# Rule Notation in Phono

Most phonologists are familiar with a standard rule notation based on expressions such as the following—

A → B / C _ D

—meaning that element A changes into (or is replaced by) element B in the environment following element C and preceding element D. In other words, each occurrence of the sequence CAD (known as the rule's **structural description**) is changed into CBD (and the change of A to B is called the **structural change**).

In developing Phono and the Spanish model in tandem, however, I found that many of the rules were too complex to be expressed with the standard notation alone. Phono's rule notation instead portrays the structural description as a *hierarchy* of one or more **if-clauses**, and the structural change as a *series* of one or more **then-clauses**. (The distinction between "hierarchy" and "series" will become clear below.) Specifically, the format of a rule (as seen by the user) consists of four fields.

- The name of the rule
- The if-lines, labeled in order by letters (A, B, C...)
- The then-lines, labeled in order by numerals (1, 2, 3...)
- The end line, consisting of the word "END" plus the rule name.

The if-clauses and then-clauses alike are expressed in terms of binary feature values, based mainly on the features defined by Chomsky & Halle (1968) in *The Sound Pattern of English*. (The two versions of Phono—"Version D" (downloadable, 2016) and "Version O" (online, 2017)—use slightly different sets of features.)

In addition to feature names and their binary values, the if- and then-lines refer to the positions of segments in the word by means of **location expressions**, always enclosed in parentheses. There are three types of location expressions: **absolute**, **relative**, and **by feature value** (the latter used only in COUNT-type if-lines, as described below). An absolute location expression consists of a positive or negative number, indicating the position of a segment counting from the beginning or end, respectively, of the word. The initial segment is "(1)", the third-from-last segment is "(–3)", and so on.

A **relative** location expression is based on the fact that every application of a rule involves a scan of the word from left to right in search of the conditions of the rule's structural description. Each segment has its moment as the **focus** of the scan, during which, if the conditions are found, the change is carried out. Relative location expressions contain an asterisk to refer to this focus segment. Additionally the expression may contain a number, positive or negative, to refer to a segment following or preceding the focus segment. So "(*)" is the focus segment; "(*+1)" is the segment immediately following the focus segment; and "(*–2)" is the second segment to the left of the focus segment.

A location expressed **by feature value** specifies a certain segment, not by its position directly, but rather by the value of one of its features, as explained for COUNT-type lines, below.

The following example is the historical rule for diphthongization in Spanish, which in conventional notation would be “/ɛ/ → /je/ and /ɔ/ → /we/”:

```
DIPHTHONG
A:  B  and  C
B:  +low(*)
C:  back(*)  =  round(*)
1:  –low(*)  +high(*)  –syll(*)  –stress(*)
2:  INSERT  e  (*+1)
3:  +stress(*+1)
END DIPHTHONG
```

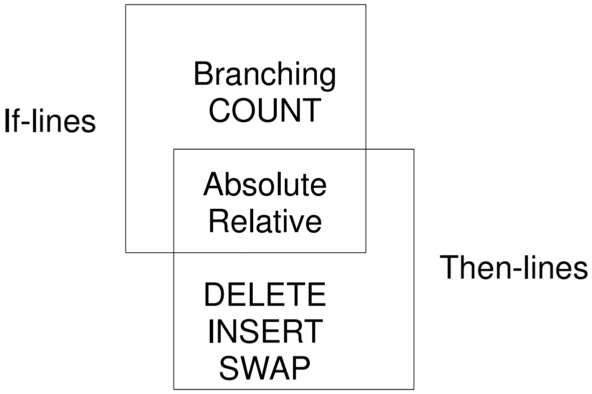
The rule is interpreted as follows:

- If A is true, then carry out the changes indicated by 1, 2, and 3.
- A is true if both B and C are true.
- B is true if the focus segment (in the left-to-right scan of the word) is [+low]. There is no need to specify “[+syll]” (syllabic) to identify it as a vowel, since there are no [+low] consonants or glides in the language at this moment in its history. In this version of the rule (for use with Phono Version D), the value [+low] is sufficient to specify three vowels: /a/ and the open mid vowels /ɛ/ and /ɔ/. (Spanish *synchronic* diphthongization is nowadays associated with stress, but there is no need to specify it in the rule because, at the historical moment in question, the open mid vowels can occur only in stressed syllables.)
- C is true if the values of the [back] and [round] features of the focus segment agree: either both plus (/ɔ/) or both minus (/ɛ/). This eliminates /a/, which is [+back, –round].
- 1: The first change is to convert the focus segment—the /ɛ/ or /ɔ/ found by lines A, B, and C—to the corresponding high glide, /j/ or /w/, by making it [+high, –syll]. Of course [+high] entails [–low], and probably [–syll] entails [–stress], but these changes must be made explicit in the rule because the present versions of Phono do not carry out these entailments automatically.
- 2: The “INSERT” line inserts the indicated segment (/e/—or, more accurately, its feature values) in the first place following the focus segment.
- 3: The newly inserted segment is given the value [+stress].

The toolkit for constructing rules consists of four types of if-lines and five types of then-lines—with the “Absolute” and “Relative” types available for both if- and then-functions—for a total of seven types, as shown in Figure 1. COUNT, DELETE, INSERT, and SWAP are keywords that appear in those respective types of lines.

A **Branching** if-line conjoins the letter labels of two subsequent if-lines with a conjunction, either “and” or “or”. The line “A: B and C” in the diphthongization rule described above is a typical example. Most rules have more than one if-line, and in that usual case, the first if-line, labeled “A”, must be a Branching line that states the relationlship between lines B and C. Then B and C, in turn, may be of other line types, or either of them may also branch into D and E, and so on. Line A, as the top member of a hierarchy, must represent *all* the if-lines of the rule combined, and each if-line below line A must be represented in a Branching line somewhere above itself.

Figure 1: Types of If-lines and Then-lines



A **COUNT**-type if-line carries out a left-to-right subscan from one designated segment to another, counting the number of occurrences of a designated feature value between them. It consists of six items in order:

- the keyword COUNT;
- a signed feature name, the **target feature value**;
- a location expression for the **head** of the subscan;
- a location expression for the **foot** of the subscan;
- a **sign** of comparison: " >", "<", or "=";
- a numerical **standard** (0, 1, 2, etc.).

The example line below means "If the count of syllabic segments from the focus segment to the final segment is greater than two...". That is, the rule applies from the initial segment up to, but not including, the second-to-last vowel in the word.

```
B:  COUNT  +syll  (*)  (–1)  >  2
```

In a COUNT-type line, the abstract feature value [+segment] (never minus) can be used to count segments *per se*. The following example—by insuring that the number of segments from focus to final position is 1—means, in effect, "If the focus segment is word-final...".

```
C:  COUNT  +segment  (*)  (–1)  =  1
```

In a COUNT-type line there exists the option of specifying the head or foot of the subscan (or both) as locations **by feature value**. When the *head* of the subscan is located by feature value, it refers to the first instanceof the specified value *preceding* the focus segment; and likewise a location by feature value for the *foot* of the subscan refers to the first instance *following* the focus. The following example, from a rule of the Spanish model, helps to locate the post-tonic vowel of a word. It counts vowels ([+syll]) in a subscan that begins with the first stressed segment (necessarily a vowel) found to the left of the focus and ends with the focus itself. If the number is exactly 2 (and if the focus segment is a vowel) then the focus is on the first vowel after the stressed vowel.

```
D:  COUNT  +syll  (+stress)  (*)  =  2
```

COUNT-type if-lines have a broad variety of uses, including determining whether the focus segment is or is not initial or final, identifying a vowel as belonging to the penultimate syllable, identifying a word as polysyllabic or as unstressed, or insuring that no vowel intervenes between a marked agent of change and the focus segment.

An **Absolute**-type if- or then-line—the simplest line type—is made up of one or more **units**, each consisting of a signed feature name followed by a location expression, as in the following example:

```
E:  +syll(*)  +low(*)  –cons(*–1)  –syll(*–1)
```

This if-line means “If the focus is a low vowel preceded by a glide...” (the focus segment, “(*)”, is plus-syllabic, plus-low; the preceding segment, “(*–1)”, is minus-consonantal, minus-syllabic). Meanwhile, when an Absolute-type line appears in the “then” section of a rule, it assigns new values to the indicated features in the indicated locations.

The **Relative**-type if- or then-line corresponds to the Greek-letter variable, or “alpha” device introduced by Chomsky and Halle in *The Sound Pattern of English* (p. 83). A Relative line refers to the dependence of one feature value in the word upon another, as either equal or opposite. Used as an if-line, the Relative line refers to various conditions of interdependence among feature values. As a then-line, it appears in many rules of assimilation (signs equal) or dissimilation (signs opposite). The line consists of five elements:

- an unsigned feature name, the **target feature**;
- a location expression, the **target location**;
- a **sign** of equality (=) or inequality (“\=” in Phono Version D, “<>” in Version O);
- a second feature name, the **standard feature**;
- a second location expression, the **standard location**.

The following example, repeated from the Spanish diphthongization rule given above, specifies the natural class of the back-rounded and the front-unrounded vowels together—[α back, α round]:

```
C:  back(*)  =  round(*)
```

A then-line for dissimilation uses a sign of inequality. The following example brings about dissimilation of nasality from that of the preceding segment:

```
1:  nas(*)  <>  nas(*–1)
```

In a Relative-type line it is possible to use the unsigned feature name [all], simultaneously for the target feature and the standard feature, meaning that all their respective feature values agree. In an if-line it serves to detect a geminate; and in a then-line, to create a geminate by total assimilation.

A DELETE-type then-line deletes a specified segment. It consists of the keyword DELETE and a location expression, either absolute in the word or relative to the focus segment. The following example would delete the word-final segment:

2:  DELETE  (–1)

An INSERT-type then-line inserts a segment in a given location in the word.  It consists of three elements:

- the keyword INSERT;
- the character to be inserted;
- a location expression.

The specified character must appear in the model's input alphabet (and the input alphabet can include only characters that are available on the keyboard).  The INSERT line inserts the character's column of feature values from the model's input alphabet into the matrix of feature values that represents the word during derivation.  In the event that the desired new segment differs in some feature values from those of the character inserted from the alphabet, these values can be adjusted by means of additional then-lines, of the Absolute or Relative types.

In INSERT-type lines, the location expression referring to the focus segment alone, "(*)", means that the focus segment is *replaced by* the new segment.  Otherwise, expressions with the focus plus or minus a number—(*+2), (*–1), etc.—mean insertion at the indicated segment *boundary*, counting right or left of focus respectively.  If the location expression in an INSERT line is of the absolute type (number without asterisk), that refers, as elsewhere, to the distance from the beginning or end of the word.  A location of "(1)" would place the new segment to be initial in the word; a location of "(–2)" would insert it as the second-to-last segment.

The following example—repeated here from the diphthongization rule given above—inserts the feature values of the segment /e/ immediately after those of the focus segment:

2:  INSERT e (*+1)

A SWAP-type then-line performs metathesis:  it interchanges the positions of two segments in the word.  Its format consists of three elements:  the keyword SWAP and the location expressions of the two segments to be interchanged.  The following example line interchanges the focus segment with the segment that precedes it:

3:  SWAP (*) (*–1)

In the event that an actual rule is too complex to be expressed as a single rule in this notation, it is possible to notate it as a sequence of two rules in the model, by means of an abstract feature named "mark".  The first notational rule can search for one set of conditions and label a segment [+mark] when they are found.  Then the second notational rule can search for the remainder of the conditions, and if they are true and the sign of [mark] is plus, then the structural change can be carried out.  In the model for Spanish, the rule for syncope (loss of an "intertonic" vowel) operates in this two-stage fashion, separating the searches for the vocalic and the consonantal features of the environment.  This sequence is usually followed by the application of a rule with a name like "UNMARK", to return the value of [mark] to minus after it has served its purpose.

This system of if/then notation has thus far proved capable of portraying all purely phonological changes that I have encountered (i.e. those without reference to grammatical functions). It has yet to be tested on languages with vowel harmony (such as Turkish) or discontinuous morphemes (such as Arabic), and it will need to incorporate additional features in order to work with tone languages (such as Chinese).

-o0o-