Towards a complete Logical Phonology model of intrasegmental changes[*]

Charles Reiss
Concordia University, Montréal

April 4, 2021

Abstract

All changes to the internal structure of phonological segments arise from combinations of two set-theoretic operations: feature deletion by set subtraction and feature insertion by unification. Once stated, the idea seems trivial—after all, it's obvious that you can make a *d* into a *t* by removing +VOICED and adding −VOICED. However, we'll see that some subtleties are required in the chain of reasoning that will lead us to a full treatment of the complex process of Hungarian voicing assimilation. This example illustrates how the problem of modeling the full range of phonological phenomena can be fruitfully decomposed.

Keywords: phonology, unification, features, rules, segments, underspecification

> We learn not to worry about purpose, because such worries never lead to the sort of delight we seek.
>
> Steven Weinberg, *To Explain the World*

1 Introduction

(Ontological) proliferation is the *enemy* of (theoretical) elegance. In contrast, combinatorics is our *friend*: when we try to model some aspect of the world, we want to account for lots of phenomena with just a few building blocks that can be put together to allow for superficial variety and complexity. One reason for this perspective is that it conforms to the intuition that at some level a theory is a form of "data compression" (Chaitin, 2006). Another reason, for linguistics—and equally applicable to other cognitive sciences—is that the "less attributed to genetic information (in our case, the topic

[*] I am grateful to Maxime Papillon, Alan Bale, Marjorie Leduc, David Ta-Chun Shen, Armel Jolin and Veno Volenec for discussion and corrections of the paper itself, and also for playing a large role in developing many of the ideas here over several years. Péter Siptár deserves special recognition as a pure scientific soul who generously advised me on the data, analysis and exposition of an account of Hungarian at odds with his own.

of UG) for determining the development of an organism, the more feasible the study of its evolution" (Chomsky, 2007) and the more plausible its neural implementation. One strategy for achieving such elegance in linguistics is to carve up a complex domain into isolable pieces, so that from the "welter of descriptive complexity" we can abstract "certain general principles governing computation that would allow the rules of a particular language to be given in very simple forms" (Chomsky, 2000).

But what if our approach is so narrow and reductive as to propose almost nothing about almost nothing? It is perhaps natural that such an anxiety arises from the combination of such a program of abstraction with the substance-free approach I adopt (Hale and Reiss, 2000, 2008; Reiss, 2017b). Substance free phonology denies a role for markedness and perceptual or articulatory considerations in phonological explanation, and initially does not make for a very promising vista of phonology as a domain for seeking intellectual delight: if we get rid of all the phonetics and minimize the computational system, what's left to keep us occupied? I address this anxiety by showing in schematic but explicit fashion how to model a wide variety of superficially diverse phenomena with just two set-theoretic mechanisms. This pared-down inventory does not entail that our analyses will be uninteresting since "[a]s concepts and principles become simpler, argument and inference tend to become more complex—a consequence that is naturally very much to be welcomed" (Chomsky, 1982). Free of any dependency on phonetic substance or teleological notions of purpose, my approach seeks delight in the pure logic of phonological computation.

I propose that the complex problem of modeling phonological phenomena can be fruitfully decomposed, and in this paper I examine only the segment-internal changes effected by the most basic mappings ('rules'). I argue that all changes to the internal structure of segments can be reduced to combinations of **feature deletion by set subtraction** and **feature insertion by unification**. Once stated, the idea seems trivial—after all, it's obvious that you can make a *d* into a *t* by removing +VOICED and adding −VOICED. However, we'll see that some subtleties are required in the chain of reasoning that will lead us to a full treatment the complex phenomenon of Hungarian voicing assimilation at the end of the paper.

In order to appreciate our narrow focus, let's consider various aspects of a simple alternation, and see how they can be distinguished. The process in Lamba (Doke, 1938) that turns an onset /l/ to [n] when the preceding onset contains a +NASAL segment raises issues of what can be a **possible environment** in a rule and how **locality** can be defined in terms of adjacency, linear order, syllable structure, and so on. The process also demands that we consider whether the **change in the features** LATERAL and NASAL happen in parallel as a single rule, or as different steps in the derivation. Even the change in a single feature, say from −NASAL to +NASAL can be analyzed either as a single rule, or as a two-step transformation: deletion of the valued feature −NASAL then insertion of its opposite +NASAL via separate rules. Other processes raise further issues about structural changes in the mappings from Underlying Representations to Surface Representations—for example, metathesis, insertion and deletion of *segments* (as opposed to features) involve manipulation of *sequences* of segments and their **precedence relations** (Raimy, 2000; Papillon, 2020).

In light of all these questions, I focus here on just the narrow topic of **feature changes within segments**, like /l/ surfacing as [n]. Under certain simplifying assumptions, I propose that a complete account of changes within segments uses only the two primitive mechanisms: **deletion** of *valued features* like +NASAL via an operation of **set subtraction**, and **insertion** of *valued features* via an operation of set **unification**.[1] Recent work (Bale et al., 2014; Bale and Reiss, 2018; Bale et al., 2020) has proposed these operations as elements of a basic inventory of computations for phonology: the novelty of this paper is the stronger claim that **just these two operations are necessary and sufficient**. This claim is embedded in the larger framework of assumptions my co-authors and I adopt and justify (or at least acknowledge as temporary idealizations).

(1) Assumptions based on previous work

a. Segment symbols are abbreviations for *sets* of valued features.

b. Segments must be consistent (i.e. they don't contain +F and −F for any feature).

c. Segments need not be complete—segments may lack a specification for some features. Assuming underspecification *simplifies* phonology in several senses that we discuss.

d. Feature insertion rules involve unification (similar to set union) with a singleton set.

e. Feature deletion involves set subtraction and can affect several features at once.

f. Greek letter variables, also known as α-notation, are part of phonology, and not just part of the metalanguage of phonologists—rules may contain variables whose domain is $\{+, -\}$.

g. Targets and environments of rules refer to natural classes, defined in a manner that slightly updates the traditional conception.

h. Rules are functions mapping between phonological representations, and the phonology of a language is a complex function composed of particular rules built from a toolkit of representational and computational primitives. For the most part, the rules developed here can be treated as functions mapping from strings of segments to strings of segments, however, where, e.g., syllable structure is referenced in a rule, mapping between more complex representations will be necessary.

All of these points will be illustrated and elcuidated as we proceed.

2 Background

In this section, I provide more details concerning the list of assumptions given in the previous section.

[1] I explore an alternative with just one operation, priority union, in Reiss (2021).

2.1 Segments are consistent, potentially incomplete feature sets

As a first approximation I treat segments as *sets* of (valued) features. For example, we can represent the labial oral stops p and b as the sets in (3ab). Among the simplifications adopted here are the following:

(2) Three simplifications

a. Not all features that are present are necessarily shown. This is just a matter of typographical convenience.

b. An actual feature set in a morpheme's lexical representation may be associated with a timing slot, and the segment might be thought of as the timing slot combined with the feature set. Since our concern here is the range of segment-internal changes, we can ignore timing slots, tones, syllable structure, etc., here.

c. We abstract away from contour segments like affricates, which potentially have a more complex structure (see Shen 2016 for discussion).

Point (2a) is important to distinguish from our use of underspecification. The set of features in (3c) corresponds to neither /p/ nor /b/, but to a third possible segment, *B*, a labial stop lacking any specification for the feature VOICED.

(3) Three labial oral stops

a. /p/ = {−VOICED, −SONORANT, −CORONAL, +ANTERIOR, +LABIAL, −NASAL, −CONTINUANT}

b. /b/ = {+VOICED, −SONORANT, −CORONAL, +ANTERIOR, +LABIAL, −NASAL, −CONTINUANT}

c. /B/ = {−SONORANT, −CORONAL, +ANTERIOR, +LABIAL, −NASAL, −CONTINUANT} (no value for VOICED)

These segments all fulfill our assumed condition that segments be *consistent*, that is that they not contain both +F and −F for any feature F. The possibility of a segment like *B* shows that we accept the existence of underspecification: segments can be *incomplete*, lacking any specification for one or more features. Further discussion of the empirical justification for underspecification, building on previous work (e.g. Keating, 1988; Choi, 1992) can be found in Bale and Reiss (2018) and related work.

2.2 Natural classes

Given our treatment of segments as sets of features, natural classes of segments, which are sets of segments, are fundamentally just sets of sets of features. For example, in a language containing the three oral labial stops, *p,b,B*, this natural class might be characterized as the set of segments whose members are all and only the segments in the language that are supersets of this set of features:

(4) Set of features that define the natural class containing only *p,b,B*
{−Sonorant, −Coronal, +Anterior, +Labial, −Nasal, −Continuant}

So, the set Q of labial stops in language L is the following

(5) $Q = \big\{q : q \supseteq \{-\textsc{Sonorant}, -\textsc{Coronal}, +\textsc{Anterior}, +\textsc{Labial}, -\textsc{Nasal}, -\textsc{Continuant}\ \}\ (\text{and } q \in L)\big\}$

Because this notation is unwieldy, we make use of traditional square bracket notation to denote classes of segments in the target and environment of rules. The brackets show that natural classes are of a different type from the set of features in the structural change of a rule.

(6) $$Q = \begin{bmatrix} -\textsc{Sonorant} \\ -\textsc{Coronal} \\ +\textsc{Anterior} \\ +\textsc{Labial} \\ -\textsc{Nasal} \\ -\textsc{Continuant} \end{bmatrix}$$

We also leave implicit the second condition given in (5), the one that requires that a segment be in the inventory of the language in question—just keep in mind that the intensional natural class description in (6) may have the extension {*p,b,B*} in one language, but maybe {*p,b*} or {*p*} in another.

The treatment of natural classes sketched relies on the superset (or subset) relation. This fails for more complex representations, including ones that do not correspond to just single segments and those that involve α-notation (Bale and Reiss, 2021). We'll ignore these complications in this paper.

2.3 Basic set operations

The basic set operations I attribute to phonological rules are *set subtraction* and *unification*. Set subtraction here has the standard definition: If A and B are sets, then $A - B = C$, where $C = \{c : c \in A \text{ and } c \notin B\}$. That is, the members of $A - B$ are all the elements of A except for those that are in B. Here are some examples:

(7) Examples of set subtraction

- $\{x, y\} - \{x, z\} = \{y\}$
- {+Voiced, −Sonorant, +Coronal} − {+Voiced, +Sonorant} = {−Sonorant, +Coronal}

The second operation I use is unification, which is related to the familiar operation of set union. Set *union* combines the members of sets into a single set: If A and B are sets, then $A \cup B = C$, where $C = \{c : c \in A \text{ or } c \in B\}$. Union is called a total operation since the union of any two sets always yields a set. Instead of union, we use a version of unification, which is typically used to combine expressions that are more complex than sets. Unification is useful here because the members of the segment sets are actually ordered pairs of values (‘+’ or ‘−’) combined with features. For a feature F, we refer to the valued features +F and −F as ‘opposites’. Unification $\sqcup$ requires the phonology to look into the valued features that are the elements of feature sets. If A and B do not contain opposites, then $A \sqcup B = A \cup B$. However, if A and B do contain any conflicting values, that is if $A \cup B$ is not consistent, then $A \sqcup B$ is *undefined*.

(8) Unification examples

- $\{+F,+G\} \sqcup \{+G,-H\} = \{+F,+G,-H\}$ (Recall that the elements of a set are unique: $\{a,b,c\}$ is the same set as $\{a,a,a,b,c,c\}$.)
- $\{+F,+G\} \sqcup \{-G,-H\}$ is undefined, because $\{+F,+G\} \cup \{-G,-H\} = \{+F,+G,-G,-H\}$ which contains the opposites $+G$ and $-G$.

Because the output of unification may be undefined, it is called a *partial* operation, unlike normal union which always yields an output set. When the result of unification is undefined we refer to *unification failure*. When unification failure occurs in the application of a phonological rule, we assume that the rule then maps the input string to an identical output.

2.4 Logical relations among segments in patterns

Our discussion makes use of three abstract segment symbols $\phi \neq \psi \neq \Delta$ corresponding to three different sets of valued features defined as follows. Unless otherwise noted, Δ is unspecified for some (unique) feature F, but is otherwise identical to ϕ and ψ, which are identical to each other, aside from having opposite values for F. For concreteness, an example of such a triplet of segments would be *p,b,B*, where, as above, *B* is a labial stop unspecified for voicing.

(9) Three abstract segments

- $\exists!$ F s.t. ϕ and ψ differ w.r.t. F

  The two segments differ w.r.t. exactly one feature.
- $\Delta = \phi - \{-\text{F}\}$ and $\Delta = \psi - \{+\text{F}\}$

  If you take −F from ϕ you get Δ, and if you take +F from ψ you get Δ (or vice versa).
- $\phi \cap \psi = \Delta$

  The intersection of the two fully specified segments is the underspecified one. This follows from the definition of Δ.
- E.g. *p,b,B*

The symbols ϕ, ψ, Δ are just part of our metalanguage for discussing phonology. These three segment variables represent a minimal case of representational distinctiveness, with opposite values and presence vs. absence restricted to a single feature. The logic presented here trivially extends to richer representational contrasts involving more than one feature. It is not only methodologically useful to start out with the simplest cases, but it is also impressive to see how much empirical coverage such cases afford—we get a lot of empirical coverage without a lot of machinery.

2.5 Greek letter variables—α-notation

The use of symbols like ϕ, ψ, Δ is distinct from the use of Greek letters as the familiar α-notation variables with the domain $\{+, -\}$, to which I now turn. Most importantly, whereas the previous discussion involved variables of our metalanguage, I assume that Greek letter variables *do* occur in phonological expressions (*pace* McCawley 1971), and they express dependencies among parts of a phonological representation. However, such variables cannot occur *alone*, unmatched in a linguistic expression to denote existence of a feature value. In other words, I reject the use of, say, αNASAL to mean “some specified value of nasal" (see Bale and Reiss, 2018, 454). I only use α-notation to express agreement or mutual dependency, the identity of values on tokens of a given feature. A valid use of α is, for example, to say that all obstruents in a cluster must be αVOICED, meaning that they must all be +VOICED or all be −VOICED. It is formally quite different to use variables to denote the mere existence of a valued feature in a segment set, and I posit that this mechanism does not exist in the phonological faculty, and I similarly reject symbols like ±VOICED. Relatedly, there are no lexical items that contain Greek letter variables, contra, for example, Kiparsky (1985, p.92). The “existential” use of variables appears to be unnecessary, and it also would wreak havoc with natural class logic in phonology. Using our example of *p,b,B*, without existential variables we cannot refer to the fully specified stops as those containing αVOICED and thereby make a natural class of *p* and *b* to the exclusion of *B*; and there is also no way to group *B* into a class with either one of the other two to the exclusion of the remaining one (Bale et al., 2014). This appears to be a desirable outcome.

2.6 Segment mapping diagrams

I treat phonological rules as functions mapping (in the simple cases considered here) strings of segments to strings of segments. As explained in Bale and Reiss (2018), we need to reject the informal traditional notion that rules map segments to segments (in a context) because this approach does not allow the rules of a language to compose into a single function that computes each derivation. Despite this view of rules, the following discussion makes use of segment mapping diagrams (SMDs) to illustrate relations among segments that appear in URs and SRs. These diagrams are just aids to exposition. In the simplest case, where there are no relevant phonological rules, the SMD showing the mapping between segments in the input and segments in the output is straighforward, as in this case of a language with segments *s* and š which are not involved in any phonological interaction. Again, the SMDs are not part of human phonological grammar, and they are not part of our theory: they are visual aids for recognizing logical patterns that arise from different combinations of basic rules.

(10) Simple SMD

$$\begin{tikzcd} \text{s} \arrow[d] & \text{š} \arrow[d] & \text{Underlying segments} \\ \text{s} & \text{š} & \text{Surface segments that show effects (if any) of the PHONOLOGY} \end{tikzcd}$$

The top row of the diagram shows segments that appear in URs, and the bottom row shows segments that occur in SRs. The arrows show the relations between the segments, with vertical arrows (most typically) showing identity mappings (or else the 'evolution' of a segment that doesn't undergo split or merger with another segment). The diagram for (10) represents the very boring fact that underlying /s/ maps to surface [s] and underlying /š/ maps to surface [š].

An SMD for a language with a neutralization rule that merges *s* and š in some context looks like this, with diagonal arrows showing non-identity mappings due to the effects of rules.

(11) Segment mapping diagram (SMD) for neutralization:

$$\begin{tikzcd}
s \arrow[d] \arrow[dr] & \check{s} \arrow[d] & \text{Underlying segments} \\
s & \check{s} & \text{Surface segments}
\end{tikzcd}$$

The diagonal arrow shows, on the one hand, the split of underlying /s/ into [s] and [š], and on the other hand, the partial merger of underlying /s/ and /š/ into [š]. As we will see, more elaborate SMDs can be used to track the derivational history of segments through multiple rules. In such diagrams, there will be intermediate levels between URs and SRs. Section 4 will show how a wide range of attested processes can be reduced to combinations of rules based on set subtraction and unification and described with such SMDs.

3 The basic operations in rules

Since the focus in this paper is on segment internal changes, I adopt a simplified picture of rule environments. All the rules identify a target segment in input sequences that are defined by reference to at most one preceding and one following segment, or else by such a sequence enhanced by reference to syllable structure position such as ONSET or CODA. My rule syntax is close to a simple version of *SPE* rule syntax, as developed in Bale and Reiss (2018).

Phonological processes are sometimes categorized as feature-filling vs. feature-changing. As the names suggest, a feature-filling process might replace sequences containing *B* by sequences containing *b* in the corresponding position, whereas a feature-changing process might replace sequences containing *p* by sequences containing *b* in the corresponding position. I assume that 'processes' like this have no theoretical status, despite the fact that they are sometimes referred to as feature-changing vs. feature-filling *rules*. Instead, feature-filling processes are analyzed here as rules built from the unification operator, and feature-changing processes are analyzed as sequences of two or more rules—first a deletion rule built from the subtraction operator, followed by at least one feature-filling rule built from the unification operator. In brief, feature changing involves deletion of a valued feature and insertion of its opposite, in separate rules.

3.1 Unification rules

Feature-filling rules consist of a natural class description of a target set of segments followed by the unification symbol (in place of the normal arrow symbol '→'), followed by the set which is the second argument of the unification operator (the set of valued features with which each member of the target set potentially unifies). For the environment segments, to the right of the slash '/' I use κ, λ, defined as natural class descriptions of sets of segments as in (5). Using these symbols we can express a rule based on unification as in (12).

(12) $\begin{bmatrix} -\textsc{Son} \\ -\textsc{Cont} \end{bmatrix} \sqcup \{+\textsc{Voiced}\}$ / κ___λ

Developing the model sketched by Bale and Reiss (2018) the interpretation of this rule is the following:

(13) Interpretation of rule (12)
Rule (12) is the function that maps any finite string of segments $x_1x_2 \dots x_n$ to the string of segments $y_1y_2 \dots y_n$ such that for each index i that is greater than or equal to 1 and less than or equal to n $(1 \leq i \leq n)$...

- IF $x_i \supseteq \{-\textsc{Son}, -\textsc{Cont}\}$ and $x_{i-1} \supseteq \kappa$ and $x_{i+1} \supseteq \lambda$ THEN
  IF $x_i \cup \{+\textsc{Voiced}\}$ is consistent, $y_i = x_i \cup \{+\textsc{Voiced}\}$.
- OTHERWISE $y_i = x_i$

Note that the rule syntax makes use of unification, but the interpretation of the rule breaks unification down into normal set union along with the notion of consistency, as discussed above.

Let's illustrate the application of such a rule by considering how it will affect input sequences containing *p,b* and *B* in the relevant environment. Let k, l each be a segment belonging to the class denoted by κ, λ, respectively.

(14) Applying rule (12)

a. If the input string is kBl then, since $B \sqcup \{+\textsc{Voiced}\} = b$, the output of the rule is kbl.

b. If the input string is kbl then, since $b \sqcup \{+\textsc{Voiced}\} = b$, the output of the rule is kbl. The rule applies vacuously since unification applies vacuously here, and the rule maps kbl to kbl.

c. If the input string is kpl then we need to consider the result of $p \sqcup \{+\textsc{Voiced}\}$. Here unification is **undefined**, because $p \cup \{+\textsc{Voiced}\}$ is not consistent—the output of set union contains both +VOICED and −VOICED as members. This is a case of unification failure, and the rule semantics given in (13) yields an identity mapping in such cases. The output of the rule is kpl. This is another case of vacuous application since the rule maps kpl to kpl.

d. The effect of the rule on any other sequence, say, *banana*, is null—the output string will be identical to the input. There is no case of an oral stop preceded by a member of class κ and followed by a member of class λ. This is also a kind of vacuous application.

To reiterate, the application of a phonological rule built from unification can be vacuous for three reasons: either the unification itself is vacuous, as in input kbl; or unification fails, as in input kpl; or rule conditions aren't met, as in input $banana$. Contrary to most of the literature, we do not say, in the last case, "the rule did not apply". In order to compose the rules into a single function, we need each rule to map every input to an output. In all three cases the rule application leads to an identity of input and output. That is all we need here to understand rules built with the unification operator.[2]

3.2 Set subtraction rules

Let's now look at a rule built with the set subtraction operator. Instead of using the unification symbol in place of the traditional arrow '→', as we did above, this rule uses the set subtraction symbol '−'. Part of the motivation for the Logical Phonology approach I adopt is the recognition that phonological rules using '→' actually encode very different basic operations. Our approach does away with such ambiguity and obviates the need for rule diacritics such as 'feature-filling' vs. 'feature-changing' labels invoked by authors such as McCarthy (1994).

(15) Subtraction of {−VOICED} from an oral stop

$$\begin{bmatrix} -\textsc{Son} \\ -\textsc{Cont} \end{bmatrix} - \{-\textsc{Voiced}\} \mathbin{/} \kappa \underline{\quad} \lambda$$

The other aspects of the rule match the example of the unification-based rule above.

We can now consider the interpretation of this rule built with the subtraction operator.

(16) Interpretation of rule (15)

Rule (15) is the function that maps any finite string of segments $x_1x_2 \ldots x_n$ to the string of segments $y_1y_2 \ldots y_n$ such that for each index i that is greater than or equal to 1 and less than or equal to n $(1 \leq i \leq n)\ldots$

- IF $x_i \supseteq \{-\textsc{Son}, -\textsc{Cont}\}$ and $x_{i-1} \supseteq \kappa$ and $x_{i+1} \supseteq \lambda$
  THEN $y_i = x_i - \{-\textsc{Voiced}\}$
- OTHERWISE $y_i = x_i$

Let's illustrate the application of this rule using the same sequences we used for the unification rule above.

(17) Applying rule (15)

[2] In other work (Bale et al., 2020), we argued that the second argument in unification rules is always a singleton set, containing just one valued feature, but that aspect of unification rules is not crucial in this paper.

a. If the input string is kBl then, since B – $\{-\text{VOICED}\}$ = B, the output of the rule is kBl, so the rule applies vacuously.

b. If the input string is *kbl* then, since *b* – $\{-\text{VOICED}\}$ = *b*, the output of the rule is kbl. The rule applies vacuously here, as well, since subtraction is once again vacuous.

c. If the input string is kpl then, we need to consider the result of p – $\{-\text{VOICED}\}$. The result is B so the output is kBl.

d. The effect of the rule on any other sequence, say, *banana*, is null—the output string will be identical to the input.

The rule applies non-vacuously only in case (17.c).[3]

3.3 Illustration of feature-filling and feature-changing

We can now see that a unification rule like (12) is sufficient to model a feature-filling process, for example by filling in +VOICED on a /B/ and leaving underlying /p/ and /b/ unchanged.

(18) A feature-filling process

UR	kBl	*kbl*	*kpl*
Unification Rule (12)	*kbl*	—	—
SR	*kbl*	*kbl*	*kpl*

The dashes '—' in this table reflect vacuous rule application. For input *kbl* the mapping is vacuous, because unification is vacuous, since *b* is already specified +VOICED. For input *kpl* the mapping is vacuous because unification fails, because p is specified −VOICED, and so unification with {+VOICED } by rule (12) is undefined. I discuss this further below.

To get a feature-changing process, we need to first apply a subtraction rule like (15) followed by a unification rule. Such a sequence of rules, (15) followed by (12), applied to kBl, *kbl* and *kpl* will yield the following derivations:

(19) A feature-changing process

UR	kBl	*kbl*	*kpl*
Subtraction Rule (15)	—	− − −	kBl
Unification Rule (12)	*kbl*	—	*kbl*
SR	*kbl*	*kbl*	*kbl*

We will see that combining these simple mechanism with judiciously chosen underlying forms and the use of α-notation will allow us to model a wide range of phenomena.

[3] The use of set substraction in phonological rules appears to allow the second argument to contain more than one element—there is no restriction to singleton sets (Bale et al., 2020). However, given our simple examples in this paper, this detail will play no further role.

3.4 Exploiting unification failure to streamline grammar

Unification failure often turns out to allow quite streamlined accounts of phonological processes. We'll start with a toy example and see more realistic cases later. Suppose a language has a three-way contrast of /p,b,B/, and that /B/ is the only segment underspecified for the feature VOICED. Suppose further that there is a process whose effect is that /B/ surfaces as [b] between vowels, but that /p/ and /b/ surface unchanged. We need a feature filling rule to insert +VOICED into /B/ and yield [b]. Here's a first version of such a rule:

(20) First version of rule voicing /B/ to [b]

$$\begin{bmatrix} +\textsc{Lab} \\ -\textsc{Son} \\ -\textsc{Cont} \end{bmatrix} \sqcup \{+\textsc{Voiced}\} \,/\, \begin{bmatrix}\ +\textsc{Syll}\ \end{bmatrix} \underline{\quad} \begin{bmatrix}\ +\textsc{Syll}\ \end{bmatrix}$$

This works, but note that unification of every other consonant in the language aside from /B/ with the set {+VOICED} will be vacuous, either because that consonant is already specified +VOICED or because it is specified −VOICED and unification with {+VOICED} will fail. So there is no need to specify the place of articulation of the target class as +LABIAL. We can streamline the rule to this second version:

(21) Second version of rule voicing /B/ to [b]

$$\begin{bmatrix} -\textsc{Son} \\ -\textsc{Cont} \end{bmatrix} \sqcup \{+\textsc{Voiced}\} \,/\, \begin{bmatrix}\ +\textsc{Syll}\ \end{bmatrix} \underline{\quad} \begin{bmatrix}\ +\textsc{Syll}\ \end{bmatrix}$$

This also works, but since, by assumption all segments other than /B/, vowels and consonants, are specified for either +VOICED or −VOICED, it is possible to make the rule even more general:

(22) Third version of rule voicing /B/ to [b]

$$\begin{bmatrix}\quad\ \end{bmatrix} \sqcup \{+\textsc{Voiced}\} \,/\, \begin{bmatrix}\ +\textsc{Syll}\ \end{bmatrix} \underline{\quad} \begin{bmatrix}\ +\textsc{Syll}\ \end{bmatrix}$$

As pointed out in Bale et al. (2020), this means "unify every segment that is a superset of the empty set with the set {+VOICED} if the segment occurs between vowels." Since basic set theory tells us that the empty set is a subset of every set, the rule unifies every segment occurring between vowels with {+VOICED}. Under our assumption that all segments but /B/ are specified for voicing, the rule will have no effect on other segments.

We can take this a step further. If it is truly the case that only /B/ is underspecified for voicing at this point in the derivation, then there is no need for a context for the rule:

(23) Fourth version of rule voicing /B/ to [b]

$$\begin{bmatrix}\quad\ \end{bmatrix} \sqcup \{+\textsc{Voiced}\}$$

This means "unify every segment that is a superset of the empty set with the set {+VOICED}." Under our assumption that all segments but /B/ are specified for voicing, the rule will apply vacuously to every other segment, in all contexts.

The interpretation of this rule is the following:

(24) Interpretation of rule (23)
Rule (23) is the function that maps any finite string of segments $x_1x_2\ldots x_n$ to the string of segments $y_1y_2\ldots y_n$ such that such that for each index i that is greater than or equal to 1 and less than or equal to n $(1 \leq i \leq n)\ldots$

- IF $x_i \cup \{+\text{VOICED}\}$ is consistent, THEN $y_i = x_i \cup \{+\text{VOICED}\}$.
- OTHERWISE $y_i = x_i$

Of course, this rule might be ordered at the end of the derivation, after other tokens of underlying /B/ have been assigned a value for VOICED by other rules. Rule (23) will fill in a default value for remaining underspecified tokens. In the following discussion this kind of economy will be exploited to varying degrees. I will not provide the interpretations for the rules in the remainder of the paper since they involve trivial extension of the examples provided so far.

I have insisted that rules are functions that map any input representation to an output representation. Sometimes an input representation will be mapped to an identical output. This is necessary in order to treat the whole phonology as a composed function. The examples of rule semantics for a unification-based rule and a subtraction-based rule have been formulated in terms of such functions. However in discussion, it is often convenient to focus on the mapping of target segments from input to output. This expository expedience should not be taken as a rejection of the nature of the rules indicated by the given interpretations. I will not continue to provide the interpretation for each rule, but it should be trivial to do so based on the examples we have seen. As shown in Bale and Reiss (2018), rule semantics become more complex once segment insertion and deletion (not the *feature* insertion and deletion considered here) are considered. This is because it becomes necessary to map between strings of different lengths, and it is not trivial to determine the correspondence between input and output segments in the two strings.

4 Combining the primitives

In this section I provide a schematic SMD using the segment variable symbols ϕ, ψ and Δ along with a description of the corresponding composed rules. In each case a simple toy example or actually attested case is given for concreteness.

4.1 Simple neutralization

The SMD in (11) showing the partial neutralization of /s/ with /š/ did not take into account the discussion in section 3.3 about the two-step analysis of feature-changing processes. In order to reflect this view, a more complete SMD for simple neutralization would have an extra level as in (25).

(25) Explicit SMD for feature-changing neutralization of ϕ and ψ

$$\begin{tikzcd}
\text{UR} & \phi \arrow[d] \arrow[dr] & & \psi \arrow[d] \\
& \phi \arrow[d] & \Delta \arrow[dr] & \psi \arrow[d] \\
\text{SR} & \phi & & \psi
\end{tikzcd}$$

As the SMD shows, in some environment, a first subtraction rule deletes the valued feature on ϕ with respect to which ϕ disagrees with ψ. This turns ϕ into Δ (which we are assuming is not present in any URs). In the next step, a unification rule inserts the valued feature which is the opposite of the previously deleted one into the newly created Δ, yielding ψ. This is simple conditioned neutralization of two fully specified underlying segments.

However, I will not generally reiterate in the SMDs *all* the vacuous mappings or show the inventory at each stage, so a less explicit version of (25), like (26) will sometimes be used.

(26) Less explicit SMD for feature-changing neutralization of ϕ and ψ

$$\begin{tikzcd}
\text{UR} & \phi \arrow[dd] \arrow[dr] & & \psi \arrow[dd] \\
& & \Delta \arrow[dr] & \\
\text{SR} & \phi & & \psi
\end{tikzcd}$$

I will also represent distinct rules at the same level in an SMD if there is no way to explicitly order them (as in the discussion of Turkish stops below).

4.2 Feature-changing allophony of ϕ and ψ

If a language has a sequence of rules just like in the previous case, but without having any underlying tokens of ψ, then we have a classic case of allophony, with ϕ and ψ in complementary distribution on the surface.

(27) Feature-changing allophony

$$\begin{tikzcd}
\text{UR} & \phi \arrow[dd] \arrow[dr] & & \\
& & \Delta \arrow[dr] & \\
\text{SR} & \phi & & \psi
\end{tikzcd}$$

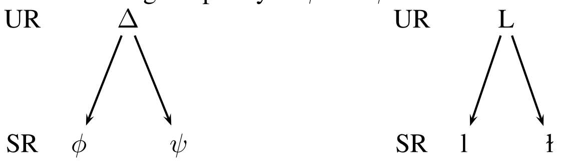
The only difference from the previous case is that there are no underlying tokens of ψ and thus there is no identity mapping to surface ψ. The rules would be the same as in the previous case—delete a valued feature from ϕ in certain environments to derive Δ, then insert the opposite value to derive ψ from Δ. This analysis is justified when

ψ occurs in a natural class of environments, but where ϕ does not and must be posited as the underlying form. The SMDs in (26) vs. (27) serve as a visual aid to see that the latter differs from the former just by virtue of having an additional identity mapping—which in turn is a result of having a different set of underlying segments present in the lexicon.

4.3 Feature-filling allophony of ϕ and ψ

In some cases, it is not possible to choose an underlying form based on elsewhere-case reasoning. In other words, it may be clear that ϕ and ψ are in complementary distribution, but it is an arbitrary decision as to which should be posited as underlying. Rather than make an arbitrary choice as a linguist, and rather than attribute an arbitrary choice between extensionally equivalent and equally elegant grammars to the learner, we can posit that a learner encodes an underlying form that is neither ϕ nor ψ, namely Δ. This situation can be called feature-filling allophony, since both surface forms are derived by feature-filling from the UR.

(28) Feature-filling allophony of ϕ and ψ

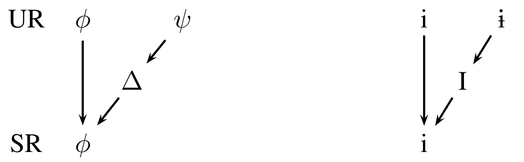


An example of this situation might be found in the English distribution of laterals. In some English dialects, dark [ɫ] occurs in codas and light [l] occurs in onsets, so both can be derived from a lateral unspecified for one (or more) features. Strictly speaking, the SMD in such a case should have a separate level corresponding to each of the two rules (one for feature-filling in onsets and another in codas), but since it is not possible to determine the mutual ordering of the two rules, we'll maintain the simple SMD of (28).

4.4 Absolute neutralization

If we flip the SMD for feature-changing allophony (27) upside down we get a situation that initially looks implausible. There are two underlying segments, but only a single surface correspondent. This SMD models the logical structure of feature-changing absolute neutralization which potentially corresponds to situations like the harmonically irregular vowels of Hungarian, like the [iː] in *híd* 'bridge' that takes back vowel suffixes as in dative *hídnak*, as opposed to the regular *víz* 'water' with dative *víznek*.

(29) Feature-changing absolute neutralization of ϕ and ψ with Hungarian parallel



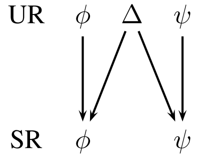
- Rules: A subtraction rule for ψ-to-Δ and a unification rule for Δ-to-ϕ

The SMD is just like that of feature-changing neutralization in (26), aside from the lack of an identity mapping for ψ. In other words, the SMD represents the old idea that the harmonically irregular vowels are the +BACK version of /i/, namely /ɨ/. The vowel /ɨ/ triggers harmony as expected, then a subtraction rule deletes the feature +BACK from it, and a unification rule inserts −BACK into the derived, underspecified vowel. What we gain by this discussion—what makes it go beyond just restating an old idea—is the demonstration that, despite our pre-theoretical prejudice against abstractness, such a solution is not formally complex, and it uses machinery that is needed for other phenomena. The theory predicts the existence of such cases, and formalization of this sort helps us overcome intuitive biases.

4.5 Three-to-two feature-filling mapping

The next pattern is just like feature-filling allophony (28) but with the addition of ϕ and ψ in the lexicon (and thus in URs). As we see in (30) all three segments ϕ, ψ and Δ are present in URs. Feature filling rules for each context fill in either +F or −F to derive ϕ and ψ, but Δ never surfaces.

(30) Three-to-two feature-filling mapping



- Rules: Two unification rules—one for Δ-to-ϕ and one for Δ-to-ψ

This structure corresponds to Inkelas and Orgun's (1995) analysis of Turkish, where it is necessary to posit a three-way contrast in root final stops, even though there are only two stops on the surface. Some roots like *sanat* 'art' have non-alternating [t] as the final segment, so the lexical form can be posited with a /t/. Others have non-alternating [d], as in *etüd* 'étude', so the lexical form has /d/. For roots with surface forms like *kanat-/kanad-* we can posit a final /D/, a coronal stop with no specification for VOICED. In such roots, the /D/ surfaces as [d] in onsets, as in *kanadım*, and as [t] in codas, as in *kanatlar*.

(31) Turkish stops

a. Non-alternating voiceless: −VOICED /t/
[sanat] ‘art’, [sanatlar] ‘art-plural’, [sanatɯm] ‘art-1sg.poss’

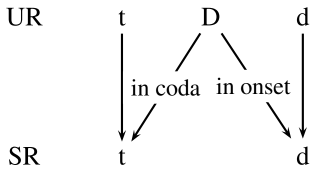
b. Non-alternating voiced: +VOICED /d/
[etyd] ‘etude’, [etydler] ‘etude-plural’, [etydym] ‘etude-1sg.poss’

c. Alternating: (no specification for VOICED) /D/
[kanat] ‘wing’, [kanatlar] ‘wing-plural’, [kanadɯm] ‘wing-1sg.poss’

For Turkish, we need one feature-filling unification rule to insert +VOICED in onsets, affecting only /D/; and we need another feature-filling unification rule to insert −VOICED in codas, again affecting only /D/. To reiterate an important point, formalization allows us to see how this pattern is just a combination of what we needed for the English /L/ analysis with the addition of the identity mappings we have also made use of.

In (32) we repeat the logical structure shown in (30) but with the Turkish segments filled in, along with the environments of the two feature-filling rules responsible for the non-identity mappings.

(32) Segment mapping (non-crucial rule ordering ignored here)

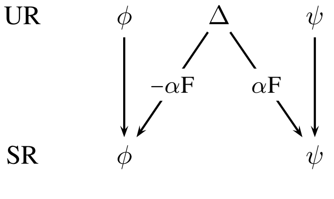


As mentioned above, we assume that all rules are ordered in a grammar, but the two rules here cannot be ordered by the analyst.

4.6 Three-to-two with α-notation

We now turn to an SMD that has the exact same three-to-two structure of the previous example involving Turkish stops, but where we can make use of a single rule to account for both mappings by introducing α-notation. This SMD is relevant when the environments that determine the mappings Δ-to-ϕ and Δ-to-ψ can be characterized in terms of opposite values of the feature that distinguishes ϕ and ψ.

(33) Three segments mapping to two with α-notation



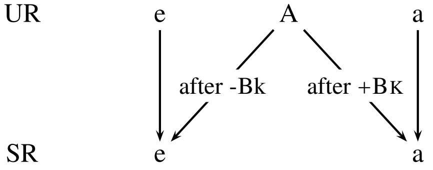
This SMD models the harmonic alternations of Turkish vowels seen in the plural suffix:

(34) Turkish plurals

Trigger vowel	SINGULAR	PLURAL	GLOSS	Suffix form
[i]	ip	ipler	‘rope’	[ler]
[e]	ek	ekler	‘joint’	
[y]	gül	güller	‘rose’	
[œ]	öç	öçler	‘revenge’	
[ɯ]	kıl	kıllar	‘body hair’	[lar]
[a]	sap	saplar	‘stalk’	
[u]	pul	pullar	‘stamp’	
[o]	son	sonlar	‘end’	

The alternating vowel can be posited to be underlyingly /A/, a non-high, non-round vowel, lacking specification for BACK.

(35) Segment mapping with an α rule



So, there are two diagonal arrows in (35) but they reflect the mapping of a single rule (36) that uses α-notation.

(36) $[-\textsc{High}, -\textsc{Round}] \sqcup \{\alpha\textsc{Back}\}$ / when preceding vowel is $[\alpha\text{Back}]$

Application of this rule to a sequence with fully specified /e/ or /a/ in target position will result in vacuous application, either via vacuous unification or unification failure. If /A/ is the target, the rule will fill in a value for BACK according to the context.

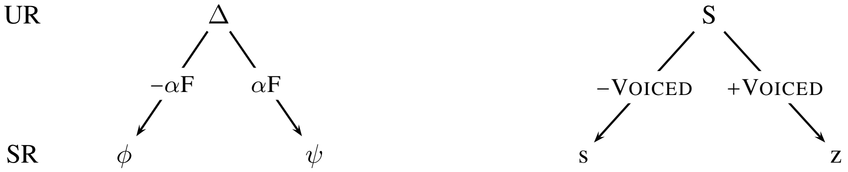
The SMDs in (32) and (35) show the similar structure of the two cases when we abstract from the details of the particular rules.

4.7 A predicted allophonic pattern

The examples of the Turkish stops and vowels just given lead to the consideration of similar patterns involving allophony, rather than neutralization. Note that the analysis of the Turkish stop /D/ is parallel to the syllable structure based analysis of the lateral allophones of English given in section (4.3). It should be possible to have a language like Turkish, but without the underlying *t* and *d*—and in fact, that’s the structure of the English lateral pattern. In principle, we should expect that another possible pattern would be an allophonic version of the Turkish vowel pattern, that is, a mapping from an underspecified segment to two allophones via a single rule that uses α-notation. In (4.3) we see feature-filling allophony, but we need two rules, one to generate clear [l] in onset position and another to generate dark [ɫ] in codas, from an underlyingly underspecified /L/.

I haven't yet found such a case based on α-notation where the value on a feature F in the environment determines the value of F on the two surfaces allophones. This requires the two surface forms to both occur in a natural class of environments definable by α-notation, with no apparent elsewhere case.[4] A hypothetical case would be a language where, say, [s] and [z] occur only as the first member of a cluster, with both derived from underspecified /S/ and the appearance of each two dependent on the voicing of the following segment. The SMD would look like this:

(37) One segment mapping to two with α-notation



If we allow cross-featural use of α-notation, then it is possible to find such patterns, for example, the distribution of [l] and [d] in Tswana. According to Odden (2013, 22) in Tswana "there is no contrast between [l] and [d]. Phonetic [l] and [d] are contextually determined variants of a single phoneme: surface [l] appears before nonhigh vowels, and [d] appears before high vowels (neither consonant may come at the end of a word or before another consonant)." Obviously these two segments differ by more than a single feature, but the pattern of interest can be derived by a feature-filling rule that inserts ±Lateral according to the value for High on the following vowel:

(38) $[+\textsc{Coronal}] \sqcup \{\alpha\textsc{Lateral}\}$ / ___ $[\text{-}\alpha\textsc{High}]$

Odden points out that the Tswana surface distribution can be handled by choosing either segment as the underlying form: "There is no evidence to show whether the underlying segment is basically /l/ or /d/ in Tswana, so we would be equally justified in assuming either" a rule that turns /d/ to [l] or a rule that turns /l/ to [d], since "[s]ometimes, a language does not provide enough evidence to allow us to decide which of two (or more) analyses is correct". Odden is writing here in Chapter 2 of an undergraduate textbook, so he doesn't mention the third option, suggested here, that neither of the surface forms is identical with the underlying form. It is hard to imagine how allophonic patterns derived from underspecified segments could be excluded in any systematic manner, so the formal approach we have adopted helps us to formulate a topic for empirical exploration—a non-cross-featural α-rule example might turn up any day.

[4] This condition, the absence of an elsewhere case, might be too strict. Consider the alternating nasal segment appearing at the end of certain prefixes in Malay, such as [mem-, men-, meŋ-]. These alternants are distributed according to the place of articulation of the root initial segment. Simplifying a bit, [mem-] occurs before labial obstruents, [men-] occurs before coronal obstruents, and [meŋ-] occurs before velar obstruents and before vowels. It is an open question whether the velar nasal before vowels is a direct reflex of the underlying segment, thus, an elsewhere form, or else the result of a set of default, feature fill-in rules.

4.8 Reciprocal neutralization

Another pattern that can be usefully identified using symbols in the logical relationship of ϕ, ψ, Δ is reciprocal neutralization. This term applies to a situation where there is good evidence to posit an underlying ϕ that surfaces sometimes as ψ, and there is also evidence to posit an underlying ψ that surfaces sometimes as ϕ. In light of our account of feature-changing as a sequence of feature deletion via a rule built from set subtraction followed by insertion via a rule built from unification, the transformation from ϕ to ψ and vice versa is most directly modeled as passing through Δ, as in (39):

(39) Schematic reciprocal neutralization

$$\begin{tikzcd}
\text{UR} & \phi \arrow[dd] \arrow[dr] & & \psi \arrow[dd] \arrow[dl] \\
& & \Delta \arrow[dl] \arrow[dr] & \\
\text{SR} & \phi & & \psi
\end{tikzcd}$$

This generic pattern can be illustrated with separate rules or with a single set of rules using α-notation. In the next two sections, I present a simplified example using α-notation, based on Hungarian voicing assimilation. I then develop this account for a more complete version of voicing assimilation in Hungarian consonant clusters. The SMD for reciprocal neutralization shows that it just is a combination of mappings like allophonic α rules (37) and something like the Turkish vowel harmony rule (36).

5 Simplified Hungarian Reciprocal Neutralization

Hungarian[5] shows patterns of neutralization that can be illustrated with the behavior of /t/ and /d/:

(40) Reciprocal neutralization SMD:

$$\begin{tikzcd}
t \arrow[d] \arrow[dr] & d \arrow[d] \arrow[dl] \\
t & d
\end{tikzcd}$$

Both /t/ and /d/ surface unchanged in final position, as in *kút* [kuːt] ‘well’ vs. *kád* [kaːd] ‘tub’.

(41) Reciprocal neutralization in Hungarian

Noun	In N	From N	To N	
kuːt	kuːdbɔn	kuːttoːl	kuːtnɔk	‘well’
kaːd	kaːdbɔn	kaːttoːl	kaːdnɔk	‘tub’
byːn	byːnben	byːntøːl	byːnnek	‘crime’

[5] The analysis in this section and the next builds on work with Alëna Aksënova and Maxime Papillon. Hungarian forms are given in orthographic or IPA representation, or both. In underlying forms, suffix vowel harmony variants appear with their surface vowel, since the present discussion is concerned only with consonant interactions.

However, /t/ undergoes voicing assimilation to a following [b], or other voiced obstruent, and surfaces as [d], as in [kuːdban]. Similarly, /d/ undergoes voicing assimilation to a following [t], or other voiceless obstruent, and surfaces as [t], as in [kaːttoːl]. As illustrated by [byːntøːl] and [kuːtnɔk], sonorants like /n/ neither assimilate nor trigger assimilation.

First, I posit a feature deletion rule based on set subtraction, as we have seen above:

(42) Deletion for Hungarian reciprocal neutralization

$$\begin{bmatrix} -\textsc{Son} \end{bmatrix} - \{\alpha\textsc{Voi}\} \ / \ \underline{\quad}\begin{bmatrix} -\textsc{Son} \\ -\alpha\textsc{Voi} \end{bmatrix}$$

This rule deletes the voicing value on an obstruent if the following obstruent has a different value.[6][7] The effect of such a rule is to map *tb* to *Db*, for example. The next rule is a feature-filling unification rule:

(43) Insertion for Hungarian

$$\begin{bmatrix} -\textsc{Son} \end{bmatrix} \sqcup \{\alpha\textsc{Voi}\} \ / \ \underline{\quad}\begin{bmatrix} -\textsc{Son} \\ \alpha\textsc{Voi} \end{bmatrix}$$

This rule maps *Db* to *db*. The same two rules will also map *dt* to *Dt* then to *tt*.

We can combine the two rules into a single SMD:

(44) Revised reciprocal neutralization SMD:

$$\begin{tikzcd}
t \arrow[d] \arrow[dr, "\_b"] & & d \arrow[d] \arrow[dl, "\_p"'] \\
t \arrow[d] & D \arrow[dl, "\_p"'] \arrow[dr, "\_b"] & d \arrow[d] \\
t & & d
\end{tikzcd}$$

Rule (42): $\begin{bmatrix} -\textsc{Son} \end{bmatrix} - \{\alpha\textsc{Voiced}\} \ / \ \underline{\quad}\begin{bmatrix} -\textsc{Son} \\ -\alpha\textsc{Voiced} \end{bmatrix}$

Rule (43): $\begin{bmatrix} -\textsc{Son} \end{bmatrix} \sqcup \{\alpha\textsc{Voiced}\} \ / \ \underline{\quad}\begin{bmatrix} -\textsc{Son} \\ \alpha\textsc{Voiced} \end{bmatrix}$

6 Full Hungarian situation

It turns out that there is more to Hungarian voicing assimilation. The consonants written *v* and *h* behave sometimes like obstruents like *t,d* and sometimes like sonorants like *n,r,l*. More specifically, the consonant written *v* (which we'll just call *v*), *undergoes* voicing assimilation to sometimes surface as [f], but this consonant does not *trigger* voicing assimilation, although it is typically realized as voiced [v]; and the consonant

[6] It is tempting to further simplify the rule by deleting the voicing value of any pre-obstruent obstruent, irrespective of the value on the following obstruent, leading to a Duke of York gambit in cases where the two had the same value for voicing to begin with. However, we cannot state such a deletion rule, because we don't use existential α, as explained in section 2.5. Such a rule would also interfere with the account of the behavior of *v* offered below.

[7] Note that the rule does not need to 'know' whether the righthand obstruent indeed differs in its voicing value from the target to its left. Consider the input tp: the rule will 'try' to subtract $\{+\textsc{Voiced}\}$ from t because p is $-\textsc{Voiced}$. However, the subtraction will be vacuous, since the set of features that constitutes t does not contain $+\textsc{Voiced}$.

written *h* or *ch* (which we'll just call *h*), *triggers* voicing assimilation to devoice obstruents, but this consonant does not *undergo* voicing assimilation, remaining voiceless [x] in codas, even before a voiced obstruent. In onsets, it will surface as [h] where it is not possible for it to undergo voicing assimilation at all.

My goal here is not to delve into the phonetic factors and diachronic factors that may have led to the synchronic behavior of Hungarian *v* and *h*. These are legitimate questions for those interested in the phonetics of sound change, but they are irrelevant to the narrow definition of phonology as mental computation assumed here. My goal is to demonstrate that we already have the means to account for the behavior of these two segments using the kinds of rules developed above, along with two simple assumptions about the two segments in question: (1) *v* is unspecified for VOICED underlyingly, but is otherwise specified as expected for a segment that alternates on the surface between [f] and [v]; and (2) *h* is unspecified for CONSONANTAL, but is otherwise specified as expected for a segment that surfaces as a voiceless fricative—it is −VOICED, which allows it to trigger devoicing, but the lack of a value for CONSONANTAL makes it immune to assimilation. I denote these two segments /V/ and /H/, respectively. We'll work through the treatment of these two 'exceptional' segments one at a time.

6.1 Hungarian *v*

6.1.1 Feature filling from context

Data illustrating the interaction of *v* with other consonants appears in (45) using the symbol /v/ as in Siptár and Törkenczy (2000)[8]

(45) *v* is a target of assimilation, but not a trigger

a. Target: *hívsz* /vs/ ⇝ [fs] 'you call', *óvtam* /vt/ ⇝ [ft] 'I protected'; *révbe* 'to port', *bóvli* 'junk', *sav* 'acid'

b. Non-trigger: *kvarc* /kv/ ⇝̸ [gv] 'quartz', *pitvar* /tv/ ⇝̸ [dv] 'porch'; *medve* 'bear', *olvas* 'read', *kova* 'flint', *vér* 'blood'

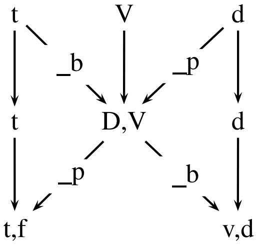
We see in (45a) that Hungarian *v* surfaces as [f] before a voiceless obstruent, as in *óvtam*, where /vt/ ⇝ [ft] 'I protected', so it apparently acts like other segments such as /z/ that are subject to voicing assimilation. However, if we assume that *v* is /V/ without an underlying specification for VOICED, then the transformation from /V/ to [f] is a feature-filling process, and thus effected by the unification based rule (43). The deletion rule (42) does not affect /V/ since there is no VOICED feature to delete. To be clear, all of the input forms in (45) should be rewritten with /V/ instead of /v/ under the assumptions I adopt here. For example, for *óvtam, sav, pitvar*, instead of /oːvtɔm, sɔv, pitvɔr/ I assume /oːVtɔm, sɔV, pitVɔr/.

The SMD in (46) shows the derivation of /V/ to both [v] and [f] by the same two rules we used in the previous section. Again, the deletion rule (42) applies vacuously,

[8] The squiggly arrow '⇝' should be read as 'leads to', meaning that the form on the lefthand side, which may be either a UR or a subsequent form in the derivation, comes out via the application of one or more rules as the form on the righthand side, which is a later stage of the derivation, and is potentially the SR. The negated form '⇝̸' implies that no subsequence (including the full sequence) of the rules maps the representation on the lefthand side to the one on the righthand side.

since /V/ has no voicing value. The voicing of the following obstruent is indicated by the environments denoted '___p' vs. ' ___b'. This value is filled onto the segment corresponding to underlying /V/ by rule (43) at this point in the derivation.

(46) *v* as *target* (output agrees with following obstruent, e.g., *óvtam* [ft])



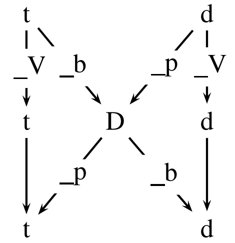
Rule (42): V has 'nothing to lose' by subtraction

Rule (43): V takes a value by unification, like (derived) D

The behavior of /V/ is formally identical to other feature-filling processes we have seen, such as the behavior of Turkish /A/ in (35).

Now consider what happens when /V/ is on the right side of a cluster, in the position of a potential trigger for voicing assimilation. A voiceless segment before a *v* does not get voiced, as we see in (45b). This is because *v*, lacking a VOICED value, does not trigger deletion of −VOICED by rule (42) in a preceding voiceless obstruent. This rule deletes the voicing value on a segment only if conflicts with the value on the next segment, and the absence of voicing on *v* cannot conflict with either +VOICED or −VOICED. The voiceless obstruents that follow *v* remain specified as voiceless. The subsequent feature filling rule (43) applies vacuously, both because the preceding obstruent has a value of its own, and because there is no value to copy from *v*.

(47) SMD for *v* as a non-*trigger* (doesn't trigger voicing agreement with preceding obstruent, as in *kvarc*)



Rule (42): No subtraction before V, since there's no mismatch

Rule (43): No feature-filling before V—/t,d/ keep input values

The two rules account for the outcome of obstruent clusters which do not contain *v*; for clusters with *v* as an affected target in the left side of an obstruent cluster; and for the (unaffected) outcome of other obstruents in clusters with a *v* on the right side.

6.1.2 A default fill-in rule

The system of two rules developed thus far fails to account for a *v* that is not to the left of another obstruent. This includes *v* before a vowel or other sonorant, or at the end of a word. Such a *v* surfaces with a +VOICED specification, that is, as [v]. A first attempt at a fill-in rule for segments derived from underlying /V/ that have not received a +VOICED specification is given in (48):

(48) $\left[\ -\textsc{Son}\ \right] \sqcup \{+\textsc{Voiced}\}$ (Only remaining V undergoes non-vacuous unification)

However, this rule can be made even more economical if we assume that at the end of the derivation every segment, consonant or vowel, other than the outputs of underlying prevocalic or word-final /V/'s has a specification for VOICED. If this is so, then the unification rule in (48) can be generalized by removing the stipulation that it apply to obstruents:

(49) $\left[\qquad\right] \sqcup \{+\textsc{Voiced}\}$ (Unify every segment with $\{+\textsc{Voiced}\}$)

See the discussion around the rule in (22) to confirm that the effect of rule (49) is to unify every segment that is a superset of the empty set (that is, every segment) with the set {+VOICED}. The possibility of unification failure enables such economical rule formulation.

6.1.3 It is what it is

Why is *v* actually /V/, a segment underspecified for voicing? Or rather, why do we posit that it is the lack of the feature VOICED that makes *v* act 'strange'? On one hand, that is *not* a phonological question. In particular, it is not a phonological question about /v/, which is the symbol for a +VOICED labiodental fricative. Under my analysis, Hungarian has no /v/. It has surfacing [v] from underlying /V/. The analysis here parallels my analysis of Russian *v* also as /V/ (Reiss, 2017a), but as mentioned above, the *phonetics* of why labiovelar fricatives seem to engage in 'exceptional' behavior cross-linguistically with respect to voicing interactions is not in the purview of a paper on phonological computation. The model of segments as sets of features adopted here allows for underspecification (and this is actually a simplification of phonology, as I have explained elsewhere (Reiss, 2012; Matamoros and Reiss, 2016; Bale and Reiss, 2018)), so there is no 'cost' for modeling Hungarian *v* as /V/. Since it is exactly the feature VOICED with respect to which *v* acts exceptionally, it is not surprising that it is the absence of this feature which, I suggest, distinguishes *v* from other obstruents.

Other approaches to the behavior of *v*, in Russian and Hungarian both, have suggested that the segment in question is not specified as an obstruent. For example Siptár and Törkenczy (2000), in a thorough discussion of the data and an analysis combining aspects of feature geometry and Government Phonology, treat *v* as underspecified for the feature SONORANT. Given that two realizations of *v* are clearly the fricatives [v] and [f], it seems reasonable to attempt to treat the underlying segment from which these obstruents are derived as specified −SONORANT, i.e., as an obstruent.[9] So, from this perspective (adopting certain idealizations), "Why /V/?" is a *purely* phonological question, and the answer is that positing this segment for *v* gives us an elegant account of Hungarian that can be constructed without extending the model of segmental changes we developed for the many other patterns of phonological processes illustrated above.

It may be useful to show graphically that in developing an account of Hungarian our model has not gotten any more complicated than what we used to account for the various languages above. In order to present an SMD for Hungarian, we will adjust our interpretation of the symbols ϕ, ψ, Δ. Instead of assuming that the three are identical aside from a single feature specification, let's abstract away from all other features, but one. In other words, let ϕ be any voiceless obstruent, [p, t, s], etc.; let ψ be any voiced obstruent, [b, d, z] etc.; and let Δ be any obstruent unspecified for voicing, such as /V/. The following SMD shows how the rule system we developed handles these three types of segment, wherever they occur, as first or second member of an obstruent cluster or elsewhere.

(50) SMD for 'normal' obstruents and /V/ in Hungarian

$$\begin{tikzcd}
\text{UR} & \phi \arrow[d] \arrow[dr] & \Delta \arrow[d] & \psi \arrow[d] \arrow[dl] & \text{Rule (42)} \\
& \phi \arrow[d] & \Delta \arrow[d] \arrow[dl] \arrow[dr] & \psi \arrow[d] & \text{Rule (43)} \\
& \phi \arrow[d] & \Delta \arrow[dr] & \psi \arrow[d] & \text{Rule (49): Default +VOICED} \\
\text{SR} & \phi & & \psi &
\end{tikzcd}$$

The first α rule deletes the first in a sequence of two incompatible voicing features. The next α rule inserts into the first consonant the value of voicing on the second consonant, if the first lacks a value for that feature. The third rule is the default +VOICED fill-in rule.

6.2 Hungarian *h*

Two simple stipulations are needed to extend the analysis of Hungarian to the underlying consonant surfacing as a voiceless glottal fricative [h] in onsets, written *h*, and

[9] For some Hungarian speakers, *v* can be more sonorant-like, but different dialects require different analyses and there is apparent free variation. At risk of oversimplifying, I focus on an idealized speaker who has just [v] and [f] on the surface, but detailed studies like Kiss and Bárkányi (2006) should be consulted. The messy facts may be related to the 'two Hungarian *v*'s' mentioned in section (6.3).

a voiceless velar fricative [x], also written *h*, or sometimes *ch*, in codas.[10] Recall that we will denote this segment /H/ and assume that it lacks a specification for CONSONANTAL.[11] There must be something about the featural make-up of this segment that allows it to cause devoicing of preceding obstruents, as in *adhat* /ɔd-Hɔt/ 'he may give' ⤳ [ɔthɔt], without allowing the *h* itself to undergo assimilation as a target, as seen in *pechből* /peH-bø:l/ 'out of bad luck' where /Hb/ ⤳ [xb]. We achieve this effect by (a) slightly revising the feature deletion rule (42) to make reference to the valued feature +CONSONANTAL; and (b) by accepting that *h* is underlying /H/, a segment that has all the features of [h] aside from CONSONANTAL. Here's the revised verison of (42):

(51) Revised deletion for Hungarian reciprocal neutralization

$$\begin{bmatrix} -\textsc{Son} \\ +\textsc{Cons} \end{bmatrix} - \{\alpha\textsc{Voi}\} \;/\; \_\_ \begin{bmatrix} -\textsc{Son} \\ -\alpha\textsc{Voi} \end{bmatrix}$$

Comments

a. 'Normal' obstruents lose VOICED before obstruent with opposite value

b. /V/ doesn't have a feature to lose

c. /H/ is not affected because it is not +CONSONANTAL

Rule (51) will not delete voicing on /V/ because there is none. It won't delete voicing on /H/ because the rule specifies that the target must be +CONSONANTAL and /H/, by hypothesis is not thus specified—it has no value for CONSONANTAL.

Since the segment /H/ will not lose its −VOICED specification by rule (51), the unification rule (43), repeated as (52), responsible for inserting a VOICED value will apply vacuously when the reflex of /H/ is the first member of the cluster—either because the following segment is also voiceless, so unification is vacuous, or because the following segment is voiced, so unification fails.

(52) (Repeated) Insertion for Hungarian *h* as target: unification fails

$$\begin{bmatrix} -\textsc{Son} \end{bmatrix} \sqcup \{\alpha\textsc{Voi}\} \;/\; \_\_ \begin{bmatrix} -\textsc{Son} \\ \alpha\textsc{Voi} \end{bmatrix}$$

The slightly revised deletion rule is not quite the only change needed to account for clusters containing *v* and *h*. Underlying /H/ also needs some fill-in rules, assuming that its reflexes are fully specified at the surface. If we assume that /H/ is the only source of segments without a value for CONSONANTAL near the end of the derivation, then we can assign +CONSONANTAL in coda position where the surface form is [x] with rule (53):

[10] See Siptár and Törkenczy (2000, Section 8.2.2) for a discussion of words where onset *h* alternates with 'zero' in codas. I follow these authors in treating such cases as non-phonological suppletive allomorphy, although there are other possibilities. In any case, the issue is orthogonal to our concerns here.

[11] This analysis follows Siptár and Törkenczy (2000, 277) in treating *h* as underlyingly underspecified for CONSONANTAL. They posit that *h* is underlyingly /X/, which is an /x/ unspecified for CONSONANTAL. I am not primarily concerned here with the *x* ~ *h* alternation and variation, and so I use /H/ rather than /X/ for the expository convenience of relating an orthographic form to a posited underlying underspecified form: *h* : /H/ as *v* : /V/ (with different features unspecified, of course).

(53) $\left[\qquad\right] \sqcup \{+\text{CONS}\}$/ in CODA (Only ‘H’ undergoes non-vacuous unification)

In onset position, we can follow the *SPE* (Chomsky and Halle, 1968) tradition of treating [h] as −CONSONANTAL:

(54) $\left[\qquad\right] \sqcup \{-\text{CONS}\}$/ in ONSET (Only ‘H’ undergoes non-vacuous unification)

Let’s (arbitrarily) assume that feature filling for CONSONANTAL in onsets is ordered after filling for codas. We can then remove the rule context and formulate a more general rule that extensionally only applies to the reflex of /H/ in onsets:

(55) $\left[\qquad\right] \sqcup \{-\text{CONS}\}$(Only ‘H’ undergoes non-vacuous unification)

Alternatively, we could leave *h* in onset unspecified for CONSONANTAL at the surface, as in the cases of surface underspecification presented by Keating (1988).[12]

6.3 Hungarian cluster derivations

Recall that the focus of this paper is on changes within segments. In order to illustrate the logic of our set theoretic approach we have made several simplifications. For example, we have only considered clusters of two consonants, but Hungarian has clusters of three consonants, with the rightmost member determining the outcome of the whole cluster, so that it looks like voicing assimilation works iteratively from right to left, e.g., *liszt-ből* [stb] $\rightsquigarrow$ [sdb] $\rightsquigarrow$ [zdb] ‘from flour’. There are alternative analyses, but this brings us into questions of how rules apply and how rule environments are stated, and I leave these questions aside for now. Here, I give derivations for schematic cases of sequences of one or two consonants, where each member of the sequence is either one of the ‘regular’ obstruents or one of the two ‘exceptional’ obstruents corresponding to *v* and *h*. In other words, all of the segments in these sequences are specified −SONORANT.

Let’s first consider the clusters of fully specified segments, corresponding to the discussion of simplified Hungarian in section 5. The derivations are in (57), using the same abbreviations that I will now explain and use in the subsequent tables.

In the tables below (57-60), a dash ‘–’ indicates, as is traditional, an identity mapping from the next cell up in the table, when the structural description of a rule is not met. For example, underlying /tp/ is unchanged by the first rule (51), since the segments do not disagree in voicing. The rule conditions are not met and this is denoted with the dash ‘–’.

In the same column, the row corresponding to the rule (52) that potentially inserts voicing value is marked with ‘vac.’, indicating the vacuous *unification* of /t/ with the set {−VOICED} coming from /p/.

The next cell down is marked ‘un.fail’ for unification failure. This is the row corresponding to the default rule (49) that fills in +VOICED. Unification of /t/ (and /p/) with {+VOICED} fails because of the conflict of −VOICED with +VOICED. Since unification fails, we again map the input to the output. Thus we have three different situations that result in vacuous rule application, but I have marked them differently:

[12] Note again that I am not going to formulate rules for the velar vs. glottal distinction between [x] and [h].

(56) Three kinds of vacuous rule application:

a. – : Input string does not match rule structural description, so target segment is unaffected

b. vac. : Unification is vacuous, so rule application does not affect target segment

c. un.fail : Unification fails because of feature value conflict, so (by definition) input representation is mapped to identical output representation

I call all three of these situations “vacuous rule application" rather than, for example, saying that the rule does not apply in situation (56a) for reasons discussed in Bale and Reiss (2018). In brief, if we want the phonology to be a function composed from the individual rules, each rule must ‘apply’ to each sequence, so that there is an output that can be handed to the subsequent rules, or ultimately to the surface representation.

Here are the derivations for clusters of fully specified segments:

(57) Derivations for Hungarian fully specified sequences

UR	tp	tb	dp	db
(51)Delete $-\alpha$Voi	–	Db	Dp	–
(52) Insert αVoi	*vac.*	db	tp	*vac.*
(49) Fill +Voi	*un.fail*	*vac.*	*un.fail*	*vac.*
(53) Fill +Cons Coda	*vac.*	*vac.*	*vac.*	*vac.*
(55) Fill −Cons	*vac.*	*vac.*	*vac.*	*vac.*
SR	tp	db	tp	db

Since the examples in (57) involve only fully specified segments, the last three rules, which are all fill-in rules, can have no effect on the inputs. The situation changes as we consider inputs with underspecified segments. Let’s first consider derivations whose inputs contain /V/ as either the first segment of an obstruent cluster (/Vp/ and /Vb); the second segment of an obstruent cluster (/tV/ and /dV/); and elsewhere (/V/).

(58) Derivations for Hungarian sequences with /V/

UR	Vp	Vb	tV	dV	V
(51)Delete $-\alpha$Voi	–	–	–	–	–
(52) Insert αVoi	fp	vb	–	–	–
(49) Fill +Voi	*un.fail*	*vac.*	tv	dv	v
(53) Fill +Cons Coda	*vac./-*	*vac./-*	*vac./-*	*vac./-*	*vac./-*
(55) Fill −Cons	*un.fail*	*un.fail*	*un.fail*	*un.fail*	*un.fail*
SR	fp	vb	tv	dv	v

None of the clusters or the single consonant /V/ meet the conditions for the first rule (51), which looks for a mismatch in voicing values on adjacent obstruents, so we have ‘–’ across the row. However, the feature filling rule (52) in the second row provides underlying /V/ with a voicing value that agrees with the following obstruent, so we get [f] before [p] and [v] before [b].

The next row corresponds to rule (49) which fills in +VOI . In the first column of inputs, /V/ has turned into [f] so unification fails; in the second column, /V/ has turned into [v], so unification is vacuous. In the last three forms, /V/ is unchanged from the UR, so the fill-in rule provides +VOICED and we get [v] in all three cases.

Consider the next row, corresponding to rule (53), which fills in +CONSONANTAL in codas. All of the consonants in the five columns are specified +CONSONANTAL (since the UR). So the unification will be vacuous when one of these falls in a coda. When a consonant is in an onset, the rule condition is not met. This is why all five columns are marked with both 'vac.' and '—'.

Finally, in the last row, rule (55) 'tries' to fill in −CONS on reflexes of the input consonants, but they have remained +CONS since the underlying representation. Unification fails in each case, as indicated, and the surface representations all come out as desired.

Now we turn to the derivation of forms with /H/ in the input. We consider /H/ as the first or second member of a cluster between vowels, so in coda or onset position, with the other segment either voiceless or voiced. We also need to consider /H/ on its own, not adjacent to another obstruent, in both onset and coda position.

(59) Derivations for Hungarian sequences with /H/

UR	Hp	Hb	tH	dH	H-ons	H-cod
(51)Delete $-\alpha$VOI	–	–	–	DH	–	–
(52) Insert αVOI	*vac.*	*un.fail*	*vac.*	tH	–	–
(49) Fill +VOI	*un.fail*	*un.fail*	*un.fail*	*un.fail*	*un.fail*	*un.fail*
(53) Fill +CONS CODA	xp	xb	*vac./–*	*vac./–*	–	x
(55) Fill −CONS	*un.fail*	*un.fail*	th	th	h	*un.fail*
SR	xp	xb	th	th	h	x

Because rule (51), the first rule, requires that the lefthand member of an input sequence be specified +CONSONANTAL, this condition is not met when /H/ precedes another obstruent, so we get '–' in the first two columns. We get '–' in the third column because /t/ and /H/ do not disagree in voicing, since both are −VOICED. In the next column, the voicing mismatch of /d/ with following /H/ causes the deletion of +VOICED on the former, yielding derived underspecified [D]. The last two columns do not contain sequences of obstruents, so the rule condition is not met.

The only non-vacuous application in the next row, for the rule (52) that copies a voicing value from the right-hand member of a sequence to the left, is the case where derived [D] becomes [t] before the voiceless /H/.

At this point in the derivation, all segments in this table are specified as either +VOICED or −VOICED, so the default rule that fills in +VOICED will alway be vacuous. I have marked this as 'un.fail' in each column, since there is unification failure with at least one segment in each case, but for the second column of forms, there is not only unification failure with the reflex of /H/ but also vacuous unification with the reflex of /b/, which is still voiced.

In the next row, rule (53) fills in +CONSONANTAL on each /H/ in coda position, yielding [x] (since we are ignoring the glottal vs. velar contrast). For the forms in the third and fourth columns, derived from /tH/ and /dH/, respectively, the reflex of /H/ is not in a coda, so we mark '–', but the rule as formulated also targets the underlying and derived [t]'s. These are already +CONSONANTAL, so the unification is also vacuous, so marked 'vac.'

The next row shows that any remaining /H/ becomes −CONSONANTAL at this point in the derivation, but for all other consonants, which are +CONSONANTAL, there is unification failure.

Finally, let's show derivations for clusters /HV/ and /VH/:

(60) Derivations for Hungarian sequences /HV/ and /VH/

UR	HV	VH
(51)Delete $-\alpha$VOI	–	–
(52) Insert αVOI	–	fH
(49) Fill +VOI	Hv	*un.fail*
(53) Fill +CONS CODA	xv	*vac.*
(55) Fill −CONS	*un.fail*	fh
SR	xv	fh

For the second column, forms like *szívhez* 'to heart' with /VH/ ⤳ [fh] confirm that our rules work for this cluster. There is a complication with finding the output for /HV/ since Hungarian has two kinds of suffixes that begin with *v*. In some Hungarian *v*-initial suffixes, the *v* assimilates fully to a preceding consonant. In the others, the *v* does not assimilate. This non-assimilating *v* is the one we are interested in. This distinction could be due to a featural difference in the two *v*'s or, as Siptár and Törkenczy (2000) assume, to a structural difference in the words containing the two kinds of suffixes. Unfortunately it is accidentally the case that the suffixes that show non-assimilating *v* do not attach to any stems ending in /H/. For example, the *va/ve* suffix attaches to verb roots, but there are no verbs ending in /H/. Thus the predicted output [xv] is never observed. Péter Siptár (p.c.), a native speaker phonologist, suggests that the expected outcome of the relevant sequence in a *wug*-test situation would indeed be [xv].[13] Fortunately, Hungarian voicing assimilation is postlexical (see fn. 22, p. 198 of Siptár and Törkenczy 2000) so it applies not only between a root and a suffix, but also across compound boundaries and word boundaries, so we can find relevant cases. Just as the inflected form *szívhez* parallels the compound *tév-hit* 'misbelief' where both show /vH/ ⤳ [fh], the accidental data gap identified above is predicted to parallel compounds like *potroh-végi* 'caudal' with /HV/ surfacing as [xv]. In fact, there actually is no data gap, since the process is postlexical. So, the system I have constructed appears to work for this cluster as well.

[13] See Siptár and Törkenczy (2000), especially sections 8.2.1-2 for useful discussion of the two kinds of *v*-initial suffixes and the status of stems ending in *h*, of which there are also two kinds. The accidental absence of the relevant combination of verb root shape and suffix behavior was confirmed by Péter Siptár (p.c.).

7 Conclusions

The point of the epigraph to this paper “We learn not to worry about purpose, because such worries never lead to the sort of delight we seek” is that this work is the antithesis of that exhorted by Prince and Smolensky (1993, 216) in a founding document of Optimality Theory: “We urge a reassessment of this essentially formalist position. If phonology is separated from the principles of well-formedness (the ‘laws’) that drive it, the resulting loss of constraint and theoretical depth will mark a major defeat for the enterprise.” The framework presented here is purely formal and has no teleology or notion of well-formedness. Phonology is not trying to optimize anything at all, to repair or to block marked structures, to find "cures" for "conditions" (Yip, 1988), or manage "trade-offs" between ease of articulation and maintenance of meaning contrasts contra, for example, the papers in Hayes et al. (2004).

Instead, the Logical Phonology approach I assume here (developing work such as Bale et al. (2014); Bale and Reiss (2018); Bale et al. (2020); Volenec and Reiss (2020)) posits a set of representational primitives, such as binary features and α variables, as well as a set of computational primitives that are organized into rules that do things like subtract or unify sets of valued features. Phonology, in this conception, involves abstracting away from instances of phonological patterns the contingencies of long chains of historical events in the course of transmission across generations (as discussed by Blevins 2003; Hale 2007; Hale and Reiss 2008, among others). Rather than seeing this as an original idea, it strikes me as the normal method of science, “the natural approach: to abstract from the welter of descriptive complexity certain general principles governing computation that would allow the rules of a particular language to be given in very simple forms, with restricted variety” (Chomsky, 2000, 122). This is exactly what I have tried to do in this paper by deriving superficial variety from a few basic notions.

Phoneticians, acquisitionists and historical linguists get their thrills, their delight, in various ways that bear on the particularities of the “welter of descriptive complexity” we see in phonological systems—*Why do lots of languages have rules that do this, but very few or maybe none have rules that do that? Why do these sounds seem to rarely appear next to each other?* These are all interesting and legitimate fields of study, but different from the study of what those systems are that are being realized in sound and transmitted across generations with more or less success. It thus seems reasonable to distinguish those tasks from that of understanding how the individual systems manifest the “general principles governing computation” and the nature of the representations that those computations take as arguments in phonological systems. These narrow questions offer their own “sort of delight”, and, to my mind, are the subject matter of phonology, defined in a usefully narrow sense.

Recall that combinatorics is our friend: the model proposed is in the spirit of Gallistel and King (2009, 79) who stress the benefit of combinatoric explosion from a small set of primitives in modeling cognitive faculties:

> A machine with a very rich store of symbols must have a means of forming them out of a not too numerous store of atomic data. No language in the world has a word for the message, “After the circus, I’m going to

> the store to get a quart of skim milk." No system for representing discrete numbers represents 1,342,791 with a single digit or number name. Minimizing the number of atomic data is desirable in any symbol system as it reduces the complexity of the machinery required to distinguish these data. This is why computers use just two atomic data and likely why nucleotide sequences use only four.

The Logical Phonology model derives the commonalities among languages as well as the welter of descriptive complexity, the superficial variety we observe, via combinatoric explosion over a small inventory of elements and operations. The model of innate mechanisms is small, with not much "attributed to genetic information." In other words, we are aiming to minimize our ontological commitments for UG, and intensionally define a set of possible human languages rich enough to contain the superficial variety we observe. I offer this simple model of segment-internal changes as a form of effective data compression in the spirit of Chaitin (2006), who says that a "useful theory is a compression of the data; from a small number of laws and equations, whole universes of data can be computed". In more down-to-earth terms we can cite Morris Halle: "Too 'simple'? You want it to be complicated?"

References

Bale, Alan, Maxime Papillon, and Charles Reiss. 2014. Targeting underspecified segments: A formal analysis of feature changing and feature filling rules. *Lingua* 148:240–253.

Bale, Alan, and Charles Reiss. 2018. *Phonology: A formal introduction*. Cambridge, MA: MIT Press.

Bale, Alan, and Charles Reiss. 2021. Just a bunch of degenerates: Natural classes in phonology.

Bale, Alan, Charles Reiss, and David Ta-Chun Shen. 2020. Sets, rules and natural classes: { } *vs.* [ ]. *Loquens* 6.

Blevins, Juliette. 2003. *Evolutionary phonology: The emergence of sound patterns*. Cambridge, MA: Cambridge University Press.

Chaitin, Gregory. 2006. The limits of reason. *Scientific American* 294:74–81.

Choi, John D. 1992. Phonetic underspecification and target interpolation: An acoustic study of Marshallese vowel allophony. Doctoral Dissertation, UCLA, Los Angeles.

Chomsky, Noam. 1982. *Some concepts and consequences of the theory of government and binding*. Cambridge, MA: MIT Press.

Chomsky, Noam. 2000. Language as a natural object. In *New horizons in the study of language and mind*, 106–133. Cambridge: Cambridge University Press.

Chomsky, Noam. 2007. Approaching UG from below. In *Interfaces + recursion = language?: Chomsky's minimalism and the view from syntax-semantics*, ed. Uli Sauerland and Hans-Martin Gärtner, 1–24. Berlin: Mouton de Gruyter.

Chomsky, Noam, and Morris Halle. 1968. *The sound pattern of English*. New York: Harper & Row.

Doke, Clement Martyn. 1938. *Text book of Lamba grammar*. Witwatersrand University Press.

Gallistel, C. R., and Adam Philip King. 2009. *Memory and the computational brain: why cognitive science will transform neuroscience*. Chichester, West Sussex, UK: Wiley-Blackwell.

Hale, Mark. 2007. *Historical linguistics: theory and method*. Malden, MA: Blackwell.

Hale, Mark, and Charles Reiss. 2000. "Substance abuse" and "dysfunctionalism": Current trends in phonology. *Linguistic Inquiry* 31:157–169.

Hale, Mark, and Charles Reiss. 2008. *The phonological enterprise*. Oxford: Oxford University Press.

Hayes, Bruce, Robert Martin Kirchner, and Donca Steriade. 2004. *Phonetically based phonology*. Cambridge: Cambridge University Press.

Inkelas, Sharon, and Orhan Orgun. 1995. Level ordering and economy in the lexical phonology of Turkish. *Language* 71:763–793.

Keating, Patricia. 1988. Underspecification in phonetics. *Phonology* 5:275–292.

Kiparsky, Paul. 1985. Some consequences of Lexical Phonology. *Phonology Yearbook* 2:85–138.

Kiss, Zoltán, and Zsuzsanna Bárkányi. 2006. A phonetically-based approach to the phonology of [v] in Hungarian. *Acta Linguistica Hungarica* 53:175–226.

Matamoros, Camila, and Charles Reiss. 2016. Symbol taxonomy in biophonology. In *Biolinguistic investigations on the language faculty*, ed. Anna Maria Di Sciullo, 41–54. John Benjamins.

McCarthy, John. 1994. The phonetics and phonology of Semitic pharyngeals. In *Phonological structure and phonetic form: Papers in laboratory phonology iii*, ed. P. Keating. Cambridge: Cambridge University Press.

McCawley, James D. 1971. *On the role of notation in generative phonology*. Indiana University Linguistics Club.

Odden, David. 2013. *Introducing phonology*. Cambridge: Cambridge University Press, 2 edition.

Papillon, Maxime. 2020. Precedence and the lack thereof: Precedence-relation-oriented phonology. Doctoral Dissertation, University of Maryland.

Prince, Alan, and Paul Smolensky. 1993. *Optimality Theory: Constraint interaction in generative grammar (technical report)*. New Brunswick, NJ: Rutgers Center for Cognitive Science.

Raimy, Eric. 2000. *The phonology and morphology of reduplication*. Berlin: M. de Gruyter.

Reiss, Charles. 2012. Towards a bottom-up approach to phonological typology. *Towards a Biolinguistic Understanding of Grammar* 169–191.

Reiss, Charles. 2017a. Contrast is irrelevant in phonology: A simple account of Russian /v/ as /V/. In *Arguments of substance: Beyond markedness*, ed. Bridget D. Samuels, 23–45. Amsterdam: John Benjamins.

Reiss, Charles. 2017b. Substance Free Phonology. In *Handbook of Phonological Theory*, ed. S.J. Hannahs and Anna Bosch, 425–452. New York: Routledge.

Reiss, Charles. 2021. Priority union and feature logic in phonology. *Linguistic Inquiry* .

Shen, David Ta-Chun. 2016. *Precedence and Search: Primitive concepts in morphophonology*. Taipei: National Taiwan Normal University dissertation.

Siptár, Péter, and Miklós Törkenczy. 2000. *The phonology of Hungarian*. Oxford: OUP.

Volenec, Veno, and Charles Reiss. 2020. Formal generative phonology. *Radical: A Journal of Phonology* 2:1–148.

Yip, Moira. 1988. The Obligatory Contour Principle and phonological rules: A loss of identity. *Linguistic Inquiry* 19:65–100.