Formalizing a phonological rule

We have begun to see situations where the **surface representation** (SR) of a morpheme, word, or phrase (essentially, its actual pronunciation)[1] is different from the **underlying representation** (UR) of that morpheme, word, or phrase (the way in which it is stored in the mental lexicon).

When this situation arises, we propose that the phonological grammar causes a **phonological process** to take place — that is, some aspect(s) of the phonological representation is changed by the grammar. In LING 101, you probably learned to model a phonological process as a **phonological rule** (or sequence of rules) that can **make changes** to a phonological representation. In LING 200, we will now expand our model of the phonological grammar to include phonological rules.

1. The structure of a phonological rule

A phonological rule takes the following form:

target → **change** / **environment**

- The **target** is the class of segments (phones, sounds) that the rule ***applies to***.

  - It must always be stated in terms of ***features*** — even if the class that the rule applies to consists of only one segment! (This is because we have proposed that features are how our model of the grammar refers to segments and segment classes.)
  - How many features do we need to specify? A useful rule of thumb: State only as many features as needed to distinguish the segment or segments in the target of the rule from the other segments that occur in the language (as seen in the data set).
  - If the rule is an ***insertion*** rule, the target is written 'Ø' (null, zero). We want to show that some segment gets inserted where at first there was nothing present.

- The **change** is a list of *only* those ***features*** that are ***changed*** by the rule.

  - We don't conceive of a rule as something that replaces one segment with an entirely different segment — instead, we see a rule as making *adjustments* to a segment. Being careful to state the change imposed by a rule only in terms of those features that actually change is a way of focusing our attention on this aspect of the way we propose that the phonology functions.
  - If the rule is a ***deletion*** rule, the change is written 'Ø' (null, zero).
  - For an ***insertion*** rule, since the target is Ø, "those features that are changed by the rule" would technically be *all* of them. However, by convention, the change in an insertion rule follows the same principles as the target in a non-insertion rule: specify only as many features as needed to distinguish the inserted sound from the other sounds that occur in the language (as seen in the data set).

1 More accurately, the surface representation of a linguistic form is the *output of the phonology* — the representation that the mental grammar sends off as *instructions* for implementing pronunciation.

- The **environment** restricts the rule to applying only in certain ***contexts***. (If there are no restrictions on the context where the rule applies, no environment is stated in the rule.)

  - Environments often refer to segments or segment classes; again, use ***features*** for this.
  - A useful notation convention: **C_0**, which means 'zero or more [–syll] segments'. This is used for rules where it is the **nearest vowel** that determines the application of the rule, whether or not there are consonants intervening.
  - Use an underscore '__' to show the position of the target with respect to the environment.

  Examples:

  - [+nas] __ after a nasal segment
  - __ $\begin{bmatrix} \text{–son} \\ \text{+cont} \end{bmatrix}$ before a fricative
  - [+cons] __ [+cons] between (non-glide) consonants
  - [–hi] C_0 __ after a non-high vowel, with zero or more consonants intervening

The basic mechanics of using formal rule notation are relatively straightforward. However, the more carefully you think about how to use segment classes and features to express **meaningful generalizations** about the pattern, the more insightful your rule will be.

Here are some general points to watch out for:

- Phonological rules never ***add*** **morphemes**
  - If (for example) a plural morpheme appears on some forms but not on others, that is because it is added by the morphological component of the mental grammar
  - Rule of thumb: If segments are added to a form in order to *change meaning*, that is not something that the phonology is responsible for
- The target of a rule is almost *never* a sequence of multiple segments
  - If two different segments of a form show differences between the UR and the SR, there is probably **more than one rule** applying
- Rules should be stated as **generally** as possible
  - Using features and segment classes helps with this
  - Are several similar segments undergoing a similar kind of change? If so, can one general rule be used to cover all cases?

As a basic strategy, **make your analysis as simple and general as possible** — don't include any information in your rule statement that doesn't *need* to be there to make the rule work. Always remember that one important reason for developing solutions to phonology problems is to explore what characteristics or entities are *crucially necessary* in order to model human mental

grammar. Therefore, we want to make our rule statements — and our phonological analyses in general — as simple and elegant as possible, in order to highlight what factors (features, rule types, etc.) really are ***necessary*** for capturing the phenomenon under investigation. And if we find that a common, phonetically plausible pattern can *only* be written as a complex, cumbersome rule, this might indicate a problem with our model itself.

The rest of this handout discusses ways of making rules more general and insightful.

---

2. Stating the target of a rule

A. Stating the target in terms of features

The **target** of a rule is the class of segments that the rule **applies to**. You can find it by:

(a) listing out all the segments you want to include in this class

(b) listing out all the segments you do *not* want to include — do this based on the inventory of sounds that **appear in the data set** you are working on

(c) putting together a (hopefully small) set of feature values that lets you **include all of the segments in (a)** while **not including any of the segments in (b)**

It is *not a problem* if the rule you write would also apply to additional segments that simply don't appear in the data set. (In fact, this is a good thing, because it means your rule is general and you are making predictions!) More on this in section 6 below.

B. Never involve morphology unless it's absolutely necessary

Here's another important point about keeping rules general: **Never assume that a rule will be specific to an individual morpheme** unless there is actual evidence showing that this is the case. (And even if you think there *is* evidence, first see if you can come up with a different way of analyzing the data set that does not need to refer to particular morphemes. Restricting a phonological rule to specific morphemes should be seen as an absolute last resort that should be avoided whenever possible.)

For example, in **Lamba**, the vowels in the neuter and applied suffixes change in height depending on the height of the preceding vowel.

> **(2) Initial description of Lamba [i]~[e] alternation**
>
> In the neuter and applied suffixes in Lamba, the high front vowel /i/ becomes the mid front vowel [e] when it is preceded by a mid vowel. (Otherwise, it remains unchanged.)

Now, let's look to see if we can make our analysis any more general.

In fact, we discover that **there is no need to restrict our rule to these particular morphemes**. There are no vowels anywhere in the data set that would be exceptions to Generalization #1 — that is, no high vowels are *ever* preceded by mid vowels *in the whole data set*. So, to make our analysis more general, we revise our description of the phenomenon to apply to the language as

a whole, not specifically to the neuter and applied suffixes. (We could test this hypothesis if further investigation into Lamba turned up verb roots with multiple vowels: we are currently predicting that the same generalization will hold within roots too.)

> **(3) Morphology-free generalization for Lamba [i]~[e] alternation**
>
> In Lamba, high front vowels become mid when they are preceded by a mid vowel.

Another good reason for treating this phenomenon as a general pattern of Lamba is that these two different verb suffixes are seen to behave exactly the same way. If we decided to write morpheme-specific rules, we would need to write one for the neuter suffix and one for the applied suffix, and it would be a coincidence that the two rules did **exactly the same thing**.

---

3. Stating the change imposed by the rule

Once you have stated the target of the rule as a segment class, in terms of features, the next step is to state the change imposed by the rule — again in terms of features.

A. Only change features that the rule is really changing

When stating the change that a rule carries out, **never repeat features** that are crucially part of the segment class that is the target of the rule. For example, if a language changes /s/ to [z] between vowels, you should *not* write this rule:

(4) BAD RULE

$$\begin{bmatrix} \text{COR} \\ +\text{cont} \\ -\text{voi} \end{bmatrix} \rightarrow \begin{bmatrix} \text{COR} \\ +\text{cont} \\ +\text{voi} \end{bmatrix} \; / \; [-\text{cons}] \_\_ [-\text{cons}]$$

(This rule makes /s/ → [z] / V_V.)

This rule is bad because it implies that the mental grammar is totally replacing /s/ with some arbitrary sound [z]. An implausible rule changing /s/ to [m] wouldn't look much different.

What we want to do instead is write a rule that captures our insight that what the mental grammar is really doing in this case is **changing the voicing** of /s/.

(5) GOOD RULE

$$\begin{bmatrix} \text{COR} \\ +\text{cont} \\ -\text{voi} \end{bmatrix} \rightarrow [+\text{voi}] \; / \; [-\text{cons}] \_\_ [-\text{cons}]$$

(Still achieves /s/ → [z] / V_V.)

So we only mention the feature that actually *has been changed* — [+voi] — and let the outcome of the rule inherit all the other features of the target, unchanged.

B. An apparent exception (but not really): Vacuous rule application

There is one case where it doesn't hurt to make the *change* of the rule mention feature values that some of the sounds in the class that is the *target* of the rule already have. This is when **stating the target of the rule in a maximally general way** leads to **vacuous** (i.e., "empty") rule application.

This is best illustrated with an example. Suppose, in the imaginary language we were just discussing, there is no phoneme /z/. The only anterior coronal fricative phoneme is /s/. In this case, it obviously won't hurt to state the target of the rule more generally than we did in (5), since there is no /z/ out there for the rule to apply to. (I.e., remove [–voi] from the target.)

(6) EVEN BETTER RULE

$$\begin{bmatrix} \text{COR} \\ +\text{cont} \end{bmatrix} \rightarrow [+\text{voi}] \;/\; [-\text{cons}] \_\_ [-\text{cons}]$$

(Still achieves /s/ → [z] / V_V.)

But now suppose a language has /s/ and /z/ as separate phonemes, along with this same rule that turns /s/ into [z] between vowels. (Note that /s/ and /z/ therefore undergo **neutralization** in this language in the context V_V.) We can still use the rule as stated in (6) *even though it will technically apply to /z/ also*. Why is this okay?

(a) The rule **doesn't contradict the data set**, because "voicing" /z/ still gives [z].

(b) The rule **doesn't technically violate the principle of not repeating unchanged information after the arrow**. Why not? Because the feature [+voi] is not a *crucial* attribute of the segment class that forms the target of this rule. We can tell this because the feature [+voi] isn't mentioned in the target of the rule — and also because some of the members of the segment class in the rule's target (as stated very generally in (6)) are voiced and some are not.

In summary, it's acceptable (and often actually a good sign of a nice, general rule) to include a feature in the statement of the change caused by the rule even if some of the sounds in the target segment class already had that feature value, as long as the feature in question *is a necessary outcome of the rule* AND *wasn't necessary in specifying the target* of the rule.

---

4. Stating the environment of a rule

Now that you have stated the target and the change, the last step in writing a rule is to state the environment where the rule applies.

As always, it is important to do this as generally as possible. For example, suppose a language turns /m/ into [b] at the beginning of a word, but /m/ appears as [m] every single time it is not at the beginning of a word. Suppose also that every time /m/ appears at the beginning of a word (and therefore changes to [b]), it also happens to be followed by a vowel.

(7) /m/ to [b] in word-initial position

A. Data set /ma/ → [ba] /emul/ → [emul]
/mog/ → [bog] /amda/ → [amda]

B. Bad rule [LAB, +nas] → [–nas] / #__ [–cons]

C. Good rule [LAB, +nas] → [–nas] / #__

Why is (7B) a bad rule? Because it isn't general enough. It *happens to be true* that the /m/-to-[b] change always happens before a vowel, but you *don't need that information* to write a rule that puts [b] in the correct places in this language. All you have to refer to in the environment to write a rule that works is word-initial position — so that is why rule (7C) is better.

Also, the same warning about **morphology** applies in stating the environment as it does for stating the target of a rule: It should be something you do only as an absolute last resort, when you have tried all other possibilities and proven that they do not work.

5. Checking a rule for incorrect predictions (counterexamples)

How do we check to make sure our rule makes the correct predictions? We need to apply it, perfectly literally (you could even say mindlessly), to the data set:

(a) Find every sound in the data set whose UR (or intermediate form, if a prior rule has already applied) matches the segment class of the rule's target.

(b) Check to see if that sound stands in the rule's environment.

(c) If so, check to make sure that the rule has applied to the sound in question.

If you find a sound to which the rule *as written* will apply, but is not supposed to according to the actual data set, then there is something wrong with your analysis. Maybe the rule isn't stated quite specifically enough. Or, maybe you have made the wrong hypothesis about the UR of some morpheme or some class of sounds. Checking your rule for incorrect predictions is an important step in any phonological analysis, because it helps you find problems like these.

6. More about generalizing rules

A. Combining individual rules to make a more general analysis

Now that you have stated your rule, and it works, it is time to think about whether this rule stands alone, or whether it should be made part of a more general rule along with some of the other rules in your analysis.

Two (or more) rules can be combined into one more general rule if:

(a) The **changes** imposed by the rules are the **same**.

(b) The **environments** where the rules apply are the **same**.

(c) The **targets** of each rule can be **combined into a single, more general segment class** that doesn't incorrectly apply to other sounds in the data set.

If you have two rules that look like they are very similar, but they don't quite match up, try to see if any of the segment classes (target, change, environment) you have stated for any of your rules could be made slightly more (or, if necessary, slightly less) general so that the different rules begin to resemble each other more closely.

B. Predictions about sounds that are not in the data set

If you follow the preferred strategy and state all your segment classes in terms that are as general as is consistent with the data, you may find that sometimes you are making claims about segments that do not appear in the data. This is a **good thing** — what you are in fact doing is **making a prediction** about how such sounds would behave *if they were present*.

The only time you *need* to make the description of a segment class more specific is if there are actual **counterexamples** in the data to the more general description of the class. For example, in the **Arabic consonants** problem, describing the segments in group (d) as "coronal stops" would be *too* general, because [n] is also a coronal stop but it belongs to group (e). That is, [n] is a counterexample to the proposal that the relevant segment class is simply defined by [–cont, COR]. This justifies referring to the more specific segment class [–son, –cont, COR], coronal obstruent stops (alternatively, referring to the property "oral" ([–nas]) would work here too).

Here is an example of a rule that can be made more general, and therefore makes predictions about sounds that are not seen (in the relevant context) in the data set. Going back to the [i]~[e] alternation in Lamba, we note that the crucial factor in the "morphology-free generalization" in (3) seems to be *vowel height*. Now, we might wonder if there is any reason to restrict our statement to apply only to high *front* vowels. In fact, the answer is no. We have no information about high back vowels occurring after mid vowels, so there is no counterevidence against making our rule apply to all high vowels (front and back).

So, let's restate our description in yet more general terms. We are now making a hypothesis about how back vowels behave — a hypothesis that can be tested against further data.

(8) Fully general statement of the Lamba [i]~[e] alternation

In Lamba, high vowels become mid when they are preceded by a mid vowel.

Do we even need to restrict our statement to *high* vowels, or can we say that *all* vowels become mid when a mid vowel precedes? This time, we see that we can't go that far. There are examples in the data set that show that a low vowel remains unchanged even when it follows a

mid vowel. So, we seem to have arrived at the final version of our descriptive statement of the pattern.

Here is how we can state our rule for vowels in Lamba using formal rule notation.

(9) Rule for Lamba vowels

$$[\,+\text{hi}\,] \rightarrow [\,-\text{hi}\,] \quad / \quad \begin{bmatrix} -\text{hi} \\ -\text{lo} \end{bmatrix} \quad C_0 \_\_$$

C. Choosing between equally general ways of stating a rule

Sometimes there are two (or more) ways of stating a rule that are (approximately) equally general and simple. In such a case, see if one or the other version of the rule seems more *insightful*. In particular, does it express something about the phonetic plausibility of the rule — does it indicate something about what is special about that particular feature change given that particular target or that particular environment?

For example, the rule for Lamba vowels in (9) above can actually be written another way, using a different feature specification for the target of the rule: [–lo] instead of [+hi].

(10) Rule for Lamba vowels, second option

$$[-\text{lo}] \rightarrow [\,-\text{hi}\,] \; / \quad \begin{bmatrix} -\text{hi} \\ -\text{lo} \end{bmatrix} \quad C_0 \_\_$$

This second version of the rule will of course apply to mid vowels as well as high vowels, but this is a case of vacuous rule application (mid vowels are already [–hi]), so it's not a problem.

Either way of stating this rule can be considered correct; both are equally general in terms of how many features we must use to designate the relevant segment classes. However, there might be a reason to prefer (10). This statement of the rule emphasizes that the rule wants a *[–lo] vowel* to change in a particular way, if a certain kind of *[–lo] vowel* precedes it. That is, it brings out a connection between the target and the environment of the rule, in a way that isn't as clearly seen in the version of the rule in (9). This version of the rule also highlights the fact that the rule's change, to [–hi], is exactly the change needed to bring the target [–lo] sound into conformity with the [–hi, –lo] sound that is the crucial factor in the environment.