1 Introduction

aṣ-ṣarf *of words* the deriving of words one from another, *of winds* shifting from one direction to another, *of wine* drinking it.
al-Fayrûz Abâdî (1329–1414)
al-qâmûs al-muḥîṭ

aṣ-ṣarf The shifting a thing from one state, or condition, to another.
Lane's *Arabic-English Lexicon*

Morphology The science of form.
Oxford English Dictionary

mor·phol·o·gy A study and description of word formation in a language including inflection, derivation, and compounding.
Webster's Third

This book might have a wide audience: computational linguists, theoretical and applied linguists, Semitists, and – who knows – maybe Biblical scholars with interest in Semitic. This is a mixed blessing. While it may serve as an interdisciplinary text, it makes introducing the matter at hand an arduous task. Nevertheless, this chapter attempts to introduce linguistic preliminaries to the nonlinguist, some computational prerequisites to the noncomputer specialist, and the basics of Semitic morphology to the nonsemitist. (To amuse the disappointed reader, I resorted to using quotations at the beginning of each chapter and elsewhere, mostly from the classical Semitic grammatical tradition. I hope this does not prove to be a further disappointment!)

In the definition of terms below, use was made of Trask (1993) and Crystal (1994). It must be noted that what follows is not intended to be an exhaustive coverage of the topics at hand. It must also be stressed that linguists may not necessarily, and often would not, agree with many of the definitions given here (the day is still to come when linguists agree on a definition for what the term "word" denotes). Definitions are given here in the context of the current work.

1.1 Linguistic Preliminaries

It has long been claimed that the morphology of many languages lies within the expressiveness of a class of formal languages known as "regular languages," and

computational morphologists have taken up this claim. This section is an introduction to morphology (Section 1.1.1) and regular languages (Section 1.1.2). Another class of formal languages, the class of context-free languages on which some morphotactic models rely, is introduced as well (Section 1.1.3).

1.1.1 Morphology

1.1.1.1 Basic Definitions

Morphology is the branch of grammar that deals with the internal structure of words (Matthews, 1974). Although linguists may argue for other definitions of morphology, they mostly agree that morphology is the study of meaningful parts of words (McCarthy, 1991). In the English word /boys/, for example, there are two meaningful units: {boy} and the plural marker {s}. Such units, called *morphemes*, are the smallest units of morphological analysis. (Morphemes are shown in braces, { }; and the phonological word in solidi, //.)

Sometimes, morphemes are not easily detected. Like /boys/, the English word /men/ is also a plural noun, but the plural morpheme in this case is embedded in the vowel [e], as opposed to [a] in singular /man/. In fact, morphemes are considered to be abstract units such as {PLURAL}. The {PLURAL} morpheme is realized in various forms called *morphs*: [s] in /boys/ and the vowel [e] in /men/.

Morphs in turn are made of *segments*. For example, {boy} consists of the segments: [b], [o], and [y]. Unless it constitutes a morph, a segment is meaningless. (Segments are shown in brackets, [ ].)

The morpheme that gives the main meaning of the word, for example, {boy} in /boys/, is called the *stem* or *root*. A *free morpheme* can stand on its own. In such a case, the morpheme and the word will be one and the same, for example, the word /boy/ and the morpheme {boy}. A *bound morpheme* requires additional morphemes to form a word, for example, the plural morpheme {s}.

Morphemes that precede the stem or root are called *prefixes*, such as {un} in English /unusual/. Those that follow are called *suffixes*, such as {s} in /boys/. In some languages, a morpheme may consist of two portions, neither of which is meaningful on its own. The first portion acts as a prefix and the second as a suffix. Such morphemes are called *circumfixes*. For example, in the Syriac word /neqṭlūn/ "to kill – IMPF PL 3RD MASC," the circumfix is {ne-ūn} "PL 3RD MASC."

The inventory of all morphs in a language constitutes the morphological lexicon. A lexicon of English need not have entries for /move/, /moved/, /moving/, /cook/, /cooked/, /cooking/, and so on. It only needs to list the unique morphs {move}, {cook}, {ed}, and {ing}. The suffixes apply to {move}, {cook}, and other verbs as well.

The sequence of lexical entries that make up a word is the *lexical form* of the word. For example, the lexical form of /moved/ is {move}β{ed}, where β denotes

a *boundary symbol* that separates lexical entries. The word itself as one sees it on paper (or as one hears it), for example, /moved/, is called the *surface form*.

One important issue in morphology is conditional changes in morphemes. As noted above, the English word /moved/ contains two morphemes: {move} and {ed}. However, the [e] in {move} is deleted once the two morphemes are joined together. In this case, the change is merely orthographic. In other cases, the change might be phonologically motivated. For example, the nasal [n] in the negative morpheme prefix {in} becomes [m] when followed by a labial such as [p]. Hence, English /inactive/ from {in}β{active}, but /impractical/ from {in}β{practical}. Such changes are expressed by *rewrite rules*, also called *productions*. The [n] to [m] change in the above case may be expressed by the rule

$$n \rightarrow m \,/\, \_\_\_\, p$$

which reads: [n] rewrites as [m] before [p].

How does one know that */edkill/, as oppose to /killed/, from the morphemes {kill} and {ed}, is invalid? The licit combinations of morphemes are expressed by another form of rewrite rules, which we shall call here *morphotactic rules* such as

$$\text{word} \rightarrow \text{stem suffix}$$

which reads: "word" rewrites as "root" followed by "suffix." Rewrite rules will be introduced further in Section 1.1.2.3.

1.1.1.2 Linear versus Nonlinear Morphology

Apart from Syriac /neqṭlūn/, the examples given above share one characteristic. The lexical form of a particular word is a sequence of morphemes from the lexicon. For example, the analysis of English /unsuccessful/ produces the lexical form {un}β{success}β{ful}. Because the surface form is generated by the concatenation of the lexical morphemes in question, this type of morphology is called *concatenative* or *linear* morphology.

In many languages, linearity does not hold. Consider the Arabic verb /kutib/ "to write – PERF PASS." This verb consists of at least two morphemes: the root {ktb} "notion of writing" and the vocalic sequence {ui} "PERF PASS." The concatenation of the two morphemes, */ktbui/ or */uiktb/, does not produce the desired result. In this case, the morphemes are combined in a *nonconcatenative*, or *nonlinear*, manner. (It will be shown in the next chapter how a third somewhat abstract morpheme dictates the manner in which the root and vocalic sequence are joined.)

The most ubiquitous linguistic framework for describing nonlinear morphology is based on the autosegmental model as applied to phonology (Goldsmith, 1976). *Autosegmental phonology* offers a framework under which nonlinear phonological (and morphological) phenomena can be described. Tense in Ngbaka, a language

Table 1.1. *Ngbaka tense is marked by tone*

Verb	Tone
kpòlò	low
kpōlō	mid
kpòló	low-high
kpóló	high

of Zaire (the modern Republic of Congo), for example, is indicated by tone, which is considered a morpheme in its own right. Consider the data in Table 1.1 (Nida, 1949). Each verb consists of two autonomous morphemes: {kpolo} "to return" and the respective tense morpheme, which is indicated by a specific tone.

Under the autosegmental model, autonomous morphemes are graphically represented on separate *tiers* as shown in Fig. 1.1. Each morpheme sits on its own autonomous tier: The morpheme {kpolo} sits on the lower tier, while the various tone morphemes, {L} "low," {M} "mid," {LH} "low-high," and {H} "high," sit on the upper tier. *Association lines* link segments from one tier to another. A pair of tiers, linked by some association line, is called a *chart*.[1]

Association lines follow specific rules of association according to two stipulations. The first stipulation is the *Well-Formedness Condition*: All vowels are associated with at least one tone segment and all tone segments are associated with at least one vowel segment, and association lines must not cross. The autosegmental representations in Fig. 1.1 meet this condition. However, the ill-formed representations in Fig. 1.2 violate the Well-Formedness Condition: In Fig. 1.2(a), the last vowel segment is not associated with a tone segment. In Fig. 1.2(b), the first tone segment is not associated with a vowel. In Fig. 1.2(c), association lines cross.

The second stipulation is the language-specific *Association Convention*, which states: Only the rightmost member of a tier can be associated with more than one member of another tier. The association of one member of a tier to more than one member of another tier is called *spreading*, for example, the spreading of the tone morphemes {L}, {M}, and {H} in Fig. 1.1.

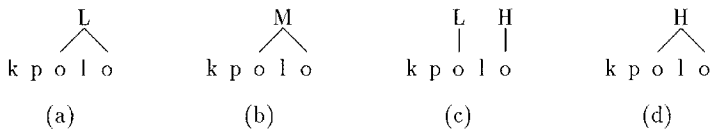

Fig. 1.1. Autosegmental representation of the Ngbaka tense in graphical form: (a) /kpòlò/, (b) /kpōlō/, (c) /kpòló/, (d) /kpóló/. Each morpheme sits on its own autonomous tier, with the stem on the lower tier and the respective tense tone morpheme on the upper tier.

[1] The term "chart" is mostly used in the computational linguistics literature, but not in the linguistic literature.

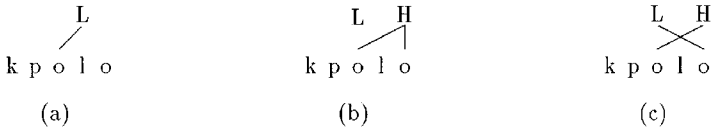

Fig. 1.2. Ill-formed autosegmental representations: (a) the last [o] segment is not linked; (b) the [L] tone segment is not linked; (c) association lines cross.

1.1.1.3 Between Phonology and Syntax

It was mentioned above that morphology is the branch of grammar that deals with the internal structure of words. Two other branches of grammar interact with morphology: phonology and syntax. The former concerns itself with the study of the sound system of languages, while the latter deals with the rules under which words combine to make sentences. Hence, phonology deals with units smaller than morphemes, while syntax describes units larger than words.

One rarely speaks of morphology without reference to phonology. (The term *morphophonology* denotes the phonological structure of morphemes.) One important aspect of phonology, which can hardly be separated from any morphological analysis of words, is *phonological processes*. These are conditional changes that alter segments. Some of the processes mentioned in this book are as follows: *assimilation*, in which one segment becomes identical to, or more like, another as in [n] → [m] above (see p. 3); *syncope*, or *deletion*, as the deletion of the first [a] in Syriac */qaṭal/ → /qṭal/[2] "to kill"; *epenthesis*, or *insertion*, as the insertion of / ʔi/ in Arabic /nkatab/ → / ʔinkatab/[3] "to write – REFL"; and *gemination*, or *doubling*, which involves the repetition of a segment (usually consonant) as in Arabic /katab/ → /kattab/[4] "to write – CAUS"; in this case, the gemination of [t] is morphologically motivated.

Another phonological phenomenon that concerns us is syllabification. The English word /morphology/, for example, consists of the syllables (separated by dots): mor·pho·lo·gy. *Open* syllables end in a vowel, for example, /lo/, while *closed* syllables end in a consonant, for example, /mor/. The components of a syllable can be represented by a smaller unit, the mora, for example, /lo/ consists of one mora while /mor/ consists of two morae; syllabic weight is defined by the number of morae in a syllable: *light* syllables contain one mora, while *heavy* syllables contain two morae.

One also rarely speaks of morphology without reference to syntax. It is not uncommon for an orthographic word in one language to represent a sentence in another. For example, Syriac /baytå/ "the house," /bbaytå/ "in the house," /dabbaytå/ "he who is in the house," /ldabbaytå/ "to him who is in the house," /waldabbaytå/ "and to him who is in the house" (Robinson, 1978). Syntax (apart from what is

[2] Syriac does not allow unstressed short vowels in open syllables, apart from few diachronic cases, for which see p. 115.
[3] Arabic is devoid of initial consonantal clusters.
[4] The Arabic causative is derived by the gemination of the second consonant.

required by morphotactics) is beyond the scope of this work. It suffices to note that for many languages, such as Semitic, the analysis of the orthographic word ventures into the realm of morphosyntax. In practical computational systems, a morphology module must account for phonology and – to some extent – syntax.

1.1.2 Regular Languages

Formal language theory establishes a hierarchy of formal languages based on their complexity and expressiveness. The class of regular languages is the most basic in the hierarchy.

Formal languages are defined in terms of *strings*, strings in terms of *alphabets*, and alphabets in terms of *sets*. These terms are introduced below.

1.1.2.1 Sets

A *set* is a collection of objects without repetition. A set can be specified by listing its objects. The following set represents the days of the week:

{ Monday, Tuesday, Wednesday, Thursday, Saturday, Sunday, Friday }

Each object in the set is called an *element* of the set. Elements are separated by a comma and are placed in braces, { }. No two elements can be the same; however, the order of the elements is not important. For instance, in the above set, Friday appears after Sunday. When the elements in the set are too long to list, one can use a *defining property* instead. The above set can be rewritten as follows:

$\{ x \mid x \text{ is a weekday} \}$

Read: x where x is a weekday.

If an element x is a member of a set A, we say $x \in A$ (read: x in A). If an element x is not a member of a set A, we say $x \notin A$ (read: x not in A). For example, given the set $A = \{ 1, 2, 5 \}$, we say $2 \in A$, but $3 \notin A$.

The set containing no elements, usually denoted by { } or $\emptyset$, is called the *empty set*.

A set A is a *subset* of another set B, designated by $A \subset B$, if every element in A is an element in B. For example, { 1,2 } is a subset of { 1,2,3,4 }; however, { 1,5 } is not a subset of { 1,2,3,4 } because the latter does not include the element 5. If A is a subset of B but may also be equal to B, we say $A \subseteq B$.

There are several operations that can be applied to sets: The *union* of sets A and B, denoted by $A \cup B$, is the set that consists of all the elements in either A or B. For example, let $A = \{ 1, 2, 3 \}$ and $B = \{ 3, 4, 5 \}$, then $A \cup B = \{ 1, 2, 3, 4, 5 \}$. Note that since a set cannot have duplicates, the union contains only one instance of the element 3. We write

$$\bigcup_{i=1}^{n} A_i$$

to denote $A_1 \cup A_2 \cup \cdots \cup A_n$.

The *intersection* of sets A and B, denoted by $A \cap B$, is the set that consists of all the common elements in A and B. For example, let $A = \{ 1, 2, 3 \}$ and $B = \{ 3, 4, 5 \}$; then $A \cap B = \{ 3 \}$. We write

$$\bigcap_{i=1}^{n} A_i$$

to denote $A_1 \cap A_2 \cap \cdots \cap A_n$.

The *difference* of sets A and B, denoted by $A - B$, is the set that consists of all the elements in A that are not in B. For example, let $A = \{ 1, 2, 3 \}$ and $B = \{ 3, 4, 5 \}$; then $A - B = \{ 1, 2 \}$.

The *complement* of a set A, denoted by $\overline{A}$, is the set that consists of all the elements in the universe that are not in A. The *universe* set contains all elements under consideration. If we assume that the universe set contains all the days of the week and

$$A = \{ \text{Monday, Wednesday, Friday} \}$$

then

$$\overline{A} = \{ \text{Tuesday, Thursday, Saturday, Sunday} \}$$

The *cross product* of sets A and B, denoted by $A \times B$, is a set consisting of all the pairs (a_1, a_2) where the first element, a_1, is in A_1 and the second element, a_2, is in A_2. For example, let $A = \{ 1, 2 \}$ and $B = \{ 3, 4, 5 \}$; then $A \times B = \{ (1, 3), (1, 4), (1, 5), (2, 3), (2, 4), (2, 5) \}$. We write

$$\prod_{i=1}^{n} A_i$$

to denote $A_1 \times A_2 \times \cdots \times A_n$. We also write B^n to denote the cross product of B by itself n times; that is, $\underbrace{B \times B \times \cdots \times B}_{n \text{ times}}$.

With the use of defining properties, the above operations can be defined as follows:

$$\begin{aligned}
A \cup B &= \{ x \mid x \in A \text{ or } x \in B \} \\
A \cap B &= \{ x \mid x \in A \text{ and } x \in B \} \\
A - B &= \{ x \mid x \in A \text{ but } x \notin B \} \\
\overline{A} &= \{ x \mid x \notin A \} \\
A \times B &= \{ (a_1, a_2) \mid a_1 \in A_1, a_2 \in A_2 \}
\end{aligned}$$

A *finite* set contains a finite number of elements. For instance, the set $\{ n \mid 1 \leq n \leq 10 \}$ is a finite set of 10 elements, that is, the integers 1 to 10. An *infinite* set contains an infinite number of elements. For example, $\{ n \mid 1 \leq n \}$ represents all positive integers, from 1 to infinity.

Any subset of the cross product $A_1 \times A_2$ is called a *binary relation*. A_1 is called the *domain* of the relation and A_2 is called the *range* of the relation. It is possible to have a relation on one set, for example, a relation on $A \times A$.

1.1.2.2 Alphabets and Strings

An *alphabet* is a finite set of symbols. *Symbols* are usually letters or characters. The English alphabet is the set

{ A,B, . . . , Y,Z,a,b, . . . , y,z }

A *string over* some alphabet is a finite sequence of elements drawn from that alphabet. If A = {a,b,c} is an alphabet, then the following sequences, *inter alia*, are strings over A: "a," "aa," "aab," "aac," "caa," and "cbbba." However, the string "aad" is not a string over A since the element 'd' is not in A. (Strings are shown in double quotes when they appear in text; characters or symbols are shown in single quotes.)

The number of elements in a string x determines the *length* of the string, denoted by $|x|$. The length of the string "ab" is two and the length of "cbba" is four. A string of length zero is called the *empty string* and is denoted by ϵ.

The terms prefix and suffix apply to strings as they apply to natural languages (see p. 2).

The *concatenation* of two strings x and y, denoted by xy, is the string formed by appending y to x. For example, if x is "may" and y is "be," then xy is "maybe."

Concatenation is used to define *exponentiation*. If x is a string, we write x^2 to denote the concatenation of x with itself twice, that is, xx. Similarly, x^3 denotes the concatenation of x with itself thrice, that is, xxx. In this vein, $x^1 = x$ and x^0 is the empty string ϵ. For example, let x be the string "ha;" we say x^0 is ϵ, x^1 is "ha," x^2 is "haha," x^3 is "hahaha," and so on.

The *Kleene star*, denoted by x^*, denotes zero or more concatenations of x with itself: ϵ, x, xx, xxx, and so on. To exclude the empty string, we use the Kleene plus notation, x^+, which denotes one or more concatenations of x with itself.

1.1.2.3 Languages, Expressions, and Grammars

The term *language*, or *formal language*, denotes any set of strings over some alphabet. For example, let A = { a,b,c } be some alphabet. All of the following sets of strings are languages over A:

L_1 = { b,ab,aab,aaab,aaaab, . . . }
L_2 = { b }
L_3 = { abcc,abca,aaba,ccca,caba, . . . }

L_1 is an infinite language over A where each string consists of zero or more instances of 'a' followed by one instance of 'b'. L_2 is a finite language over A, and it consists of only one string, the string being a symbol from the alphabet. L_3 is a finite language over A whose strings are of length four. The language { abc, add }, however, is not a language over A since "add" is not a string over A; this is so because 'd' is not in A.

Expressions are used to describe the strings of a language. The strings in L_1, for example, can be expressed by the expression a*b: zero or more instances of 'a' followed by one 'b'. The language L_2 can be expressed by the expression b since it contains only that element. Expression may contain other operators such as disjunction, | (read 'or'). For instance, the strings in L_3 begin with either an 'a' or a 'c', followed by three arbitrary symbols from A; this can be described by the expression $(\mathrm{a} \mid \mathrm{c})A^3$.

Given two alphabets, one can use expressions to describe languages over the two alphabets. Consider the following two alphabets, which represent capital and small letters, respectively:

$$C = \{ \mathrm{A,B}, \ldots, \mathrm{Y,Z} \}$$
$$S = \{ \mathrm{a,b}, \ldots, \mathrm{y,z} \}$$

The language CS^* consists of all strings that start with one capital letter followed by zero or more small letters, e.g. "I," "Good," "Bed." The language

$$(C \mid S)S^3 \,\mathrm{ing}$$

consists of all strings that start with either a capital or small letter, followed by three small letters, followed by "ing," for example, "booking," and "Writing."

Languages are described by grammars. A *formal grammar* consists of an alphabet and a set of rewrite rules. Generally, a *rewrite rule* consists of a left-hand-side and a right-hand side separated by an arrow, for example,

$$\mathrm{y} \rightarrow \mathrm{i\ e}$$

Read: 'y' rewrites as 'i' followed by 'e'. Applying the rule on the string "entrys," which consists of the stem {entry} concatenated with the plural morpheme {s}, results in "entries," after replacing 'y' by "ie." This is the rule that applies to English plurals ending in a 'y.' However, there is nothing preventing the rule from applying to any 'y.' Applying the rule on "may" results in the undesired "maie."

To restrict the application of rules, one specifies *left* and *right contexts*, separated by an environment bar ___ . The rule only applies when the contexts are satisfied. The above rule can be rewritten as

$$\mathrm{y} \rightarrow \mathrm{i\ e} \,/\, \_\_\_\ \mathrm{s}$$

Read: 'y' rewrites as 'i' followed by 'e' before 's.' Here, the left context is not specified. (The slash, /, separates the contexts from the right-hand side.) The above

S → the A
A → old B
B → man
B → woman

Fig. 1.3. A set of rewrite rules that generate the sentences *the old man* and *the old woman*. Nonterminal symbols start with a capital letters.

rule only applies when there is an 's' to the right of 'y;' hence, it does not apply to "may."

So far, *terminal symbols* were used in rules; that is, symbols drawn from the alphabet in question. It is also possible to use *nonterminal symbols*; that is, symbols that are derived from other symbols. These are designated with capital letters. Consider the following alphabet whose symbols are actual words, {man, old, the, very, woman}, and the rules in Fig. 1.3. The first rule states that a sentence S rewrites as the word "the" followed by A. According to the second rule, the symbol A in turn, rewrites as the word "old" followed by B. Now B rewrites as either "man" or "woman" according to the third and fourth rules, respectively. This grammar generates the two sentences: "the old man" and "the old woman". The derivations can be illustrated graphically by *parse trees* as in Fig. 1.4.

Grammars, and hence languages derived from them, are of various complexities. The least complex are *regular language*s. These can be generated by rewrite rules of the form

$$A \rightarrow a\ B$$

or

$$A \rightarrow a$$

Here A and B are nonterminal symbols and a is a terminal symbol. The formal definition of regular languages over an alphabet Σ is as follows:

(i) The empty set is a regular language.
(ii) For each a in Σ, $\{ a \}$ is a regular language.

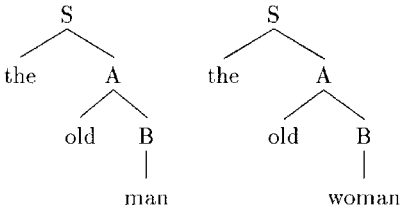

Fig. 1.4. Parse trees for the sentences generated by the rules in Fig. 1.3.