ic, ity, ness* and *less*, we should be able to form *atom*, *atomic*, *atomicity*, *atomnessless*, among others, but not **atomity*, **atomlessity.*

```
Class {Stems}
atom {toN}

Class {LatinateSuffix}
ic {fromN}
   {toA}

ity {fromA}
    {toN}

Class {NativeSuffix}
ness  {fromN}
      {toN}

less {fromN}
     {toA}

Constraints

LatinateSuffix >> Stems
NativeSuffix >> LatinateSuffix | Stems
{fromN} > {toN}
{fromA} > {toA}
```

In the above notation (reflecting an actual implementation) *ic* belongs to the head class `LatinateSuffix` but also to `fromN` and `toA`, reflecting that the suffix is latinate and changes a noun into an adjective. The relevant constraints are that latinate suffixes must follow stems, and that nonlatinate suffixes must both follow stems and latinate suffixes. The above snippet suffices to capture the general order constraints with respect to the strata-based derivational view mentioned previously.

3.3.2. Feature constraints: circumfixes

Circumfixes are a classical simple case of co-occurrence that can be captured using the feature constraints. To continue with English, an example of a circumfix is the combination *em*+adjective+*en*, as in *embolden.* However, the suffix *en* can occur on its own, as in *redden*, while the prefix *em* cannot.[6] This can be modelled as follows:

```
Class {LatinatePrefix}

em
     +[Circ emen]

Class {Stems}

bold  {toA}

Class {NativeSuffix}
```

[6] The prefix *em* is actually modeled to be underlyingly *en* where the nasal assimilates in place to the following consonant.

```
en    {fromA}
       {toV}
      U[Circ emen]
```

Here, the prefix *em*, carries $+[Circ\ emen]$, requiring the presence of a feature-value pair $[Circ\ emen]$ somewhere else in the derivation. This can be satisfied by the suffix *en*. However, this suffix can also surface on its own since it does not carry the coercion $+$ operator on the feature-value pair, but only the unification operator. The interplay between these two operators yields the desired morphotactics.

4. Implementation

While we wish to remain somewhat agnostic as to the preferred computational models of morphological analysis and parsing, we shall here outline a possible implementation of the proposed formalism in terms of finite-state automata/transducers, since these are a popular mode of building morphological analyzers and generators.[7]

We assume the standard regular expression notations where Σ denotes the alphabet, $L_1 \cup L_2$ is the union of two languages, $\overline{L}$ is the complement of language L, # is an auxiliary boundary marker denoting a left or right edge of a string. Also, in our notation, symbol and language concatenation is implied whenever two symbols are placed adjacent to each other. Following this, our earlier notation $+[FV]$ denotes the language that consists of one string with five elements concatenated (we assume F and V to represent features and values, respectively, and $+$, $-$, [,], {, }, and $\sqcup$ to be single symbols).

4.1. Context restriction

As an auxiliary notation, we shall assume the presence of a regular expressions context-restriction operator ($\Rightarrow$) in the compilation of automata and transducers as this alleviates the task of defining many morphotactic restrictions. We take:

$$X \Rightarrow Y_1\_Z_1, \ldots, Y_n\_Z_n$$

to characterize the regular language where every instance of the language X is immediately preceded by the language Y_i and immediately followed by Z_i, for some i. The reader is urged to consult Yli-Jyrä and Koskenniemi (2004) for a very efficient method of compiling such statements into automata.

4.2. Unification

With the above, we can build $\sqcup[FV]$, for some feature-value combination present in our grammar, as:

$$\sqcup[FV] \Rightarrow$$
$$\#\overline{\Sigma^*(\sqcup \cup +)[F\overline{V}]\Sigma^*}\ \_\ \overline{\Sigma^*(\sqcup \cup +)[F\overline{V}]\Sigma^*}\#$$

That is, the presence of a $\sqcup[FV]$ is allowed only in the environment where both the left and right-hand sides do not contain a string $\sqcup[FV_x]$ such that V_x is not V and the operator preceding is either $+$ or $\sqcup$.

4.3. Coercion

Similarly, we can build the $+$ operator as follows:

$$+[FV] \Rightarrow \Sigma^*\_\ (\sqcup \cup +)[FV], \_\ (\sqcup \cup +)[FV]$$

Here, the statement implies that any presence of $+[FV]$ is allowed only if the string also contains a similar $[FV]$ somewhere to its left or right, where the operator is either $+$ or $\sqcup$.

4.4. Exclusion

The exclusion ($-$) operator is built similarly, as:

$$-[FV] \Rightarrow$$
$$\#\overline{\Sigma^*(\sqcup \cup +)[FV]\Sigma^*}\ \_\ \overline{\Sigma^*(\sqcup \cup +)[FV]\Sigma^*}\#$$

This defines the languages where an instance of some string $-[FV]$, where F and V are features and values, respectively, is allowed only if surrounded by strings that do not contain $[FV]$ with the operator either $+$ or $\sqcup$.

5. Order constraints

In order to address the compilation of the order constraints ($<$, $>$ and $\ll$, $\gg$), one would have to make assumptions about the exactly how the morphemes, features, values, and class labels are represented as automata. Supposing every morpheme is followed by

[7] A parser for Navajo verbal morphology has been built this way: converting the contents of a grammar into regular expressions, and then building automata that constrain the morphotactic level (Hulden and Bischoff, 2007).

its bundle of features, so that a word on the morphotactic level is represented as: $M_1\{Class\}op[F_1V_1]\ldots op[F_nV_n]M_2\{Class\}\ldots$, where *op* is one of $\sqcup, +, -$, the presence of a constraint $Class_1 \ll Class_2$ can be represented as:

$$\overline{\Sigma^*\{Class_2\}\Sigma^*\{Class_1\}\Sigma^*}$$

that is, the language where no instance of the string $Class_2$ precedes $Class_1$. The $\gg$ operator can be defined symmetrically.

The immediate precedence $Class_1 < Class_2$ can be defined as:

$$\overline{\Sigma^*\{Class_1\}\overline{\Sigma^*\{\Sigma^*}\{\overline{Class_2}\}\Sigma^*}$$

representing the language where no $Class_n$ string may intevene between a string $Class_1$ and $Class_2$. Note that the brackets { and } are single symbols in Σ in the above.

6. Conclusion

We have presented a formalism for specifying morphotactics that allows for separate description of morpheme order and morpheme co-occurrence. These are controlled by a small number of operators on features, or classes of morphemes. The order-related operators have the power to state that a class of morpheme must either precede, or immediately precede some other class of morphemes, while the co-occurrence operators allow for unification of feature-value pairs, exclusion of feature-value pairs, or coercion, i.e. expression of a demand that some feature-value pair be present.

We have also sketched a way to implement the formalism as finite-state automata through first converting the notation into regular expressions, which can then be compiled into automata or transducers using standard methods.

Bibliografía

Beesley, Kenneth and Lauri Karttunen. 2003. *Finite-State Morphology.* CSLI, Stanford.

Bickel, Balthasar, Goma Banjade, Martin Gaenszle, Elena Lieven, Netra Paudyal, Ichchha Purna Rai, Manoj Rai, Novel Kishor Rai, and Sabine Stoll. 2007. Free prefix ordering in chintang. *Language*, 83.

Faltz, Leonard M. 1998. *The Navajo Verb.* University of New Mexico Press.

Hardman, M. J. 2001. *Aymara: LINCOM Studies in Native American Linguistics.* LINCOM Europa, München.

Hulden, Mans and Shannon T. Bischoff. 2007. An experiment in computational parsing of the Navajo verb. *Coyote Papers: special issue dedicated to Navajo language studies*, 16.

Kimball, Geoffrey D. 1991. *Koasati Grammar.* Univ. of Nebraska Press, London.

Koskenniemi, Kimmo. 1983. *Two-level morphology: A general computational model for word-form recognition and production.* Publication 11, University of Helsinki, Department of General Linguistics, Helsinki.

Mohanan, Karuvannur P. 1986. *The theory of lexical phonology.* Reidel, Dordrecht.

Yli-Jyrä, Anssi and Kimmo Koskenniemi. 2004. Compiling contextual restrictions on strings into finite-state automata. *The Eindhoven FASTAR Days Proceedings.*

Young, Robert W. 2000. *The Navajo Verb System: An Overview.* University of New Mexico Press, Alburquerque.