

English Noun plurals: A Cyclic Account

Andrew van der Spuy

Department of Linguistics, School of Literature and Language Studies,
University of the Witwatersrand, PO Wits, 2050 South Africa

E-mail: andrew.vanderspuy@wits.ac.za

Abstract

This article describes Cyclic Morphology, a theory of morphological generation that falls into the category of theories which Stump (2001) calls lexical–realizational. An account of the morphology of English noun plurals is given in order to illustrate the workings of the theory. Technical terms in the theory are explained and exemplified. It is shown why the theory can be classified as lexical, and argued that a lexical theory is to be preferred over an inferential one: first, it allows all morphological generation to take place in the lexicon, thus avoiding the problem of accounting for derivation that takes place after inflection; and second, a lexical theory requires a grammar with fewer components, and may be more economical than an inferential theory.

Keywords: noun plurals, morphology, morphological theory, lexical theories, realizational theories

1. Introduction

Stump (2001) classifies theories of morphology according to two distinctions: theories may be incremental or realizational, and they may be lexical or inferential. In an incremental theory, morphemes add meaning (e.g. [number plural]) to a basic form, as bricks put together constitute a wall. In a realizational theory, morphemes express or realize abstract meaning, as a house realizes an architect’s plans, without being those plans. In a lexical theory, morphemes are items in the lexicon, and both derivation and inflection take place before a word is inserted into the syntax. In an inferential theory, morphemes are added by morphological rule; inflection takes place outside the lexicon, either in a separate morphological component, or in the syntax. This allows four possible kinds of theory, which can be presented in the form of a table:

Table 1. Stump’s typology of morphological theories, with examples

	Incremental	Realizational
Lexical	Lieber (1981, 1992); Selkirk (1982)	Distributed Morphology (Halle and Marantz 1993); Cyclic Morphology
Inferential	Articulated Morphology (Steele 1995)	Extended Word-and- Paradigm Theory (Anderson 1992); Paradigm Function Morphology (Stump 2001)

Stump classifies Halle and Marantz's (1993) theory of Distributed Morphology as 'lexical-realizational'. Halle and Marantz actually envisage morphological composition as taking place in both the Vocabulary (their equivalent of the lexicon)¹ and the Morphological Structure, an "added level" which "is the interface between syntax and phonology" (Halle and Marantz 1993:114).

Stump argues that realizational theories are to be preferred over incremental ones for two reasons. Firstly, incremental theories do not account for extended exponence – the expression of a single morphological feature by more than one affix (or morpheme) (Matthews 1974). An example is the following locative form of a Zulu noun:

- (1) e-thaf-eni
 locative-veld-locative
 "In/to the veld."

As can be seen, the feature [locative] is marked twice.

Secondly, incremental theories do not account for underdetermination. Underdetermination can be defined as a failure of any morphological concomitant of a particular feature to appear where expected (cf. Stump 2001:7–8). For example, Zulu nouns are divided into classes according to the set of agreement morphemes associated with each (Canonici 1990). In most cases, the class of the noun is marked by a prefix on the noun. The noun meaning "veld" in (1) belongs to class 5, but in this case there is no (overt) prefix.

Stump (2001) also presents three reasons why inferential theories are to be preferred over lexical ones. In this article, however, a lexical–realizational theory of inflectional morphology, called Cyclic Morphology (CM), will be presented. This theory circumvents Stump's objections to lexical theories, owing to the fact that it is a unificatory theory. This will be discussed in detail in section 7 of this article.

The structure of the article is as follows. Section 2 briefly explains the inability of incremental theories to account for extended exponence and underdetermination. Section 3 presents the essentials of CM. Section 4 demonstrates how CM can apply to a familiar and relatively simple kind of morphology like the regular English plural. Section 5 discusses certain aspects of the theory: its lexical nature, reasons for preferring a lexical theory, the algorithm by which it generates word-forms, and the principle by which selection is ordered. Section 6 shows how the theory accounts for irregular English plurals, and section 7 argues that the unificatory nature of CM allows it to avoid the problems that Stump claims must arise from lexical theories. Section 8 examines the question of how economical the theory is. Section 9, the concluding section, suggests directions for future research.

2. Stump's criticisms of incremental theories

Incremental theories of morphology work on the assumption that affixes have features which contribute to the features of the word which is being formed (Lieber 1981, 1992; Selkirk 1982). However, as pointed out by Stump (2001), there are two problems with such theories.

¹ The term "lexicon" is rejected as too vague (Halle and Marantz 1993:113).

First, they do not account for the phenomenon of extended exponence which is found in many of the languages of the world. Second, they do not account for underdetermination.

2.1. Extended exponence

Examples of extended exponence can be found in many languages, but it is debatable whether they occur in English. Spencer (1991:51) gives as an example the word *written*, because the form of the root /rɪt/ occurs only in the perfect participle, and therefore can be taken as an exponent of the feature [perfect] (in addition to the suffix /ən/). However, many morphologists would prefer to claim that /rɪt/ is the allomorph of *write* that occurs in the perfect, but is not itself an exponent of [perfect]. In CM, the fact that a form occurs in the context of certain morphological features means that it is an exponent of those features: therefore Spencer's analysis will be assumed here to be a valid example of extended exponence. Given this assumption, a similar English example is the word *children*, where the feature [plural] is expressed both by the allomorphic form of the root /tʃɪldr/, and by the suffix /ən/.

- (2) tʃɪldr-ən
 child.pl-pl²
 “children”

The root allomorph /tʃɪldr/ is assumed in this example to reflect both the semantics of the basic root /tʃɑːld/ and the feature [number pl]. The suffix /ən/ also reflects the meaning [number pl], which means that this feature is reflected twice. An incremental theory would predict that extended exponence should not occur. In such a theory, when a morpheme is added to the word, it contributes a feature to the word as a whole, rather than reflecting features of a more abstract construct. Once a feature has been added to a word in this way, there is no need for morphemes which further exemplify that feature. When a value for a particular feature has been assigned, then the principle of “least effort” (Chomsky 1995) would require that no further morphemes having that feature should be added to the word. As Lieber says (1992:106) in an analysis of verb inflection in Vogul:

The Tense/Aspect (T/A) markers must attach first to the verb stem The values for [Pres[ent]] and [Pret[erite]] will percolate to the categorical signature Note that a second T/A morpheme is blocked from attaching now, since there is no longer any room in the categorical signature for its T/A values to percolate to.

That is, an item cannot be marked for a feature which it already has. Yet extended exponence does exist in natural languages, and in fact, it is fairly common. How, then, can we account for this phenomenon? Section 6 of this article will show how CM does indeed account for extended exponence.

² The following abbreviations and symbols are used in this article: A – adjective; C – consonant; CM – Cyclic Morphology; ffv – final fricative voicing; N – noun; PFM – Paradigm Function Morphology; pl – plural; sg – singular; V – verb/vowel; vd – voiced; ~ – is an allomorph of; • – new cycle; ⊃ – unifies with, is extended by; → – resolves as; — – vacuous application of a cycle.

2.2. Underdetermination

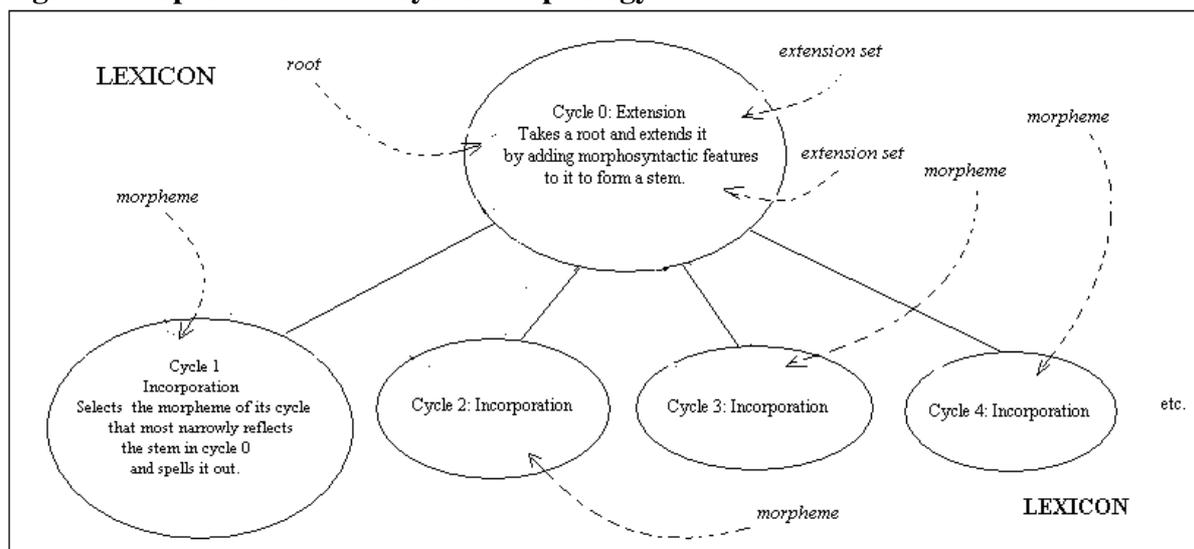
Stump's second criticism of incremental theories is that they fail to account for underdetermination (2001:7). An example of underdetermination is observable in English noun morphology, where the feature [number sg] is never marked by any corresponding overt affix. Incremental models of morphology would predict that this feature should be marked on all nouns, as it could not otherwise be added to the basic root; the fact that underdetermination exists indicates that incremental theories are wrong in this regard. Of course, it could be assumed that features that are not overtly marked are marked by zero morphemes; however, following Pullum and Zwicky (1991), certain theories do not postulate zero morphemes: these include Anderson's Extended Word-and-Paradigm Theory (1992), Stump's Paradigm Function Morphology (2001) (PFM), and CM.

It could also be assumed that all count nouns are marked [number sg] in the lexicon, and that the plural morpheme is feature-changing; the plural morpheme would therefore be a derivational affix. This would lead to problems in a complex morphology like Latin, however, where many oppositions contain an underdetermined member. Treating all these as derivational would considerably undermine the usefulness of the "derivational/inflectional" distinction. If it is assumed, in an incremental theory, that number is an inflectional feature in English, then both singular and plural would need to be morphologically marked.

3. A cyclic, lexical-realizational theory of morphology

The theory of CM consists of a theory of the structure of the lexicon, coupled with a generative algorithm which ensures that correct word-forms are produced. The essentials of the theory are probably best conveyed in a diagram – see Figure 1. In the explanation that follows (and in the remainder of the paper), technical terms are presented in bold *italics* when first introduced, and explained as soon as possible thereafter. A complete glossary is included in the Appendix.

Figure 1. Representation of Cyclic Morphology



In the preliminary generative cycle, labelled cycle 0, a *root* is selected from the lexicon. A root is assumed to be a set of semantic and morphosyntactic features. In cycle 0, the root is

extended by means of *extension sets*; that is, it unifies with sets of appropriate morphosyntactic features (e.g. number features for an English noun; number and case features for a Latin noun; tense, aspect, voice, person and number features for an English verb). The extended root is the *stem*.

Once the stem has been formed, the *incorporating* cycles begin. These are labelled cycle 1, cycle 2, etc. Each incorporating cycle selects from the lexicon a morpheme that *reflects* the stem (that is, has features in common with it), and *spells out* the morpheme. Different languages and different word-types differ in the number of incorporating cycles they require. For example, whereas a simple English plural noun requires two cycles (one to spell out the root, and one for the inflection), an agglutinating language like isiZulu might require six or seven cycles to generate its verbs.

Because roots, extension sets and morphemes are all contained in the lexicon, the generative process is assumed to take place in the lexicon as well. Although this article describes only inflectional morphology, the theory accounts for derivational morphology too. This will be examined in a future article.

The only ordering principle is *Pāṇini's Principle*: a *narrower* match must be chosen before one that is less narrow (Anderson 1969; Kiparsky 1973). "Narrower" is defined as follows: if two sets X and Y both match a set Z, X is the narrower match if it has more features matching features of Z than Y does; or if it has fewer *variable features* matching features of Z than Y does. A variable feature is a feature whose value is a variable, as [pl X], or a set of alternatives, as [number sg|pl]. If several sets are equally narrow matches, then one is chosen at random.

4. An example of how the theory works: regular English plurals

This section provides a practical illustration of how the theory works, deploying regular English plurals as examples. Consider the regular noun *dog*. The fragmentary lexicon in (3) generates the singular and plural forms of this word, which itself serves as a generative model for thousands of other regular nouns. Some other relevant items also appear in the lexicon.

- (3)
- (a) [lexeme *dog*, category N, proper -, count +, semantics 'dog']³
 - (b) [lexeme *z*, category N, number pl, cycle 2]
 - (c) [category N, number sg|pl]
 - (d) [lexeme X, category N, cycle 1]
 - (e) [category N, count -, number sg]
 - (f) [lexeme *lʌk*, category N, proper -, count -, semantics 'luck']
 - (g) [lexeme X, category N, number pl, pl X, cycle 2]

As is the case in most theories, the lexicon is assumed to be an unordered list of items: for this reason, the items in (3) are presented in a random order. Lexical items consist of sets of semantic or morphosyntactic features, or both; they fall into three categories: roots, extension

³ Morpheme boundaries are not indicated in the notation, because the forms which are the values of the attributes [lexeme] and [~] are intended to represent entire morphemes. If some part of a morpheme is not relevant, for instance in the 'final fricative voicing' allomorph discussed in examples (23), (24), it is represented by a cover symbol X or Y.

sets, and morphemes. These are distinguished by their internal characteristics, and not by any external designation.

A well-formed root (e.g. (3a)) contains at least the following features: [lexeme] (whose value is a phonological representation), [category] (whose value is a lexical category, e.g. N, V, A), and [semantics] (whose value is given as a word in inverted commas, but is assumed to be the complex feature structure that makes up the concept associated with the word in question). A root lacks a feature [cycle].

A morpheme (e.g. (3b)), like a root, has a feature [category], but it also has a feature [cycle], whose value is an integer greater than or equal to 1. Most morphemes have a feature [lexeme], but they usually lack a feature [semantics]. Rather, the meanings they reflect are those of morphosyntactic features like [number pl] or [tense past].

An extension set (e.g. (3c)) is any set that has some features in common with the root or stem, and no features that *contradict* it. A feature is said to contradict another feature if they both have the same attribute but different values, for example [number sg] and [number pl]. Furthermore, an extension set must have at least one feature (other than [cycle]) that the root or stem lacks – this is a feature that can *extend* it. Extension sets lack the features [semantics] and [cycle], and they usually lack the feature [lexeme].

The process of generation is presented in (4) below. Lines are designated (a)-(d) for ease of reference.

- (4) (a) • [lexeme dɒg, category N, proper -, count +, semantics ‘dog’]
 ∪ [category N, count +, number sg|pl]
 (b) [lexeme dɒg, category N, proper -, count +, number pl]
 (c) • [lexeme X, category N, cycle 1] → [lexeme dɒg, category N, cycle 1] /dɒg/
 (d) • [lexeme z, category N, number pl, cycle 2] /z/

The symbol • at the beginning of line (4a) shows the commencement of a cycle. The first cycle in the process of generation is called cycle 0 (see Figure 1). First, a root is selected from the lexicon. In this case, (3a) [lexeme dɒg, category N, proper -, count +, semantics ‘dog’] is chosen. This, the item to be generated, is called the *generand*. Then the lexicon is scoured for sets which may *extend* the generand, that is, add appropriate features to it. An extension set must have features in common with the generand, and no features that *contradict* it. The symbol ∪ is used to show the process of extension, which is essentially a process of set unification. Table 2 shows why most of the sets in the fragmentary lexicon (3) cannot be extension sets for the generand.

Table 2. Generand: [lexeme *dog*, category N, proper -, count +, semantics ‘dog’]

Set	Is it a possible extension set?	Reason
[lexeme <i>dog</i> , category N, proper -, count +, semantics ‘dog’]	No	As it is the generand, it has all its features in common with the generand. However, it has no features that can extend it. A set can therefore never be an extension set for itself.
[lexeme <i>z</i> , category N, number pl, cycle 2]	No	Its feature [lexeme <i>z</i>] contradicts [lexeme <i>dog</i>] in the generand.
[category N, number sg pl]	Yes	Its feature [category N] matches the generand, and the generand does not have a feature [number sg pl].
[lexeme X, category N, cycle 1]	No	Its feature [lexeme X] matches [lexeme <i>dog</i>], because X is a variable over all possible values; its feature [category N] matches the generand; but the only remaining feature is [cycle 1], which may not be used to extend a root.
[category N, count -, number sg]	No	Its feature [count -] contradicts the feature [count +] in the generand.
[lexeme <i>lk</i> , category N, proper -, count -, semantics ‘luck’]	No	Its features [lexeme <i>lk</i>], [count -], and [semantics ‘luck’] all contradict the corresponding features in the generand. A single contradictory feature is sufficient to disqualify an item from being an extension set.
[lexeme X, category N, number pl, pl X, cycle 2]	No	A set with a feature [cycle] cannot be an extension set.

Only one set in the fragmentary lexicon is a possible extension set for the generand, namely (3c) [category N, number sg|pl]. This set is now unified with the generand. The feature [number sg|pl] is added to the generand. The value [sg|pl] is read “either singular or plural”: when a feature like this one with alternative values is added to a generand, one of the alternatives must be chosen, according to the meaning that the speaker wishes to express. In this case, the choice is [number pl], and the generand becomes [lexeme *dog*, category N, proper -, count +, semantics ‘dog’, number pl]. The new form of the generand appears at the beginning of line (4b). A root that has been extended in this way is called a *stem*.

As there are no further sets that can extend the generand, the next stage is *incorporation*. The lexicon is scoured for items with the feature [cycle] which *reflect* the generand. A set X reflects another set Y if X has a feature [cycle], and every feature of X (except [lexeme] and [cycle]) matches some feature of Y. Therefore (3g) [lexeme X, category N, number pl, pl X, cycle 2] is not a candidate for incorporation, as its feature [pl X] does not reflect any feature in the generand. (This set generates irregular plural morphemes – see section 6.) There are two sets in (3) that meet the conditions mentioned, namely (3b) [lexeme *z*, category N, number pl, cycle 2] and (3d) [lexeme X, category N, cycle 1].

(3d) has an *earlier* [cycle] value than (3b) does, and so it becomes the generand in a new cycle. As it contains a variable feature, [lexeme X], it must first be *resolved*. That is, the variable must be replaced by a constant value that reflects the corresponding feature in the *matrix generand* (the generand in cycle 0). This process of resolution is indicated by an arrow in line (4c), repeated here as (5):

(5) [lexeme X, category N, cycle 1] → [lexeme dog, category N, cycle 1] /**dog**/.

Next, the generand is spelt out as /dog/ (shown by the phonological representation in bold at the end of line (4c)/(5)). Cycle 1 is complete. Processing returns to the incorporation phase of the matrix cycle, cycle 0: now the other compatible item, (3b), is selected (line (4d)). A morpheme cannot be further extended, nor can it incorporate any other item, so it is spelt out as /z/. The word /dogz/ is complete.

The generation of the words *cats* and *dishes* will be very similar to that of *dogs*. The following items are required in the lexicon:

- (6) (a) [lexeme kæt, category N, proper -, count +, semantics ‘cat’]
 (b) [lexeme dɪʃ, category N, proper -, count +, semantics ‘dish’]

Cats is generated as follows:

- (7) • [lexeme kæt, category N, proper -, count +, semantics ‘cat’]
 ∪ [category N, number sg|pl]
 [lexeme kæt, category N, proper -, count +, semantics ‘cat’, number pl]
 • [lexeme X, category N, cycle 1] → [lexeme kæt, category N, cycle 1] /**kæt**/
 • [lexeme z, category N, number pl, cycle 2] /**z**/

It will be seen that the steps followed here are exactly the same as those followed in the generation of /dogz/ in (4) above. The form produced in (7) is /kætz/, which then undergoes the phonological rule that devoices a final obstruent after a voiceless obstruent, to produce the form /kæts/. The generation of the word *dishes* is similar. At spell-out, the form /dɪʃz/ is produced; a phonological rule of schwa-insertion converts this to /dɪʃəz/. In CM, it is assumed that morphology and phonology operate separately, the former serving as input for the latter. There are some exceptions to this: these arise when the incorporation of a particular morpheme brings about the replacement of one value of [lexeme] with another, or when the resolution process assigns a constant value to a feature [lexeme] which has a variable value (see (18), (19), (20) and (23), (24) below.) Such replacement is strictly morphophonological rather than phonological.

The examples above have shown how the plurals of count nouns are generated. A non-count noun like *luck* is generated as follows in (8):

- (8) (a) • [lexeme λk , category N, proper -, count -, semantics ‘luck’]
 \cup [category N, count -, number sg]
 (b) [lexeme λk , category N, proper -, count +, semantics ‘luck’, number sg]
 (c) • [lexeme X, category N, cycle 1] \rightarrow [lexeme λk , category N, cycle 1] / λk /
 (d) • [cycle 2] —

Once again, the choice of possible extension sets can be summarised in table form:

Table 3. Generand [lexeme λk , category N, proper -, count -, semantics ‘luck’]

Set	Is it a possible extension set?	Reason
[lexeme dog , category N, proper -, count +, semantics ‘dog’]	No	Its feature [lexeme dog] contradicts [lexeme λk] in the generand.
[lexeme z , category N, number pl, cycle 2]	No	Its feature [lexeme z] contradicts [lexeme λk] in the generand.
[category N, number sg pl]	Yes	Its feature [category N] matches the generand, and the generand does not have a feature [number sg pl].
[lexeme X, category N, cycle 1]	No	A set with a feature [cycle] cannot be an extension set.
[category N, count -, number sg]	Yes	Its features [category N] and [count -] match the generand, and the generand does not have a feature [number sg].
[lexeme λk , category N, proper -, count -, semantics ‘luck’]	No	As explained in (Table 2), a set cannot be an extension set for itself.
[lexeme X, category N, number pl, pl X, cycle 2]	No	A set with a feature [cycle] cannot be an extension set.

The root [lexeme λk] can be expanded by either of two extension sets, namely (3c) [category N, number sg|pl] and (3e) [category N, count -, number sg]. At this point a choice has to be made as to which of them to select. (3e) is a *narrower* match for the generand than (3c), because it has two features in common with it, while (3c) has only one feature in common with it; so, by Pāṇini’s Principle, (3e) applies first. The generand is expanded to [lexeme λk , category N, proper -, count -, semantics ‘luck’, number sg] (line (8b)). Because the generand is now marked for the feature [number], (3c) is no longer an extension set for it, as the only feature it could have contributed is [number sg|pl]. The process of incorporation now begins. The only morpheme in the lexicon that reflects the generand is (3d) [lexeme X, category N, cycle 1]. This resolves to (3d) [lexeme λk , category N, cycle 1] (line (8c)) and spells out as / λk /.

There is no morpheme which can be incorporated in cycle 2: there are two [cycle 2] morphemes in the lexicon, namely [lexeme z , category N, number pl, cycle 2] and [lexeme X, category N, number pl, pl X, cycle 2], but their feature [number pl] contradicts the feature [number sg] of the generand. Therefore cycle 2 applies vacuously. This is symbolised by “—” in the table (line (8d)). It must be stressed that “—” does not represent a zero morpheme. Zero morphemes, in theories which postulate them, are phonologically null entities which

nevertheless have morphosyntactic features associated with them. In CM, there are no zero morphemes: the vacuous application of a cycle simply means that nothing happens in that cycle. There is no ‘entity’, nor are there any morphosyntactic features, in the cycle at any point.

5. Discussion of some aspects of the theory

This section discusses certain salient aspects of the theory. First, in section 5.1, it is argued that the theory is lexical; section 5.2 suggests that lexical theories have certain advantages over other theories; the algorithm that drives generation is presented and explained in section 5.3; and finally, in section 5.4, the criterion for selecting one extension set or one morpheme before another is examined.

5.1. The lexical nature of the theory

An anonymous reviewer of an earlier, and somewhat different, draft of this article observed that items do not necessarily form part of the lexicon simply because of a linguist’s decree. Supporting evidence must always be provided for any assertion that a particular set of items is part of the lexicon. What evidence can be given, then, for the assertion that affixal morphemes and extension sets, as well as roots, are all part of the lexicon? (The status of roots as lexemes is, of course, considered to be non-controversial.) Affixes are assumed to be lexical in several theories, e.g. those of Lieber (1981, 1992), Selkirk (1982) and Scalise (1984). In CM it is assumed that if an item has a phonological form and a set of semantic or morphosyntactic features, then it is a lexeme. (This is, of course, a standard assumption: cf. Chomsky 1965:87; Halle and Marantz 1993:113). The vast majority of morphemes meet this criterion, although in some morphemes the value of the feature [lexeme] may be a variable: that is, its phonological form is dependent on that of other items, e.g. [lexeme X, category N, number pl, pl X], which can be realized as /ən/, /ɪm/ or /tə/, depending on circumstances (see (13), (14), (25), (26) and (27)). A small minority of morphemes have no [lexeme] feature, and thus no phonological form: there are no examples in the data discussed here, but they are postulated for languages like Swahili in order to account for affixes that may appear in different cycles, as either subject or object markers: this will be discussed in a future article. Such items are not null morphemes, however, because they acquire a feature [lexeme] by extension.

Extension sets are considered to be part of the lexicon on the following grounds. First, some extension sets capture generalizations about the lexical features of items, thus keeping the lexicon as succinct as possible. An example used in this article is (3e) [category N, count -, number sg], which captures the generalization that non-count nouns are treated as singular in English. Indeed, it is precisely because such extension sets make generalizations about the lexicon that they are presumed to be part of the lexicon.

Second, those sets which add optional features to items, or present a choice of features, may compete with features for which an item is already marked. For example, English has a number of *pluralia tantum* nouns like *trousers*, *scissors*; these will be marked [number pl] in their lexical entries, as

- (9) [lexeme trauzər, category N, proper -, count +, number pl]

thus blocking the addition of the feature [number sg|pl] by merger with the extension set (3c) [category N, number sg|pl]. Because extension sets like (3c) may compete with lexically stipulated features, they are also presumably part of the lexicon.

Third, the optional features introduced by some extension sets, like the feature [(pl ar)] in the example of *radii/radiuses* in (29) below, are paralleled by features that are lexically marked on some items, like [pl ən] on [lexeme oks]. Such sets can therefore be regarded as conveying lexical information about the lexemes which they extend, and so they can also be regarded as part of the lexicon.

As has been argued here, some extension sets are part of the lexicon. By Ockham's Razor, it can be assumed that all extension sets are lexemes.

The set types which constitute the input to the theory's generative algorithm, namely roots, morphemes and extension sets, all have a claim to being part of the lexicon. A simple inference is that generation itself takes place in the lexicon.

5.2. Reasons for preferring a lexical theory

There are at least two arguments for choosing a lexical theory of morphology over an inferential one. First, Lieber (1981) argues that inflectional morphology is located in the lexicon because some derivational forms are based on inflected forms. For example, German noun compounds can be formed out of inflected stems, nominalised verbs in Old English were derived from inflected (non-present) stems, and there are "a number of cases of derivation from [inflected] verb stems in Latin and Tagalog" (1981:7). Booij (1994) presents similar evidence of post-inflectional derivation. A theory of morphology which assumes that inflectional morphological processing takes place outside the lexicon would have to address this point.

A second reason for choosing a lexical theory is that it could provide a less complex model of grammar. A grammar with fewer components is conceptually simpler than one with more components, provided that the overall complexity of the theory is no greater than that of theories where lexicon and morphology are separated. In CM, as explained in section 5.2, there is no need to postulate a separate morphological component: morphemes and all the rules that apply to them are stored in the lexicon. The question of whether this makes the grammar simpler overall will be addressed in section 8 of this article.

5.3. The generative algorithm

As explained in section 3, the generation of a word-form takes place in two phases. In the first phase, *extension*, morphosyntactic features are added to an item by unification with extension sets. In the second phase, *incorporation*, morphemes that reflect the extended item are selected and spelt out. The two phases can be combined into a single recursive generative algorithm, as described in (10) below.

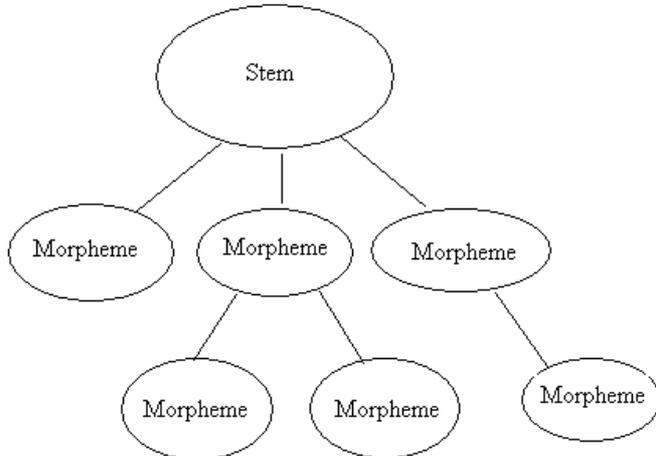
(10) *The generative algorithm*

- (a) *Selection*. Choose a root from the lexicon, or accept a morpheme from the matrix cycle. This is the generand.
- (b) *Resolution*. If the generand contains variable features, resolve it.
- (c) *Extension*. If the generand is a root, extend it until no further extension is possible. The resulting form is a stem.
- (d) *Incorporation*. Identify all morphemes that reflect the generand. Select the earliest of these morphemes, X. X becomes the generand in a new cycle. Apply the Generative Algorithm to X.
- (e) *Spell-out*. If no morpheme has been selected in step (d), spell out the value of the generand's feature [lexeme].
- (f) *Reiteration*. Repeat steps (d)-(f) until no further incorporation is possible and all selected morphemes have been spelt out.

The only ordering principle is *Pāṇini's Principle*, as explained in section 3.

The advantage of combining the two phases into a single recursive generative algorithm is that such a recursive algorithm allows incorporated cycles to generate other cycles in a hierarchical structure, should this prove necessary. Such a hierarchical structure is illustrated in Figure 2. (Hierarchical structures like these are not necessary in English, but some authors have claimed that they do occur in natural language – see the discussion in section 7.)

Figure 2. Hierarchical structure



The various steps of the algorithm have all been illustrated in section 4 above. The instruction at the beginning of (10d), “Identify all morphemes that reflect the generand”, will normally be taken to apply only if the generand is a stem in cycle 0. However, if a language allows structures like that in Figure 2, this would have to be modified.

The process of extension requires the unification of sets, and the process of incorporation requires that the incorporated morpheme should reflect the generand. This makes CM a unification grammar (cf. Sag et al. 1986).

5.4. Ordering

5.4.1. Ordering of cycles and incorporated morphemes

Morphemes are marked with a feature [cycle], which takes a numerical value. Morphemes with the same [cycle] value are mutually exclusive. The relative order of cycles is determined by the numerical value of the [cycle] feature: a morpheme with a lower (“earlier”) value precedes one with a higher value. When several morphemes with the same [cycle] value compete for incorporation into a particular word, choice is made according to Pāṇini’s Principle: the one which most narrowly reflects the generand is incorporated. This ensures that, in each cycle, the morpheme that expresses as many of the features as possible is selected. (If in a particular cycle no affix expresses any of the features of the generand, then the cycle applies vacuously.) Once a morpheme is selected, a new cycle begins and the selected morpheme becomes the generand in the new cycle.

5.4.2. Ordering of extension sets

When several extension sets can potentially combine with a generand, choice among them is again controlled by Pāṇini’s Principle.

6. The theory applied to irregular English plurals.

As demonstrated in section 4, the theory can account for the inflections of regular English nouns. A more challenging test, of course, is whether the theory will correctly produce the plurals of irregular nouns. This section, looking specifically at the irregular plurals *oxen*, *children*, *geese*, *sheep*, *leaves*, *cherubim*, and *radii* demonstrates how these plurals are created.

To generate the word *oxen*, the lexicon requires the following additional item:

- (11) [lexeme oks, category N, proper -, count +, pl ən, semantics ‘ox’]

This lexeme is selected as the generand. Its feature [pl ən] is a diacritic feature that marks it as irregular. Like [lexeme dog], it is extended by (3c), and becomes [lexeme oks, category N, proper -, count +, pl ən, semantics ‘ox’, number pl]. No further extension is possible, and so incorporation begins. The following three sets reflect the generand:

- (12) (a) [lexeme X, category N, cycle 1] (= (3d))
 (b) [lexeme z, category N, number pl, cycle 2] (= (3b))
 (c) [lexeme X, category N, number pl, pl X, cycle 2] (= (3g))

There is only one [cycle 1] morpheme, namely (12a): this is selected, and resolves as [lexeme oks, category N, cycle 1]. It spells out as /ɒks/.

There are two [cycle 2] morphemes that reflect the generand; of these, (12c) has a feature [pl X] which matches the generand’s special diacritic feature [pl ən] (X being a variable over all possible values). The fact that it has more features that match the generand than (12b) does makes it a narrower match for the generand, and, by Pāṇini’s Principle, it is the one that must

be chosen. Two of its features have the variable value X, so it must be resolved: as the variable in the feature [lexeme X] is also found in the feature [pl X], the former must obtain its value from the latter. The feature [pl X] copies its value from the matrix generand, and resolves as [pl ən]. The feature [lexeme X] then resolves as [lexeme ən]. No other sets can be incorporated into this morpheme, and so it spells out as /ən/. The word /ɒksən/ is complete. Note that the regular plural morpheme /z/ is blocked from selection by Pāṇini's Principle.

- (13) • [lexeme ɒks, category N, proper -, count +, pl ən, semantics 'ox']
 ∪ [category N, number sg|pl]
 [lexeme ɒks, category N, proper -, count +, pl ən, semantics 'ox', number pl]
 • [lexeme X, category N, cycle 1] → [lexeme ɒks, category N, cycle 1] /**ɒks**/
 • [lexeme X, category N, number pl, pl X, cycle 2]
 → [lexeme ən, category N, number pl, pl ən, cycle 2] /**ən**/

The plural form *children* can be analysed in several ways. It could be assumed that the plural allomorph of the root *child* is /tʃɪld/, and the plural affix is /rən/. Alternatively, the root allomorph could be taken as /tʃɪldr/, with the plural suffix being /ən/. The latter analysis is adopted here because it is more general, allowing for three lexical items (/tʃaɪld/, /ɒks/, and in religious contexts, /brɛðr/ ~ /brʌðər/) that take the plural /ən/, rather than two that take /ən/ and one that takes /rən/.

In order to generate *children* the following lexical items are required:

- (14) (a) [lexeme tʃaɪld, category N, proper -, count +, pl ən, semantics 'child']
 (b) [lexeme tʃɪldr, category N, number pl, ~ tʃaɪld, cycle 1]

As with *oxen*, the generand (14a) is selected, and extended to [lexeme tʃaɪld, category N, proper -, count +, pl ən, semantics 'child', number pl]. In this case there are four items that can potentially be incorporated. They are:

- (15) (a) [lexeme tʃɪldr, category N, number pl, ~ tʃaɪld, cycle 1] (= (14b))
 (b) [lexeme X, category N, cycle 1] (= (3d))
 (c) [lexeme X, category N, number pl, pl X, cycle 2] (= (3g))
 (d) [lexeme z, category N, number pl, cycle 2] (= (3b))

In cycle 1, (15a) is a narrower match for the generand than (15b), because a feature [~ X] is deemed to reflect a feature [lexeme X] in the generand, and it therefore has more features that match the generand. (15a) spells out as /tʃɪldr/. Then, as with *oxen*, (15c) is selected in cycle 2, is resolved as [lexeme ən], and spells out as /ən/.

- (16) • [lexeme tʃaɪld, category N, proper -, count +, pl ən, semantics 'child']
 ∪ [category N, number sg|pl]
 [lexeme tʃaɪld, category N, proper -, count +, pl ən, semantics 'child', number pl]
 • [lexeme tʃɪldr, category N, number pl, ~ tʃaɪld, cycle 1] /**tʃɪldr**/
 • [lexeme X, category N, number pl, pl X, cycle 2]
 → [lexeme ən, category N, number pl, pl ən, cycle 2] /**ən**/

We have seen two kinds of irregular plural, one where the plural morpheme is irregular (*oxen*), and one where both the plural morpheme and the root are irregular (*children*).

We now turn to a third kind, where the root has an irregular allomorph and there is no additional plural marker. An example of this is *goose*, with its plural *geese*. In this instance, the following lexical items are required:

- (17) (a) [lexeme gus, category N, proper -, count +, umlaut +, semantics ‘goose’]
 (b) [lexeme XiC, category N, umlaut +, number pl, ~ XVC, cycle [1, 2]]

The feature [umlaut +] of (17a) is a diacritic feature that marks it as an irregular root. The feature [cycle [1, 2]] of (17b) means that this morpheme straddles two cycles: it can be called a *multiple-cycle morpheme*. It enters into the selection process in cycle 1: if selected, it blocks any other morpheme from being selected in cycle 2. Stump (2001:141) refers to morphemes generated by more than one rule block (his equivalent of cycles) as belonging to “portmanteau rule blocks”. Morphemes like this occur in several languages.

(17a) is extended as [lexeme gus, category N, proper -, count +, umlaut +, semantics ‘goose’, number pl]. No further extension is possible, so incorporation begins. The following lexemes can potentially be incorporated:

- (18) (a) [lexeme X, category N, cycle 1] (= (3d))
 (b) [lexeme XiC, category N, umlaut +, number pl, ~ XVC, cycle [1, 2]] (= (17b))
 (c) [lexeme z, category N, number pl, cycle 2] (= (3b))

(18b) falls into both cycle 1 and cycle 2. Being the narrowest match, it is selected and spelt out as /gis/. Because it is marked [cycle [1, 2]], nothing further can be incorporated into the matrix generand and generation of the word /gis/ ‘geese’ is complete.

- (19) [lexeme gus, category N, proper -, count +, umlaut +, semantics ‘goose’]
 ∪ [category N, number sglpl]
 [lexeme gus, category N, proper -, count +, semantics ‘goose’, number pl]
 • [lexeme XiC, category N, umlaut +, number pl, ~ XVC, cycle [1, 2]]
 → [lexeme gis, category N, umlaut +, number pl, ~ gus, cycle [1, 2]] /gis/

The words *feet* and *teeth* are generated in a way similar to *geese*, as are the words *mice* and *lice*, except that the latter two will incorporate a morpheme (20e). Lexical entries pertaining to these words are given below:

- (20) (a) [lexeme fʊt, category N, proper -, count +, umlaut +, semantics ‘foot’]
 (b) [lexeme tuθ, category N, proper -, count +, umlaut +, semantics ‘tooth’]
 (c) [lexeme maʊs, category N, proper -, count +, umlaut +, semantics ‘mouse’]
 (d) [lexeme laʊs, category N, proper -, count +, umlaut +, semantics ‘louse’]
 (e) [lexeme Xaɪs, category N, umlaut +, number pl, ~ Xaʊs, cycle [1, 2]]

In the generation of /maɪs/ or /laɪs/, (20e) will be chosen for incorporation over (18b), although they have the same number of features, because the feature [\sim Xaʊs] of (20e) contains more constant features than the feature [\sim XVC] of (18b).

Words like *sheep*, where the plural is unmarked, are accounted for by including the diacritic feature [pl unmarked] in their lexical entry. A special lexeme, (21b), reflects such roots.

- (21) (a) [lexeme *ʃip*, category N, proper -, count +, pl unmarked, semantics ‘sheep’]
 (b) [lexeme X, category N, number pl, pl unmarked, cycle [1, 2]]

Sheep is generated as follows:

- (22) • [lexeme *ʃip*, category N, proper -, count +, pl unmarked, semantics ‘sheep’]
 ∪ [category N, number sg|pl]
 [lexeme *ʃip*, category N, proper -, count +, pl unmarked, semantics ‘sheep’, number pl]
 • [lexeme X, category N, number pl, pl unmarked, cycle [1, 2]]
 → [lexeme *ʃip*, category N, number pl, pl unmarked, cycle [1, 2]] /**ʃip**/

The special morpheme (21b) is selected in cycle 1, because it is the narrowest match for the generand. As it is marked [cycle [1, 2]], it blocks any other [cycle 2] morpheme from being selected. Words similarly marked include *salmon*, *grouse*, *deer*; and *buffalo*, *elk* and *elephant* in a hunting context.

Words like *leaf* are also specially marked, with a diacritic feature [ffv +] (“final fricative voicing”). A lexeme may be so marked only if it ends on a fricative consonant.

- (23) (a) [lexeme *lif*, category N, proper -, count +, ffv +, semantics ‘leaf’]
 (b) [lexeme XC_[vd +], category N, ffv +, number pl, \sim XC, cycle 1]
- (24) [lexeme *lif*, category N, proper -, count +, ffv +, semantics ‘leaf’]
 ∪ [category N, count +, number sg|pl]
 [lexeme *lif*, category N, proper -, count +, ffv +, semantics ‘leaf’, number pl]
 • [lexeme XC_[vd +], category N, ffv +, number pl, \sim XC, cycle 1]
 → [lexeme *liv*, category N, ffv +, number pl, \sim lif, cycle 1] /**liv**/
 • [lexeme *z*, category N, number pl, cycle 2] /**z**/

In cycle 1, (23b) is selected over (3d) [lexeme X, category N, cycle 1] because it reflects the generand more narrowly. C is a variable standing for “consonant”. In the resolution process, the [\sim] feature’s value [XC] is replaced with the value /lif/; then the [lexeme] feature replaces it with a corresponding sequence ending in a voiced consonant, that is, /liv/. A number of words follow the same pattern as *leaf*, including *sheaf*, *shelf*, *half*, *path*, *wreath*, *house*, and, in some varieties, *roof*; but not *giraffe*, *wraith* or *moose*. The former would be marked [ffv +], the latter not.

English has some words which have alternate plurals, like the Hebrew borrowing *cherub*, which forms its plural as either *cherubs* or *cherubim*.

- (25) (a) [lexeme tʃerəb, category N, proper -, count +, ex Hebrew, semantics ‘cherub’]
 (b) [lexeme X, category N, number pl, pl X, cycle 2] (= (3g))
 (c) [category N, ex Hebrew, number pl, (pl ɪm)]

The feature [ex Hebrew] is a diacritic feature meaning “derived from Hebrew” (Latin *ex* “from”). The optional rule (25c) accounts for the Hebrew plural /ɪm/. If a speaker applies the rule, then *cherub* must take item (25b) as its plural-marking affix, as shown in (26); otherwise, item (3b) [lexeme z, category N, number pl, cycle 2] will be chosen, as in the generation of *dogs* (shown in (4)).

- (26) • [lexeme tʃerəb, category N, proper -, count +, ex Hebrew, semantics ‘cherub’]
 ∪ [category N, number sg|pl]
 [lexeme tʃerəb, category N, proper -, count +, ex Hebrew, semantics ‘cherub’, number pl]
 ∪ [category N, ex Hebrew, number pl, (pl ɪm)]
 [lexeme tʃerəb, category N, proper -, count +, ex Hebrew, number pl, pl ɪm]
 • [lexeme X, category N, cycle 1] → [lexeme tʃerəb, category N, cycle 1] /tʃerəb/
 • [lexeme X, category N, number pl, pl X, cycle 2]
 → [lexeme ɪm, category N, number pl, pl ɪm, cycle 2] /ɪm/

Similar rules apply to words like the Greek-derived *stomata*. The following lexical items are required:

- (27) [lexeme stəʊmə, category N, proper -, count +, ex Greek, semantics ‘stoma’]
 [lexeme Xə, category N, ex Greek, (pl tə)]

It will be seen that these parallel the Hebrew example given above, and generation takes place in the same way.

Somewhat more complex than *cherub* are borrowings like *radius* (from Latin).

- (28) (a) [lexeme reɪdrəs, category N, proper -, count +, ex Latin, declension 2, semantics ‘radius’]
 (b) [lexeme Xəs, category N, proper -, count +, ex Latin, declension 2, (pl ar)]
 (c) [lexeme X, category N, number pl, pl aɪ, ~ Xəs, cycle 1]

Extension set (28b) allows items like *radius* to optionally add the feature [pl ar]. If this option is chosen then, in cycle 1, the morpheme (28c) is selected. The feature [~ Xəs] copies the [lexeme] feature of the generand to become [~ reɪdrəs]. The feature [lexeme X] then resolves as [lexeme reɪdɪ], and generation proceeds as for *cherub*.

- (29) • [lexeme reɪdrəs, category N, proper -, count +, ex Latin, declension 2, semantics ‘radius’]
 ∪ [category N, number sg|pl]
 [lexeme reɪdrəs, category N, proper -, count +, ex Latin, declension 2, semantics ‘radius’
 number pl] ∪ [category N, proper -, count +, ex Latin, declension 2, (pl ar)]
 [lexeme reɪdrəs, category N, proper -, count +, ex Latin, declension 2, number pl, pl ar]

- [lexeme X, category N, number pl, pl ar, ~ Xəs, cycle 1]
→ [lexeme reɪdɪ, category N, number pl, pl ar, ~ reɪdɪəs, cycle 1] / **reɪdɪ**/
- [lexeme X, category N, number pl, pl X, cycle 2]
→ [lexeme ar, category N, number pl, pl aɪ, cycle 2] / **ar**/

If the optional rule is not applied, then *radius* forms its plural in the same way as *dishes* above, generating *radiuses*:

- (30) [lexeme reɪdɪəs, category N, proper -, count +, ex Latin, declension 2]
 ∪ [category N, number sg|pl]
 [lexeme reɪdɪəs, category N, proper -, count +, ex Latin, declension 2, number pl]
- [lexeme X, category N, cycle 1] → [lexeme reɪdɪəs, category N, cycle 1] / **reɪdɪəs**/
 - [lexeme z, category N, number pl, cycle 2] / **z**/

False Latinate plurals, like **octopi*, can be accounted for as follows. Speakers who use such plurals have presumably generalized the optional rule, omitting the diacritic feature [declension 2], which shows that the rule should apply to only a subset of Latin nouns ending in /əs/, as in (31). This overgeneralization allows them to apply the rule to any Latin noun ending in /əs/.

- (31) [lexeme Xəs, category N, proper -, count +, ex Latin, (pl ar)]

Other foreign plurals can be accounted for similarly to *cherubim*, *stomata* and *radii*.

These examples show that the theory can generate a diverse range of morphological phenomena in a consistent and succinct way. Because morphemes are seen as the *expression* of features, rather than the *bearers* or *contributors* of features, the theory can account for extended exponence. This is witnessed in words such as *children* and *leaves*, in both of which the root is modified to express [number pl], and a plural affix is added. It also accounts for underdetermination, for example in words like *sheep*, where the plural form simply lacks an affix, by lexical stipulation.

7. How the theory avoids the problems of a lexical theory

As explained in section 5, CM is a lexical theory. Stump (2001) identifies certain problems that are intrinsic to lexical theories; but CM manages to avoid these problems. It will be argued in this section that this is due to its being a unification grammar. The three problems that Stump identifies are, first, that lexical theories necessitate a distinction between concatenative and nonconcatenative morphology, but that this distinction is theoretically unmotivated (2001:9); second, that in a lexical theory criteria for assigning features to affixes are arbitrary (2001:10); and third, that in a lexical theory the structure of words is assumed to be hierarchical, although there is no evidence for this (2001:11–12). Each of these will be discussed in turn below.

By “nonconcatenative morphology”, Stump means the kind of alternation seen in *goose/geese* or *man/men*. In CM, the basic root, e.g. [lexeme gus] is listed in the lexicon, and assigned a diacritic feature (in this case [umlaut +]) to ensure that it behaves differently from regular nouns. Both the regular and the irregular plural morphemes are also listed in the lexicon, in

this case as [lexeme z, category N, number pl, cycle 2] and [lexeme XiC, category N, umlaut +, number pl, ~ XVC, cycle [1, 2]] respectively. In terms of the unificatory design of CM, the plural stem of [lexeme gus] will incorporate the irregular [lexeme XiC], this form being chosen above others by Pāṇini's Principle. As it is a multiple-cycle morpheme, it will block the incorporation of the regular plural morpheme. This example shows that CM makes no distinction between concatenative and nonconcatenative morphology: the two kinds compete on equal terms, as Stump argues they should.

Stump's second objection to lexical theories, that criteria for assigning features to affixes are arbitrary, also does not hold true for CM. Unlike the kind of theory which Stump criticizes, affixes in CM do not subcategorize for roots and stems; rather, in terms of the unificatory design, affixes are selected according to the number of features they have in common with the stem. Therefore, morphemes have to be marked with morphosyntactic features, as they do not have subcategorization properties. In this theory, then, affixes are assigned features according to the following principle:

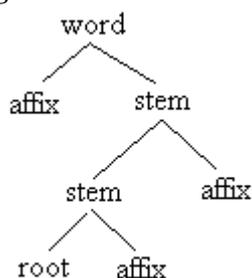
- (32) The features of a morpheme X are all the features common to the environments in which X occurs.

For example, the morpheme [lexeme z] occurs in a wide range of environments, whose only common features are [number pl], and the fact that it always appears in the second cycle of a word: thus its features are [lexeme z, number pl, cycle 2].

The form /tʃɪldr/ occurs only in the plural of /tʃɑɪld/, in cycle 1; its features are therefore [lexeme tʃɪldr, number pl, ~ tʃɑɪld, cycle 1].

Stump's third objection is that a lexical theory would imply that morphological structures are hierarchical, so that in a language where several affixes are attached to a root or stem the structure of a word would be as follows:

Figure 3. Word structure



The hierarchical design would follow from the idea that affixes subcategorize for roots and stems. Janda (1983) and Anderson (1992) have argued that there is no empirical evidence for such a hierarchical morphological structure in any language; but Lieber (1981:51–52) and Hudson (2007:82–83) give examples from Latin, which show that the inflectional morphology of that language seems to have a hierarchical structure.

In CM, the incorporation phase of the generation of a word-form selects each morpheme with direct reference to the abstract stem generated in cycle 0, thereby producing a flat structure, as

shown in figure 1. However, CM also allows some flexibility here. If it were to be convincingly shown that some language had a definite hierarchical structure to its words, this could be produced, as shown, for example, in figure 2. The recursive algorithm leaves open the possibility of hierarchical structure, without dictating it.

CM, despite being a lexical theory, manages to avoid the problems which Stump ascribes to lexical theories. It has been shown here that this is due to its unificatory design as seen specifically in its process of incorporation.

8. The comparative economy of the theory

As mentioned in section 5.2, a lexical theory is potentially more economical than an inferential theory, as it has fewer components. However, this would also depend on its having no other complexities to offset this. In this section, CM is compared to Stump's (2001) inferential–realizational Paradigm Function Morphology (PFM) in terms of complexity. It is argued that it is at least as economical as this theory.

As explained in section 4, CM postulates a lexicon which comprises three kinds of item, namely, roots, extension sets, and morphemes. In addition, it requires the generative algorithm described in section 5.3. This algorithm has six steps, namely selection, resolution, extension, incorporation, spell-out and reiteration. Extension and incorporation are moderated by Pāṇini's Principle.

In PFM, roots are contained in the lexicon, but are processed outside it. (This is not explicitly stated, but it is strongly implied by Stump's distinction between lexical and inferential theories (2001:1), and his classification of his own theory as inferential (2001:32).) Regarding the various kinds of rule postulated, Stump (2001:28) states that PFM "presumes the existence of several different rule types. Chief among these are paradigm functions, realization rules and morphological metageneralizations." Realization rules are of two types: rules of exponence and rules of referral (Stump 2001:36).

The function of selection in CM is performed by the paradigm functions in PFM. The paradigm functions are roughly equivalent in their effects to Cyclic Morphology's extension sets, and rules of exponence are equivalent to incorporation of morphemes, together with spell-out. The function of morphological metageneralizations is to make generalizations about rules (Stump 2001:47–50). In CM, such generalizations are achieved by variable features of morphemes. Rules of referral capture generalizations where a set of forms with a particular property consistently serves as the form for a different property (for example, in isiZulu, the majority of class 4 agreement morphemes are identical to class 9 agreement morphemes). In CM, such generalizations are also achieved by variable features. In PFM, rules are grouped into blocks. CM's equivalent is the feature [cycle], coupled with reiteration. These parallels can be summarized in table form as follows:

Table 4. Paradigm Function Morphology versus Cyclic Morphology

Paradigm Function Morphology	Cyclic Morphology
Paradigm functions	Selection
Paradigm functions	Extension sets
Realization rules	
Rules of exponence	Incorporation; spell-out
Rules of referral	Variable features
Morphological metageneralizations	Variable features
Rule blocks	Reiteration; the [cycle] feature.
Pāṇini's Principle	Pāṇini's Principle

If types of lexeme are counted as rule types, CM has seven rule types but requires just one component (the lexicon), whereas PFM has six rule types, but requires two components (the lexicon, and extra-lexical processing). Nonetheless, CM can, apparently, achieve the same results as PFM and appears to be as economical as PFM. However, this is only a sketch. A detailed comparison would not only require a careful evaluation of the relative complexity of the different rules, but could also involve empirical testing. Tests would include, for example, building computer programs which generate words according to the principles of PFM and CM, and measuring their respective sizes and speeds. It would also require comparing their generative powers over a wide range of languages and morphological complexities.

9. Conclusion

This article has presented a description of Cyclic Morphology, a lexical-realizational theory of morphology which, as a realizational theory should, accounts for extended exponence and underdetermination. This particular lexical model does not exemplify the three problems of lexical theories identified by Stump: nonconcatenative morphology is included in the lexicon, features are assigned to affixes in a principled way, and it does not postulate a hierarchical model of composition.

There are still many questions to be answered. For example, can the theory explain the highly complex morphology of languages like Yavapai (Lieber 1992), Tagalog (Anderson 1992) or Georgian (Anderson 1982, 1992)? It certainly accounts for various kinds of inflectional morphology, but in this article nothing has been said about derivational morphology. How does it account for compounding and other forms of derivation?

A question that has been addressed only cursorily is whether the theory is more economical than other theories, e.g. Stump's (2001) Paradigm Function Morphology, Anderson's (1992) Extended Word-and-Paradigm Theory, or Halle and Marantz's (1993) Distributed Morphology. Also, more broadly, how does the theory envisage the relationship between morphology and syntax? Although these questions are too complex to answer in an article of this nature, it can be said that, given its general style, CM is more likely to accord well with monostratal, non-derivational models of syntax like Lexical-Functional Grammar (Bresnan 2001), Head-Driven Phrase Structure Grammar, or Combinatory Categorical Grammar

(Steedman 2000), than with a derivational model like the Minimalist Program (Chomsky 1995).

It is not the aim of the present article to answer the questions above, which will all be explored in future work. This future work will show that, using the principles described here, Yavapai, Tagalog and Georgian morphology can be readily accommodated. However, derivational morphology will require an extra rule, one that changes feature values, and will also require lexical cross-referencing. The theory will also be shown to be able to generate syntactic structures, demonstrating that it could thereby serve as the basis of a theory of morphosyntax and the lexicon.

Acknowledgements

I would like to thank Alison Channon, Sandra Linton, Gereon Mueller, Chris Old, Lionel Posthumus and Jochen Zeller for reading various drafts of the manuscript, and for their helpful and encouraging comments. Two anonymous reviewers for *Stellenbosch Papers in Linguistics* have suggested numerous improvements. All these suggestions have helped to make the article more readable. The remaining errors and defects are of course my own.

References

- Anderson, S. R. 1969. West Scandinavian Vowel Systems and the Ordering of Phonological Rules. PhD dissertation, MIT, Cambridge, Massachusetts.
- Anderson, S.R. 1982. Where's morphology? *Linguistic Inquiry* 13: 571-612.
- Anderson, S.R. 1992. *A-Morphous Morphology*. Cambridge: Cambridge University Press.
- Booij, G. 1994. Against split morphology. *Yearbook of Morphology 1993*: 27-50.
- Bresnan, J. 2001. *Lexical-Functional Syntax*. Oxford: Blackwell.
- Canonici, N. 1990. Noun classes and subclasses. *South African Journal of African Languages* 10(2): 52-58.
- Chomsky, N. 1965. *Aspects of the Theory of Syntax*. Cambridge, MA: The MIT Press.
- Chomsky, N. 1995. *The Minimalist Program*. Cambridge, MA: The MIT Press.
- Halle, M. and A. Marantz. 1993. Distributed morphology and the pieces of inflection. In K. Hale and S.J. Keyser. *The View from Building 20*. Cambridge, MA: The MIT Press.
- Hudson, R. 2007. *Language Networks: The New Word Grammar*. Oxford: Oxford University Press.
- Janda, R.D. 1983. 'Morphemes' aren't something that grows on trees. In J.F. Richardson, M. Marks and A. Chukerman (eds). *Papers from the Parasession on the Interplay of Phonology, Morphology and Syntax*. Chicago: Chicago Linguistic Society, 70-95.
- Kiparsky, P. 1973. 'Elsewhere' in phonology. In S. Anderson and P. Kiparsky (eds). *A Festschrift for Morris Halle*. New York: Holt, Rinehart and Winston.
- Lieber, R. 1981. *On the Organization of the Lexicon*. Bloomington: Indiana University Linguistics Club.
- Lieber, R. 1992. *Deconstructing Morphology*. Chicago: The University of Chicago Press.
- Matthews, P.H. 1974. *Morphology*. Cambridge: Cambridge University Press.
- Pollard, C. and I.A. Sag. 1994. *Head-driven Phrase Structure Grammar*. Chicago: The University of Chicago Press.
- Pullum, G.K. and A.M. Zwicky. 1991. A misconceived approach to morphology. *Proceedings of the West Coast Conference on Formal Linguistics* 10: 387-398.

- Sag, I.A., R. Kaplan, L. Karttunen, M. Kay, C. Pollard, S. Shieber, and A. Zaenen. 1986. Unification and Grammatical Theory. In *Proceedings of the Fifth Annual Meeting of the West Coast Conference on Formal Linguistics*. Stanford: CSLI Publications, 238-254.
- Scalise, S. 1984. *Generative Morphology*. Dordrecht: Foris.
- Selkirk, E.O. 1982. *The Syntax of Words*. Cambridge, MA: MIT Press.
- Spencer, A. 1991. *Morphological Theory*. Oxford: Blackwell.
- Steedman, M. 2000. *The Syntactic Process*. Cambridge, MA: The MIT Press.
- Steele, S. 1995. Towards a theory of morphological information. *Language* 71: 260-309.
- Stump, G.T. 2001. *Inflectional Morphology: A Theory of Paradigm Structure*. Cambridge: Cambridge University Press.

Appendix: glossary and definitions

Note. The notation att(X) used in this appendix is read “the attribute of feature X”; val(X) is read “the value of feature X”.

Additional features. If X and Y have features in common, then the additional features of X are those not contained in Y.

Constant features are features whose values are not sets of alternatives, or variables.

Contradict. A feature X contradicts a feature Y if X and Y have the same attribute but different values (e.g. [category N] and [category V]).

Earlier. A cycle X is earlier than a cycle Y if X is numerically smaller than Y (e.g. [cycle 2] is earlier than [cycle 3]).

Features are grammatical properties of lexical items, for example [tense past] or [number plural]. A feature consists of an *attribute* (that is, a name) followed by a *value*, for example [number pl]. The order follows the practice used in Head-driven Phrase Structure Grammar (Pollard and Sag 1994), and in the f-structure representations of Lexical-Functional Grammar (Bresnan 2001). Features are not necessarily binary, for example the feature [lexeme] can have an indefinite number of values. The value of a feature may be an atom, as [cycle 1] or a set, as [cycle [1, 2]].

Generand. The lexeme that is being processed at any given point in the process of generation.

Extend. A set X extends a set Y if some features of X match features in Y, no feature of X contradicts any feature of Y, and X contains some additional features.

Extension. To extend a set Y by the set X is to add the additional features of X to Y.

Incorporation. A set X can be incorporated into Y if X reflects Y. In the process of incorporation, X is selected and a new cycle of the generative algorithm applies, resulting in X being spelt out.

Match (of features). A feature X matches a feature Y

- (a) if $X = Y$ (e.g. a feature [number pl] matches a feature [number pl]);
- (b) if $\text{att}(X) = \text{att}(Y)$ and $\text{val}(X)$ is a variable (e.g. a feature [lexeme X] matches a feature [lexeme dog]);
- (c) if $\text{att}(X)$ is a variable, and $\text{val}(X) = \text{val}(Y)$ (e.g. a feature [X N] matches a feature [category N]);
- (d) if $\text{att}(X) = [\sim]$, $\text{att}(Y) = [\text{lexeme}]$ and $\text{val}(X) = \text{val}(Y)$ (e.g. a feature [\sim gus] matches a feature [lexeme gus]. This last stipulation allows [\sim] features to link allomorphs to their corresponding basic roots.

Match (of sets). A set X matches a set Y if every feature in X matches some feature in Y.

Matrix cycle. The cycle during which the current cycle was initiated.

Matrix generand. The generand in the matrix cycle.

Morpheme. A lexical item containing a feature [cycle]. Morphemes realize stems, which are more abstract.

Multiple-cycle morpheme. A morpheme which occupies more than one cycle at once, for example a morpheme with the [cycle] feature [cycle [1, 2]].

Narrower. If two sets X and Y both match a set Z, X is the narrower match if it has more features matching features of Z than Y does; or if it has fewer variable features matching features of Z than Y does.

Pāṇini's Principle. A narrower match is chosen before a less narrow one.

Private features. [lexeme] and [cycle] are private features. The concept of private features is needed because these are features which do not have to match features in the generand, but which determine the behaviour of the selected morpheme or category.

Public features. Any feature that is not a private feature is a public feature.

Reflect. A feature [category X] reflects a feature [category Y] if $X = Y$ (e.g. [category V] and [category V]). A set X reflects a set Y if X is not an extension set, and every public feature of X matches or reflects some feature in Y. A set may not reflect itself.

Resolution. If a generand X has one or more variable features, then it needs to be resolved. If X has a feature [lexeme Y], and no other variable features, then [Y] is replaced with the value of [lexeme] in the matrix generand. If X has a feature [lexeme Y] and another feature [Z Y], then $\text{val}([Z Y])$ is replaced with the value of the feature [Z] in the matrix generand, and [lexeme Y] copies its value from this. (If $Z = \sim$, then [Y] copies the value of the feature [lexeme] in the matrix generand.) If X has a feature [lexeme Y] and another feature [Z W], where [Y] and [W] are sequences of phonological (cover) symbols, as $[VC_{[vd +]}]$, [VC], then [W] will take its value from [Z W] in the matrix generand, and the value [Y] will be adjusted according to the features of the phonological symbols. In the example given, if [W] ends on a voiceless consonant, [Y] will end on the corresponding voiced consonant.

Root. A lexical item, with features [lexeme], [category] and [semantics]. Roots remain abstract; once extended into stems, they are spelt out by means of morphemes.

Stem. An extended root.

Vacuous application. If a cycle X is due to be processed, and no morphemes with the feature [cycle X] reflect the matrix generand, then no morpheme is incorporated. The cycle applies vacuously. This is symbolized by “—” in the generative tables.

Variables. The symbols X, Y, Z, W are used as variables. C and V are used as cover symbols for consonants and vowels respectively.

Variable feature. A feature whose value is a variable or a set of alternatives.