Sets, Rules and Natural Classes: {} *vs.* [ ]

Alan Bale, Charles Reiss & David Ta-Chun Shen

tract**

We discuss a set-theoretic treatment of segments as sets of valued features and of natural classes as intensionally defined sets of sets of valued features. In this system, the empty set { } corresponds to a completely underspecified segment, and the natural class [ ] corresponds to the set of all segments, making a feature ±SEGMENT unnecessary. We use unification, a partial operation on sets, to implement feature-filling processes, and we combine unification with set subtraction to implement feature-changing processes. We show how unification creates the illusion of targeting only underspecified segments, and we explore the possibility that only unification rules whose structural changes involve a single feature are UG-compatible. We show that no such Singleton Set Restriction can work with rules based on set subtraction. The system is illustrated using toy vowel harmony systems and a treatment of compensatory lengthening as total assimilation.

Resumen

1 Introduction

Developing ideas introduced in Bale, Papillon, and Reiss (2014) and Bale and Reiss (2018), this paper explores some of the consequences of analyzing segments as sets of features and, as a corollary, analyzing natural classes as sets of sets of features. We adopt the (fairly standard) view that features rather than segments are the primitives of phonological representation: the "alphabetic symbols that we use freely [are] nothing more than convenient ad hoc abbreviations for feature bundles, introduced for ease of printing and reading but without systematic import" (Chomsky & Halle, 1968, 64).[1] Taking these feature bundles to be sets allows us to apply ideas from set theory to phonology.[2] In particular, we discuss the logical possibility that some segments may be fully underspecified and thus are represented by the empty-set, denoted by either ∅

---

[1] Some work in phonology (e.g., Flemming, 2005; Mielke, 2008) appears to accept both segments and features as primitives or to posit segments as primitives with features being 'assigned' to segments on a language specific, ad hoc basis. This is not the place to explain why we find these positions vague and untenable, but the reader should be aware of their existence.

[2] See Reiss (2003) for some arguments against adopting the kind of feature geometric approaches to representations advocated by Sagey (1986), McCarthy (1988) and many others.

1

or {}, depending on the context.[3] We also discuss how the empty set can be used to define a natural class over all segments.

Our discussion of the empty segment starts by highlighting some general properties of phonological rules, in particular the fact that the target of rules are natural classes. As argued in section 2, this basic property of rules has significant consequences for the empty segment, namely the only plausible natural class that contains the empty segment also contains all other segments. Hence, it is impossible to target the empty segment without also targeting every other segment. As we outline in section 4, there are indeed phonological processes that plausibly involve rules that target the empty segment, namely complete assimilation. We argue that complete assimilation (as well as partial assimilation) is best treated by decomposing the traditional arrow of phonological rules into two separate operations, one that involves feature deletion and another that uses unification (a partial function similar to set union). Feature deletion creates partially underspecified or fully underspecified segments, while unification "fills-in" these underspecified segments. This "two step" procedure allows us to technically target all segments for assimilation but yet only affect the underspecified segments. The process is discussed in the abstract in section 4, and then as a plausible analysis of vowel harmony and compensatory lengthening in sections 5 and 6. As part of our discussion, we explore the hypothesis that assimilation is done *one feature at a time* in the sense that there is a separate rule of assimilation for each feature. We demonstrate how such a hypothesis not only simplifies our account of certain phonological phenomenon but also makes novel empirical predictions.

2   Features, Segments and Natural Classes

Bale et al. (2014) introduce an innovation to phonological notation to better reflect the type-theoretic nature of the components of rules, based on the idea that segments are *consistent* sets of valued features, that is, sets that may contain $+F$ or $-F$, for a given feature F, but not both (see section 4).[4] The basic idea behind this innovation is best demonstrated by providing an example of a prototypical phonological transformation.[5]

Suppose we have a language with a five-vowel inventory that raises its mid vowels to their high counterparts before a nasal consonant. Informally, we might indicate such a process as in (1):

(1)   **A Simple Phonological Process**:

---

[3]One reason to favor { } over $\emptyset$ is that in Bale and Reiss (2018) we use a variant of the empty set symbol, $\varnothing$, to denote morphemes with no phonological content. The empty braces { } are less likely to be confused with that symbol than the normal empty set symbol, $\emptyset$.

[4]For purposes of this paper, we abstract away from several crucial points, such as how these sets are associated with timing slots, and the related issue of contour segments, such as prenasalized stops and affricates, which, in violation of the consistency requirement, do appear to contain conflicting tokens or feature values within a single segment (*e.g.*, $+$NASAL and $-$NASAL). We think the logical issues raised in this paper and by Bale et al. (2014) would survive extension to structures more complex than sets, a topic of our current research.

[5]Our use of the terms 'transformation', 'process' and 'rule' will be a bit awkward, since we treat so-called feature changing rules as processes consisting of two rules.

o,e	$\rightarrow$	u,i	/ __	n, m, ŋ
Target		Change		Environment

The targeted vowels *o,e* each turn into the corresponding high vowel before any one of the nasals *n,m,ŋ* that define the environment. Traditional formalizations of such a process might represent the target set, the set of mid vowels in the language, as [−HIGH, −LOW], as in (2):[6]

(2) **Traditional Formulation of Process**:
   [−HIGH, −LOW] →[+HIGH] / __ [+NASAL]

Unpacking this traditional notation we see that the rule refers to the set of targets *via* a superset relation: Chomsky and Halle's "feature bundle" segments are sets of valued features, and a segment is a target of the rule if and only if it is a superset of the set {−HIGH, −LOW}.[7] The target of the process is thus a set of segments; that is, a set of sets of valued features, as in (3):

(3) Target—natural class of *segments*
   $X = \{x : x \supseteq \{-\text{HIGH}, -\text{LOW}\}\}$

The characterization of the environment of our rule must also be a set of segments, namely the set of segments that are a superset of the set of features {+NASAL}, as in (4):

(4) Environment—natural class of *segments*
   $Y = \{y : y \supseteq \{+\text{NASAL}\}\}$

The square brackets in the target and the environment in traditional rules like (2) are used to symbolize these natural classes:

(5)   a. $\{x : x \supseteq \{-\text{HIGH}, -\text{LOW}\}\}$ = [−HIGH, −LOW]

       b. $\{y : y \supseteq \{+\text{NASAL}\}\}$ = [+NASAL]

Although the square brackets are used to specify natural classes in the target and environment of traditional rules, they are used differently to specify the structural change ([+HIGH] in (2) above). This change does not refer to a set of segments—the interpretation of the change is not parallel to the "natural class" interpretation of the target and environment. The structural change lists the features that are changed or added to the target segments when they appear in the rule environment.

   In traditional notation, square brackets do not have the same set-theoretical meaning for the structural change as they do for the target and environment. To address this inconsistency, we replace the square brackets with standard set braces. The set-brace notation, as in (6), makes clear the difference between natural classes—i.e., sets of sets of (valued) features—and mere sets of (valued) features.

---

[6]Alternatively, the target set might be represented as just [−LOW], if such a process is assumed to apply vacuously to vowels specified +HIGH.

[7]See discussion in Bale et al. (2014), especially fn. 6, about how this traditional interpretation differs from that of *SPE* (Chomsky & Halle, 1968).

(6)   Structural Change—set of *features*:
      {+HIGH}

With this notation, the process in (2) is expressed instead as (7).

(7)   **Process expressed with consistent bracketing**:
      $[-\text{HIGH}, -\text{LOW}] \rightarrow \{+\text{HIGH}\}$ / ___ [+NASAL]

    a. { ... } denotes a set of features

    b. [...] denotes a set of sets of features, a set of segments constituting a natural class

Our revised notation adopts Bertrand Russell's type theory, like earlier work in phonology (Peterson & Harary, 1961), and consistently distinguishes a natural class of segments from a set of features.In this paper, we explore the hypothesis that all phonological rules fit the format represented in (7). In other words, all rules intensionally target a natural class and all environmental triggers are intensionally defined in terms of natural classes.

Several points follow from our interpretation of natural classes in rules. First, the notation works well with underspecification. If a language has a voiced /d/, a voiceless /t/ and a coronal stop unspecified for voicing /D/, for example, as Inkelas (1995) proposes for Turkish, then the representation in (8) describes the natural class that contains all three of these segments.

(8)   A natural class with an underspecified member[8]

    i. Description of class:
        $[+\text{CORONAL}, -\text{CONTINUANT}, -\text{SONORANT} \ldots] =$
        $\{x : x \supseteq \{+\text{CORONAL}, -\text{CONTINUANT}, -\text{SONORANT} \ldots\}\}$

    ii. Members of the class

        a. /t/ = { −VOICE, +CORONAL, −CONTINUANT, −SONORANT ... }

        b. /d/ = { +VOICE, +CORONAL, −CONTINUANT, −SONORANT ... }

        c. /D/ = { +CORONAL, −CONTINUANT, −SONORANT ... }
        (no VOICE feature)

So, our set theoretic interpretation allows us to refer to a natural class that includes a member whose specification is a subset of that of other members—for example, the specification of /D/ is a subset of the specification of /t/. This is a non-trivial result in light of a widely cited characterization of natural classes: "We shall say that a set of speech sounds forms a *natural class* if fewer features are required to designate the class than to designate any individual sound in the class" (p.328 Halle, 1964). In our new characterization, the same number of features may be needed to characterize a class C and some member segment $q \in C$. However, by paying attention to types, we can distinguish reference to C from reference to q. For example, compare the

---

[8]The ellipsis '...' denotes whatever other features may be relevant to characterizing the segments in question, in this case, anything except VOICE.

representation in (8.i) with that in (8.ii.c). The former is a set of sets of valued features; the latter is a set of valued features.

Notice that a natural class can contain just a single member, but the representation of such a class is not identical to the representation of its unique member:

(9)   The segment /t/ vs. the class containing only the segment /t/

- The segment /t/
  $\{ -\text{VOICE}, +\text{CORONAL}, -\text{CONTINUANT}, -\text{SONORANT} \dots \}$

- The class containing only the segment /t/
  $[-\text{VOICE}, +\text{CORONAL}, -\text{CONTINUANT}, -\text{SONORANT} \dots] =$
  $\{x : x \supseteq \{-\text{VOICE}, +\text{CORONAL}, -\text{CONTINUANT}, -\text{SONORANT} \dots\}\}$

We refer to the segment /t/ with features in set braces, but the set containing just /t/ has the same features listed in square brackets.

There are, however, some segments that cannot constitute a natural class by themselves. Our new notation makes it clear why it is impossible to characterize a class containing only an underspecified segment, without containing more specified ones that occur in the same language. This follows from the logic of the subset/superset relation: the features of /D/ are a subset of the features of, say, /t/, so it is impossible to represent a natural class that contains /D/ but not /t/. To reiterate, although we can refer to a natural class containing just /t/, and not /d/ or /D/, we cannot refer to a class containing just /D/, and not /t/ and /d/ in the target or environment of a rule.

This logical result immediately appears to be at odds with the existence of feature-filling rules—for example, the analysis of Turkish (Inkelas, 1995) requires that /D/ surface as voiced in onsets and voiceless in codas, but /t/ does not get voiced in onsets. How can we manage to target /D/ in an onset voicing rule, but not /t/, if we can't refer to the natural class containing just /D/? We will return to this issue below.

Extrapolating from the impossibility of referring to /D/ (without /t/ and /d/), which is underspecified only for VOICED, it follows that no rule can target a segment p whose representation contains a subset of the information in the representation of a segment q, without also targeting q. Pushing this to the logical limit, what happens if a segment corresponds to the empty set of valued features, the *fully* underspecified segment, which we denote as $\{\ \}$? The answer is that it is impossible to target this segment without targeting every other segment.

Here's why: a basic result of set theory is that the empty set is a subset of every set.[9] This means that every set is a *superset* of the empty set. If we write a rule with the target class specified without any features, we get the expression in (10). Such a rule must target all segments.

(10)   Empty Target—natural class of segments $X = \{x : x \supseteq \{\ \}\} = [\quad]$

For concreteness, consider a rule with such a target, as in (11):

(11)   **Process with an empty target**:
        $[\quad] \rightarrow \{+\text{HIGH}\} / \underline{\quad} [\ +\text{NASAL}\ ]$

---

[9]Given any set A, there is no member of $\emptyset$ that is not a member of A—since $\emptyset$ has no members.

5

The notation [ ] in (11) does indeed allow the rule to target the fully underspecified segment, but it cannot refer to such a segment to the exclusion of all other segments. Despite the fact that there can be no rule that only targets a segment with an empty feature set, we will show in Section 4 how to achieve that result indirectly.

Of course, we may actually want to refer to the natural class of *all* segments, and the representation in (10) shows us how to do so. Since the representation of [ ] refers to the natural class of *all* segments, there is no need for the feature $\pm$SEGMENT (Shen, 2016). Chomsky and Halle (1968, p.64) introduced this feature to distinguish segments from morpheme and word boundary symbols. The use of such a feature was subsequently discussed in various contexts, such as (Hyman, 1985), and sometimes rejected, but more often ignored until a particular need arose, then opportunistically invoked. Independent of the merits of any prior arguments for or against this feature, we now see that a set theoretic representation of targets in which natural classes are represented as sets of sets vitiates the need for a +SEGMENT feature.

3 Sources of emptiness

Before discussing rules that include the empty segment as a potential target, it is important to first discuss how such segments enter into a phonological derivation in the first place. There are at least three ways commonly cited in the literature, using various notations. First, the empty segment can be part of the phonological representation of a morpheme. In this case, { } is present in the *lexicon* (see Borrelli 2013, Côté 2008, Dell and Elmedlaoui 1985, Trommer 2015, among others).[10] Second, the empty segment can be introduced by a phonological insertion rule, analogous to any other segment epenthesis rule.[11] Third, the segment can be derived by a phonological process, namely feature deletion.[12] Since the first two ways of introducing an empty segment is fairly straightforward, in this section we focus on the potential derivation of an empty segment through feature deletion.

In Bale et al. (2014), we argue that feature changing rules must be analyzed as the deletion of features *via* set subtraction followed by insertion *via* unification. In this context, rules of total assimilation, including compensatory lengthening rules, can be modelled by deriving { } from a fully specified underlying segment, then unifying this

---

[10]There is a large literature proposing various kinds of 'empty' segments, including moras, CV-slots, X-slots or syllable constituents unassociated to a segmental melody. Bendjaballah and Haiden (2008) present a scale of eight levels of 'emptiness' relevant to templatic morphology systems alone. We cannot provide an analysis of each case where empty or "ghost" segments have been proposed, but by providing a formalization of one kind of empty element it will be easier to evaluate such proposals and examine the extent to which there is unity of behavior among these elements in terms of well-understood set-theoretic notions.

[11]For example, epenthesis of default vowels that end up as copies of other vowels in the word can be analyzed as phonological insertion of { } along with an associated timing slot. Lengthening of a vowel under particular syllable structure or stress conditions can also be analyzed as insertion of { } and a timing slot. Note that our suggestions are not in conflict with analyses that involve merely inserting a mora or a CV or X timing slot. Such proposals have to be enhanced by a model of how timing slots end up associated with segmental (featural) content. Presumably, the same mechanism can be applied to both epenthesis and lengthening cases.

[12]The segment /{ }/ may perhaps also be introduced by a morphological process, for example, a morphologically induced lengthening. However, this process precedes the phonological derivation, and we won't consider it here.

{ } with the complete feature-set of another segment. The mechanism for filling in features in a derived empty segment will be the same as that used for an underlying empty segment, discussed below in Section 4. Let's consider now a mechanism for deriving { } prior to the feature-filling stage.

Let's suppose there is a language L_1 which has a compensatory lengthening process that deletes all the features in a nasal segment that appears in coda position and then lengthens the preceding vowel, as represented by the mappings from underlying forms (UR) to surface forms (SR) in (12):

(12)   Compensatory lengthening for deletion of nasals in coda

       a. tan-so ⤳ ta:so

       b. tan-upi ⤳ tanupi

       c. tak-so ⤳ takso

       d. tak-upi ⤳ takupi

       e. tem-so ⤳ te:so

       f. tem-upi ⤳ temupi

       g. tum-so ⤳ tu:so

       h. tum-upi ⤳ tumupi

Only forms (a), (e) and (g) show a change from UR to SR, since these are the only forms where the UR has a potential coda nasal. In our system, the derivation of such forms involves two steps after the initial syllabification. First, a rule that removes all the features of the nasal in coda position will apply. Informally, we might be tempted to use the following notation to symbolize such a process:

(13)   The wrong notation: [+NAS] → {   } / in CODA

But this is not quite right given our discussion above. The structural change portion of the rule in our notation refers to the features that are affected, not to a segment or a class of segments. For example, the rule '[+NAS] → {+VELAR } / in CODA' does not represent the replacement of nasal segments in coda position with {+VELAR}, the underspecified segment containing only the feature +VELAR. Instead it represents the process of adding the +VELAR feature to nasal segments in the coda position. To represent a process of removing features, we need a new notational system.

Bale et al. (2014) argue that the traditional arrow '→' in phonology should be deconstructed into set theoretic operations that can be incorporated into basic rules. The operation for removing features from sets is set subtraction. We can take advantage of this operation to derive empty segments. There are at least two possible ways we could do this. One way is to subtract a set from itself, since for any set A, it is the case that $A - A = \emptyset$. A second way is to subtract from a given set A all possible valued features—all combinations of $+$ or $-$ and a feature. We will use this second method.

Let's formally represent a valued feature, such as +VOICE, as an ordered pair where the first member contains a plus or minus symbol and the second contains the name of the feature (e.g., +VOICE = $\langle +, \text{VOICE} \rangle$). Suppose that $\mathcal{V}$ is the Cartesian product of

$\{+,-\}$ and a set $\mathcal{F}$ of all possible features (i.e., $\mathcal{V} = \{+,-\} \times \mathcal{F}$). Thus, $\mathcal{V}$ is the set of all possible matches between a +/- value and a feature. Given this definition, we can turn any feature set x into the empty set by subtracting $\mathcal{V}$ i.e., $x - \mathcal{V} = \emptyset$ for any segment x.[13] With this notation in mind, consider the rule in (14):

(14) $[+\text{NASAL}] - \mathcal{V} \,/$ in CODA

This rule can be interpreted as follows:[14]

(15)   The interpretation of '$[+\text{NASAL}] - \mathcal{V} \,/$ in CODA' is a function from a string $x_1 x_2 x_3 \ldots x_n$ to a string $y_1 y_2 y_3 \ldots y_n$ such that for each x_i where $1 \leq i \leq n$:

   1. If $x_i \in \{z : z \supseteq \{+\text{NASAL}\}\}$ and x_i is in a coda, then $y_i = x_i - \mathcal{V}$,
   2. Otherwise $y_i = x_i$.

Of course we could have simply stipulated that y_i (i.e., the output segment) was the emptyset (i.e., if $x_i \in \{y : y \supseteq \{+\text{NASAL}\}\}$ and x_i is in the coda, then $y_i = \emptyset$), however, we favor the formulation in (15) as it makes use of set subtraction, an operation we already need for other purposes (see Bale et al., 2014). Thus, by using set-subtraction, we relate this rule to other potential rules that may subtract a subset of $\mathcal{V}$ from a segment instead of the complete set.[15]

The target segments of rule (15) are the members of the natural class of nasals in a language, say, {m,n,ŋ}, occurring in coda position. The resulting corresponding segment in the output strings (via set subtraction) is the same for any target segment—it is always the segment { }. We now turn to the next step in the process of compensatory lengthening, a rule that fills in feature values via unification.

4   Emptiness in rules

We have indicated how the segment { } can be introduced into a derivation. We have also pointed out that it is not possible to refer exclusively to this segment in the target or environment of a phonological rule—the only natural class that contains the empty segment also contains every other segment. In this section, we will show that, while it

---

[13]Note that $\mathcal{V}$ contains *all* valued features, $-\text{HIGH}$, $+\text{HIGH}$, *etc.* Recall that set subtraction does not care if the subtracted set contains elements that are not in the set to be subtracted. For example, $\{b,c,d\} - \{b,c\} = \{b,c,d\} - \{a,b,c,e,f,g\} = \{d\}$.

[14]We use the subtraction symbol to represent the function that maps segments that fit the structural description to the same segment minus the structural change. This is a common extension of notation. To be clear, we are *not* using the symbol '$-$' to represent the subtraction of the structural change from the structural description, but rather the subtraction of the structural change from each segment in the class denoted by the structural description.

[15]Our discussion above grants a special status to $\mathcal{V}$, the set of all valued features. However, it remains an open question whether phonological UG actually has access to a symbol for this set, or it is a metalanguage symbol with which phonologists can encode the case of subtraction of all valued features from a segment. This issue is related to the question of whether the set subtractions discussed here are implemented as a single operation or as a series of individual subtractions. It seems to be the case that something like the SSR, discussed in section 4.3, is untenable for subtraction rules. In brief, the reason is that as we apply more and more composed unification rules, the targets of the rules become more and more highly specified; whereas iterating subtraction leads to less and less specified targets.

is true that we cannot target { } intensionally, we can ensure that this segment is the only non-vacuously affected target of a rule. In other words, when the effect of the rule on every *other* segment is vacuous, it can appear extensionally as if only { } is targeted.

4.1 Unification

The critical idea we need, presented in Bale et al. (2014), is that feature-filling is the result of *unification*, which is a partial operation. This idea is probably best explained by contrasting unification with set union. Set union is not partial: for any two sets, A and B, the union of the two, $A \cup B$, is always defined (as the set that includes all the elements of A and all the elements of B, but no other elements). Unification, in contrast, is defined only when its output is *consistent*. For phonological purposes, we define *consistency* as follows:

(16) CONSISTENCY: A set of features ρ is consistent if and only if there is no
     feature $F \in \mathcal{F}$ such that $+F \in \rho$ and $-F \in \rho$.

When Consistency is met, the output of set unification is identical to the output of set union:

(17) UNIFICATION: For any two sets, A and B, the unification of A and B, $A \sqcup B$,
     is defined iff $A \cup B$ is consistent. When defined, $A \sqcup B = A \cup B$.

When $A \sqcup B$ is not defined, we will say that unification *fails*.
    To understand how unification can be used in phonological rules, let's reconsider the Turkish data discussed above, involving two fully specified coronal stops (/t/ and /d/) as well as a coronal stop underspecified for voicing (/D/). In Inkelas' analysis, /D/ is voiced to [d] in onsets, but both /t/ and /d/ surface unchanged. As discussed above, there is no way of targeting just /D/ using the rule notation discussed above (any natural class containing /D/ will also contain /t/ and /d/), however there is a way to make a rule that *appears* to target just /D/ using unification:

(18) [+COR, −SON, −CONT] $\sqcup$ {+Voice} / in Onset position

This rule will have the following interpretation:

(19) The interpretation of '[+COR, −SON, −CONT] $\sqcup$ {+Voice} / in Onset
     position' is the function that maps any string of the form $x_1 x_2 x_3 \ldots x_n$ to the
     string $y_1 y_2 y_3 \ldots y_n$ such that for each i, where $1 \leq i \leq n$:

     1. if $x_i \in [+ \text{COR}, - \text{SON}, - \text{CONT}]$ (i.e., $x_i \supseteq \{+ \text{COR}, - \text{SON}, - \text{CONT}\}$)
        and x_i is in the onset of some syllable, and $x_i \sqcup \{+ \text{VOICE}\}$ is defined,
        then $y_i = x_i \sqcup \{+ \text{VOICE}\}$
     2. otherwise $y_i = x_i$.

The rule targets the natural class that contains the three segments /t, D, d/. Let's see how the rule applies in various cases, assuming monosyllabic forms:

(20) Rule (26) illustrated

    a. Input /tab/: /t/ $\sqcup$ {+VOICE} is undefined, so output is [tab]

    b. Input /Dab/: /D/ is in syllable onset and /D/ $\sqcup$ {+VOICE} = [d], so output is [dab]

    c. Input /dab/: /d/ is in syllable onset and /d/ $\sqcup$ {+VOICE} = [d], so output is [dab]

    d. Input /baD/: /D/ is not in syllable onset, so output is [baD]

In (20a), unification fails, because /t/ is $-$VOICE, so the union of /t/ with {+VOICE} is not consistent. Therefore the output string is the same as the input string, [tab]. In (20b), unification adds the valued feature {+VOICE} to /D/, yielding an output string [dab]. In (20c), unification does not fail, but it is vacuous, since /d/ $\cup$ {+VOICE} is just [d]. The output string is identical to the input string, namely [dab]. In (20d), /D/ is not in a syllable onset, so the output string is identical to the input string, namely [baD]. The phonological rule targets all three segments in the natural class, but has a non-vacuous effect only on the underspecified segment, and only when that segment occurs in an onset.

4.2 Feature filling rules and the empty segment

The preceding example made use of /D/, a segment unspecified for the single feature, VOICE, and a rule that added a value for this feature to segments lacking one. This logic can be extended to segments lacking values for any number of features, even to the empty segment. To illustrate how we can extensionally target the empty segment without doing so intensionally, let's construct a simple example of a phonological process in a language defined within a grammar with a very limited inventory of features. Consider a system with just two binary features $\pm F, \pm G$. If we do not stipulate that segments be complete, that is, if we allow underspecification, these features can be combined into nine different segments (feature sets):

(21) Nine segments definable with two binary features
$\{+F, +G\}, \{+F, -G\}, \{+F\}, \{-F, +G\}, \{-F, -G\}, \{-F\}, \{+G\}, \{-G\}, \{\}$

Now consider a language L_2 that has only five of these segments, the fully specified ones, $\{+F, +G\}, \{+F, -G\}, \{-F, +G\}, \{-F, -G\}$ and the empty segment $\{\ \}$. Suppose that L_2 has a process that assimilates the empty segment to the preceding segment if that segment is $\{+F, +G\}$, so that we observe the following input-to-output mapping:

(22) Input-to-output mapping in L_2:

$$\{+F, +G\}\{\} \rightsquigarrow \{+F, +G\}\{+F, +G\}$$

This process can be informally characterized as stating that the empty segment becomes $\{+F, +G\}$ when it occurs after $\{+F, G\}$:

(23) Find the empty segments that are adjacent to segments with $+F$ and $+G$, and replace those empty segments with the segment $\{+F, +G\}$.

However, the informal process in (23) intensionally targets the empty segment, which, as we discussed in section 2, cannot constitute a natural class. Yet, if we use set-unification as the primary operator, we can write a rule that targets a natural class and yet achieves an almost identical input-to-output mapping as the one expressed by the informal process in (23). Consider the rule in (24).

(24) $[\ ] \sqcup \{+F, +G\} \,/\, [+F, +G]\,\underline{\quad}$

This rule will have the following interpretation:

(25) The interpretation of '$[\ ] \sqcup \{+F, +G\} \,/\, [+F, +G]\underline{\quad}$' is the function that maps any string of the form $x_1 x_2 x_3 \ldots x_n$ to the string $y_1 y_2 y_3 \ldots y_n$ such that for each i, where $1 \leq i \leq n$:

1. if $x_i \in [\ \ ]$ (i.e., $x_i \supseteq \{\}$) and $x_{i-1} \in [+F, +G]$ (i.e., $x_{i-1} \supseteq \{+F, +G\}$) and $x_i \sqcup x_{i-1}$ is defined, then $y_i = x_i \sqcup x_{i-1}$

2. otherwise $y_i = x_i$.

The rule in (24), as it is interpreted in (25), is a function from strings to strings that targets every segment that appears to the right of $\{+F, +G\}$ (since the natural class defined by $[\ ]$ contains all possible segments and the natural class defined by $[+F, +G]$ contains only the segment $\{+F, +G\}$). The rule attempts to unify these targets with $\{+F, +G\}$.

However, the effect of this unification on the target is sometimes vacuous (and the rule application is vacuous), when the target segment in an input string is the segment $\{+F, +G\}$ itself. The rule application is also vacuous when the unification is undefined because the union of a target segment and $\{+F, +G\}$ is inconsistent: unification fails and thus the segment remains unaffected by the rule. The only noticeable (non-vacuous) application of the rule will occur when the target is the empty segment. The relevant possible mappings are presented in (26), where we consider minimal two-segment sequences as inputs to the rule.

(26) Three scenarios for a two-segment sequence of the form $\{+F, +G\}X$

- **Where X is $\{\}$:**
  $\{+F, +G\}\{\} \rightsquigarrow \{+F, +G\}\{+F, +G\}$ (unification applies and fills in values of input $\{\ \}$)
- **Where X is $\{+F, +G\}$:**
  $\{+F, +G\}\{+F, +G\} \rightsquigarrow \{+F, +G\}\{+F, +G\}$ (unification applies vacuously, $\{+F, +G\} \sqcup \{+F, +G\} = \{+F, +G\}$)
- **Where X is $\{+F, -G\}$, $\{-F, +G\}$ or $\{-F, -G\}$:**
  Unification failure—X remains unchanged because set union yields an inconsistent set (see (25) condition 2 above, and the rule semantics in Bale and Reiss 2018).

11

1. $\{+F, +G\}\{+F, -G\} \rightsquigarrow \{+F, +G\}\{+F, -G\}$, since $\{+F, +G\} \sqcup \{+F, -G\}$ is undefined.
2. $\{+F, +G\}\{-F, +G\} \rightsquigarrow \{+F, +G\}\{-F, +G\}$, since $\{+F, +G\} \sqcup \{-F, +G\}$ is undefined.
3. $\{+F, +G\}\{-F, -G\} \rightsquigarrow \{+F, +G\}\{-F, -G\}$, since $\{+F, +G\} \sqcup \{-F, -G\}$ is undefined

Hence, although the rule *intensionally* targets *all* segments to the immediate right of $\{+F, +G\}$, the only time the output is different from the input is when the segment to the right of $\{+F, +G\}$ is the empty set.

This example differs from the Turkish illustration with /t, D, d/ in two ways. First, there are two features involved in the rule, not just one, and second, the second argument of the unification operation happens to be identical to the segment defining the rule environment, in contrast to the condition in Turkish that $+$VOICE is supplied in Onsets. Despite these differences, the logic of the two examples is identical with respect to the mechanics of unification.

4.3   Do we really want to unify segments with non-singleton sets?

In section 4.2 we unified segments with a set containing two valued features. We assumed that the toy language in question L_2 had only five segments, four that are fully specified and one that is completely unspecified. Let's consider a language that contains all nine of the definable segments given the features F and G. We'll assume that this L_3 has the same rule as L_2, repeated in (27):

(27) $[\ ] \sqcup \{+F, +G\} \,/\, [+F, +G] \underline{\quad\quad}$

Here are some cases that arise when the rule applies to strings containing some of the nine definable segments preceded by $\{+F, +G\}$:

(28)   Some applications

    a. $\{+F, +G\}\{\ \ \} \rightsquigarrow \{+F, +G\}\{+F, +G\}$

        Both valued features are filled in.

    b. $\{+F, +G\}\{+F, +G\} \rightsquigarrow \{+F, +G\}\{+F, +G\}$

        Unification is vacuous.

    c. $\{+F, +G\}\{-F, -G\} \rightsquigarrow \{+F, +G\}\{-F, -G\}$

        Unification fails so target is unchanged.

    d. $\{+F, +G\}\{+G\} \rightsquigarrow \{+F, +G\}\{+F, +G\}$

        Unification fills in $+F$, but is vacuous for $+G$.

    e. $\{+F, +G\}\{-G\} \rightsquigarrow \{+F, +G\}\{-G\}$

        Unification fails so target is unchanged.

The logic for all five cases should be clear, but pay particular attention to (e). Since the union $\{-G\} \cup \{+F, +G\} = \{+F, +G, -G\}$, which is inconsistent, it follows that the unification $\{-G\} \sqcup \{+F, +G\}$ is undefined. Therefore, the output string must be the same as the input string, namely $\{+F, +G\}\{-G\}$.

We propose that this might be the wrong result for modelling human phonology. We suspect that a formal system that would yield $\{+F, +G\}\{+F, -G\}$ in such a case is correct. In other words, the conflict with respect to the feature G should not prevent the addition of the valued feature $+F$ via unification. A simple way to achieve this is to suppose that the second argument in a phonological unification must be a singleton set, and thus rule (27) needs to be reformulated in terms of a sequence of two rules, each containing unification with a singleton set. In the remainder of this section we sketch out such a system and show how it achieves the effect we are claiming better reflects the human language faculty. In section 5 we work with data sets and toy languages based on Turkish vowel harmony phenomena to illustrate the kinds of languages we specifically want to rule out. We cannot *prove* the non-existence of such languages, but if no examples are attested, our explicit proposal might explain why.

To give a flavour of the kind of system we intend to develop, let's take a closer look at case (e) in (28) above. As it is, the output of the unification process is the same as the input. However, if we divide the unification process into two rules, one that first attempts to unify the target with $\{+F\}$ and then another that then attempts to unify the target with $\{+G\}$, the output for case (e) would be different even though the output for all the other cases would be the same. For example, if we first unify $\{-G\}$ with $\{+F\}$ we get $\{+F, -G\}$ as an output. Then, if we try to unify this output with $\{+G\}$, the unification will fail. As a result the final output would be $\{+F, -G\}$ rather than just $\{-G\}$. Note that we get the same result if we change the order of the two unification processes. If we first try to unify $\{-G\}$ with $\{+G\}$, the result is undefined, so we get the output $\{-G\}$. If we then unify this output with $\{+F\}$, the result is $\{+F, -G\}$.

We have not proved that this is the correct result, but rather illustrated a distinction that leads to different outputs. However, we will make a strong empirical claim and propose the following as a property of phonological rules:

(29) **Singleton set restriction on unification (SSR)**: The second argument of unification in a phonological rule, the one that corresponds to the structural change, is always a singleton set.

In other words, unification must proceed in a derivation one feature at a time. If the SSR is valid then we need to reformulate the processes expressed in (28a-e). First let's decompose rule (27) to conform to the SSR in (29). We will arbitrarily order unification with $+F$ before unification with $+G$.

(30) Reformulation of (27) using SSR:

- R_1: [ ] $\sqcup \{+F\}$ / $[+F, +G]$ ___
- R_2: [ ] $\sqcup \{+G\}$ / $[+F, +G]$ ___

Each rule is a function from strings to strings, so we can compose the rules in normal fashion. For example, we can apply R_1 to a string $abcd$ and then apply R_2 to the output:

$R_2 \circ R_1(abcd)$. Let's redo the derivations in (28) using the two rules of (30) instead of the one in (27), which we now propose is not a licit human phonological rule at all:

(31)   Some applications

  a. $R_2 \circ R_1(\{+F, +G\}\{\ \}) \rightsquigarrow \{+F, +G\}\{+F, +G\}$

   Both valued features are filled in :$+F$ is filled in on the righthand
   segment by R_1 and .$+G$ is filled in on the righthand segment by R_2.

  b. $R_2 \circ R_1(\{+F, +G\}\{+F, +G\}) \rightsquigarrow \{+F, +G\}\{+F, +G\}$

   Each rule yields a vacuous unification.

  c. $R_2 \circ R_1(\{+F, +G\}\{-F, -G\}) \rightsquigarrow \{+F, +G\}\{-F, -G\}$

   Each unification fails so target is unchanged.

  d. $R_2 \circ R_1(\{+F, +G\}\{+G\}) \rightsquigarrow \{+F, +G\}\{+F, +G\}$

   Unification fills in $+F$, but is vacuous for $+G$.

  e. $R_2 \circ R_1(\{+F, +G\}\{-G\}) \rightsquigarrow \{+F, +G\}\{+F, -G\}$

   Unification fills in $+F$ (by R_1) and unification fails for $+G$, so $-G$
   on the target remains.

The mappings in (31) are identical to those in (28) aside from case (e).

   It is crucial to note that the SSR applies only to structural changes in unification rules. The rules above in (30) have two valued features specified in the environment segment, and thus are different from the following rules which each have a single feature specified in the environment segment:

(32)   Less specific environment specifications

  • R_A: [ ] ⊔ $\{+F\}$ / [$+F$] ___
  • R_B: [ ] ⊔ $\{+G\}$ / [$+G$] ___

The reader can confirm that R_1 in (30) applied to the string $\{+F, -G\}\{\ \}$ yields the identical string $\{+F, -G\}\{\ \}$, whereas R_A in (32) applied to the same input yields $\{+F, -G\}\{+F\}$. This is because R_A does not require that the preceding segment be $+G$, but R_1 does. Think of R_1 as a rule that spreads +VOICE from segments that are +CONTINUANT, whereas R_A is like a rule that spreads +VOICE regardless of other feature values.

4.4   Unification and the SSR with α-notation

We now demonstrate how the system we propose interacts with α-notation. The SSR allows for the single member of the set denoting the structural change to be a feature with a Greek letter variable as its value. This variable denotes identity with other tokens of the same variable elsewhere in the rule:

(33)   [+COR, −SON, −CONT] ⊔ $\{\alpha \text{Voice}\}$ / ___[αVoice]

This rule says to unify each coronal stop with the set containing the voicing value of the following segment. We have to consider six cases, namely strings in which each segment /t, D, d/ occurs before a voiced segment, and strings in which each occurs before a voiceless segment. Let's first consider what happens when each coronal stop occurs before a voiced segment:

(34)   Unification of /t, D, d/ with { +VOICE}

      a. Input /matba/: /t/ ⊔ {+VOICE} is undefined, so output is [matba]

      b. Input /maDba/: /D/ ⊔ {+VOICE} is [d] so output is [madba]

      c. Input /madba/: /d/ ⊔ {+VOICE} is [d] so output is [madba]

In (34a), unification fails, because /t/ is −VOICE, so the union of /t/ with {+VOICE} is not consistent. Therefore the output string has [t]. In (34b), unification adds the valued feature {+VOICE} to /D/, yielding [d] in the output string. In (34c), unification does not fail, but it is vacuous, since /d/ ∪ {+VOICE} is just [d].

Now let's see what happens before a voiceless segment:

(35)   Phonological unification illustrated with { −VOICE}

      a. Input /matpa/: /t/ ⊔ {−VOICE} is [t] so output is [matpa]

      b. Input /maDpa/: /D/ ⊔ {−VOICE} is [t] so output is [matpa]

      c. Input /madpa/: /d/ ⊔ {−VOICE} is undefined, so output is [madpa]

In (35a), unification is vacuous since /t/ is already −VOICE, so the output is [t]. In (35b) unification adds the feature −VOICE to /D/, yielding [t]. In (35c) unification fails, because /d/ is +VOICE, so the union of /d/ with {−VOICE} is not consistent. Therefore the output string has [d].

5   The SSR and vowel harmony

We will now illustrate the implications of the SSR using something closer to attested data. Our ultimate goal is to explore what would be a possible empirical argument for or against the SSR—i.e., evidence for whether there exist unification rules that fill in multiple features on an underspecified segment while 'bypassing' less underspecified segments.

5.1   Turkish vowel harmony

Let's consider a limited dataset illustrating Turkish vowel harmony, one that only contains monosyllabic roots and words with a single suffix. For example, notice the vowel alternations in (36).

(36) Basic Turkish vowel harmony

Nom. Sg	Nom. Pl	Gloss	
ip	ipler	'rope'	
ek	ekler	'joint'	[−Back]
gül	güller	'rose'	
öç	öçler	'revenge'	
kıl	kıllar	'body hair'	
sap	saplar	'stalk'	[+Back]
pul	pullar	'stamp'	
son	sonlar	'end'	

The vowel in the plural suffix always surfaces as −High and −Round, but it agrees with the value for Back on the preceding vowel. One way to represent the underlying phonological form of this suffix is to posit a vowel /A/ which is underspecified for Back. We might imagine that the value for this feature could be filled in by a unification rule that targets /A/, something like this:

(37)   Unify the segment /A/ with $\{\alpha Back\}$ when the preceding vowel is specified $\alpha Back$.

However, we now know that any rule that targets /A/ must target the natural class that includes /e/ and /a/ as well, (since /A/ ⊆ /e/ and /A/ ⊆ /a/) so the correct formulation is more like this:

(38)   Rule targeting natural class:
        $[−\text{High}, −\text{Round}] \sqcup \{\alpha \text{Back}\}$ / when preceding vowel is $[\alpha \text{Back}]$

The effect of the rule on suffixes with fully specified underlying vowels will be vacuous.

It turns out that alternating suffixes with +High vowels also assimilate with respect to Back on a preceding vowel:

Nom. Sg.	Gen. Sg	Gloss	Root vowel class
ip	ipin	'rope'	[−Back, −Round]
ek	ekin	'joint'	
gül	gülün	'rose'	[−Back, +Round]
öç	öçün	'revenge'	
kıl	kılın	'body hair'	[+Back, −Round]
sap	sapın	'stalk'	
pul	pulun	'stamp'	[+Back, +Round]
son	sonun	'end'	

The high suffix vowel in these genitive forms also assimilates to the value of Back on the preceding vowel, just like the non-high vowel in the plural. Rule (38) can be revised to apply to this vowel as well, by removing the specification that the target vowel must be −High and −Round:

(39)   [ ] ⊔ $\{\alpha \text{Back}\}$ / when preceding vowel is specified $\alpha Back$

16

We can thus treat back harmony in high and non-high vowels as a single process.

The non-high vowel of the plural suffix is always −ROUND, but the high vowel of the genitive suffix also alternates with respect to the feature ROUND. Let's assume that underlyingly this high vowel is /I/, specified as +HIGH, but unspecified for both ROUND and BACK. We need another rule to fill in values for ROUND:

(40) $[\ \ ] \sqcup \{\alpha\text{ROUND}\}$ / when preceding vowel is specified αRound

Rounding harmony does not affect the vowel of the plural suffix, but we don't need to specify that the targets of the rounding rule are +HIGH, since unification of sets containing conflicting values of ROUND will fail. So, application of a rule based on unification of any value for ROUND with the −ROUND of /A/ will be vacuous (either because /A/ is vacuously unified with $\{-\text{ROUND}\}$, or because unification of /A/ with $\{+\text{ROUND}\}$ fails due to inconsistency). Note that the two rules in (39) and (40) happen to obey the SSR.

Unconstrained by the SSR, we could posit an alternative to the two rules (39) and (40). We could propose to use (38) to fill in BACK on /A/ and then use a single rule to fill in both underspecified features on an alternating high vowel suffix /I/, something like (41):[16]

(41) $[+\text{HIGH}] \sqcup \left\{ \begin{array}{c} \alpha\text{BACK} \\ \beta\text{ROUND} \end{array} \right\}$ / when preceding vowel is specified $\begin{array}{c} \alpha\text{BACK} \\ \beta\text{ROUND} \end{array}$

However, this would treat the BACK assimilation as being due to two separate rules, a result that we think misses a critical generalization.

In contrast, by insisting that rules are formulated in accordance with the singleton set restriction, BACK assimilation would be due to a single process. To see how, let's go through a derivation. Consider again the rules in (39, 40), repeated in (42):

(42)   Turkish with simple rules

R_a. $[\ \ ] \sqcup \{\ \alpha\text{BACK}\ \}$ / when preceding vowel is specified αBACK

R_b. $[\ \ ] \sqcup \{\ \beta\text{ROUND}\ \}$ / when preceding vowel is specified βROUND

The rules can apply in either order, but we illustrate with BACK assimilation applied first. The formula in (43) shows the composition of the two rules applied to the input /ip-lAr/. This composed function has the effect of first 'attempting' to add a value for BACK to each target segment, and then attempting to add a value for ROUND, with one application of unification for each feature.

(43) $R_b \circ R_a(\text{ip-lAr})$

In the derivations below, the abbreviation 'v.u.' refers to 'vacuous unification', for example, when a −ROUND vowel unifies with $\{-\text{ROUND}\}$. The abbreviation 'u.f.' refers to 'unification failure', another kind of vacuous application which occurs, for example, when a −ROUND vowel unifies with $\{+\text{ROUND}\}$. The tables show **only the features that get added** to the suffix vowels by each rule.

---

[16]The environment could equivalently be specified as "when preceding vowel is a superset of $\{\alpha\text{BACK}, \beta\text{ROUND}\}$."

(44)   Turkish plural derivations: /A/ is {−HIGH, −ROUND}

UR	ip-lAr	ek-lAr	gül-lAr	öç-lAr	kıl-lAr	sap-lAr	pul-lAr	son-lAr
$R_a : \sqcup\{\alpha\text{BK}\}$	−BK	−BK	−BK	−BK	+BK	+BK	+BK	+BK
$R_b : \sqcup\{\alpha\text{RD}\}$	v.u.	v.u.	u.f.	u.f.	v.u.	v.u.	u.f	u.f.
SR	ipler	ekler	güler	öçler	kıllar	saplar	pullar	sonlar

So, in the course of the derivation, the vowel of /ip-lAr/ starts out as {−HIGH, −ROUND}, then receives −BACK via rule R_a, and is unaffected by rule R_b. It surfaces as [e]. Let's see what happens with the suffix containing /I/:

(45)   Turkish genitive derivations: /I/ is {+HIGH}

UR	ip-In	ek-In	gül-In	öç-In	kıl-In	sap-In	pul-In	son-In
$\sqcup\{\alpha\text{BK}\}$	- BK	- BK	- BK	- BK	+ BK	+ BK	+ BK	+ BK
$\sqcup\{\alpha\text{RD}\}$	−RD	−RD	+RD	+RD	−RD	−RD	+RD	+RD
SR	ipin	ekin	gülin	öçin	kılın	sapın	pulun	sonun

In the course of the derivation, the suffix vowel of /ip-In/ starts out as {−HIGH}, then receives −BACK via rule R_a, and −ROUND via rule R_b. It surfaces as [i].

5.2   Hypothetical languages: Purkish and Surkish

We have just shown that we can derive the surface forms of Turkish using one rule per harmonizing feature. In other words, the SSR is not too weak to model Turkish. Since the SSR restricts set of possible grammars (i.e., any grammar that obeys the SSR is also a grammar in a phonological system that does not enforce the SSR), we will adopt the SSR as a guiding principle unless the phonological data can be better explained by adopting more complex rules. In this section, we discuss some of the differences between a phonological theory that is constrained by the SSR and one that is not. We begin by first presenting a possible variation of Turkish, which we call Purkish. Purkish is identical to Turkish except that it has a third underlying underspecified vowel, namely /U/ = {+HIGH, +ROUND}. Critically /U/ has no specification for BACK. We hypothesize that this vowel also undergoes assimilation with respect to the previous vowel (i.e., it surfaces as having the same BACK specification as the previous vowel). As we discuss below, not only do rules that obey the SSR provide a better account of Purkish, but the analysis reveals the limits of the SSR.

Let's consider some details of Purkish. By hypothesis, this hypothetical language has the same eight surface vowels as Turkish, as well as the underlyingly partially

specified /A/ and /I/. Also by hypothesis, Purkish /A/ and /I/ show the same exact alternations as their Turkish counterparts in the same environments. It is thus possible to posit the same rules for the two languages in order to account for this alternation. However, as mentioned above, Purkish also has another partially specified vowel /U/, let's say realized on a comitative suffix /-Uk/. Critically, /U/ is more specified than /I/. Note that there is no way to target, in a rule, the vowel /I/ without also targeting the more specified vowel /U/, because any natural class that contains /I/, also contains /U/.

Now, if /U/ assimilates with respect to the BACK specification of the previous vowel (as mentioned above), it is relevant to note that only one of the rule systems hypothesized in the previous section will be able to account for the input to output mapping. For example, consider a derivation with the two rules that obey the SSR in the derivation table in (46).

(46) With Singleton Set Restriction
      Purkish comitative derivations: /U/ is {+HIGH, +ROUND}

UR	ip-Uk	ek-Uk	gül-Uk	öç-Uk	kıl-Uk	sap-Uk	pul-Uk	son-Uk
$\sqcup\{\alpha\text{BK}\}$	$-$ BK	$-$ BK	$-$ BK	$-$ BK	$+$ BK	$+$ BK	$+$ BK	$+$ BK
$\sqcup\{\alpha\text{RD}\}$	u.f.	u.f.	v.u.	v.u.	u.f.	u.f.	v.u.	v.u.
SR	ipük	ekük	gülük	öçük	kıluk	sapuk	puluk	sonuk

The vowel in the suffix /-Uk/ always surfaces with the feature +ROUND, since that is the underlying value, unlike the underlying /I/ which is unspecified for ROUND. As a result, the rule that 'attempts' to unify /U/ with the ROUND specification of the previous vowel will always have no effect, either due to unification failure or due to vacuous unification. In contrast, /U/ will be affected by the rule that 'attempts' to unify it with the BACK specification of the previous vowel, since /U/ is underspecified for this feature. The end result is a system that maps /U/ to [ü] or [u].

Let's reconsider the same underlying forms with respect to the rules in the previous section that violated the SSR. The effect of these rules is quite different, as outlined in (41).

(47) No Singleton Set Restriction
      Purkish comitative derivations: /U/ is {+HIGH, +ROUND}

UR	ip-Uk	ek-Uk	gül-Uk	öç-Uk	kıl-Uk	sap-Uk	pul-Uk	son-Uk
$\sqcup\left\{\begin{matrix}\alpha\text{BK}\\\beta\text{RD}\end{matrix}\right\}$	u.f.	u.f.	$-$BK $+$RD	$-$BK $+$RD	u.f.	u.f.	$+$BK $+$RD	$+$BK $+$RD
SR	ipUk	ekUk	gülük	öçük	kılUk	sapUk	puluk	sonuk

The SRs in (47) have three different mappings of underlying /U/—it maps /U/ to [ü], [u], and [U]. If Purkish does indeed have the surface forms represented in (46), then

the rule in (47) is unable to derived the right result. To get the right result and maintain the rule that violates the SSR, one would need to posit a further rule to provide a BACK value on the suffix of roots that are −ROUND. This seems like an unnecessary complication to our grammar.

However, not only is the added complexity a problem, the mappings in (47) also demonstrate a key limitation of a grammar that obeys the SSR. Following our natural class logic, it is impossible to fill in values of BACK on the less-specified /I/ but not do so on the more specified /U/. In other words, a system with the input-output mappings of (47), call it Surkish, is not a possible human language if we assume the SSR.[17] Thus, to argue against the SSR, one would have to demonstrate that input to output mappings like in (47) provide a better analysis of some attested language. We do not know of any examples of such languages, and hence we propose adopting the SSR as a working hypothesis for a restricted model of phonological UG.

6 Compensatory lengthening as total assimilation

Let's return to our example of compensatory lengthening in L_1 above, where we saw that unification with full feature sets occurred in forms (12a), (12e) and (12g). The input-output mappings in (12) are repeated in (48) below:

(48) Compensatory lengthening for deletion of nasals in coda

      a. tan-so ⤳ ta:so

      b. tan-upi ⤳ tanupi

      c. tak-so ⤳ takso

      d. tak-upi ⤳ takupi

      e. tem-so ⤳ te:so

      f. tem-upi ⤳ temupi

      g. tum-so ⤳ tu:so

      h. tum-upi ⤳ tumupi

The first rule we need deletes all the features of a coda nasal, creating an empty segment { }, as we saw in (14), repeated in (49):

(49) $[+\text{NASAL}] − \mathcal{V}$ / in CODA

This rule creates the empty segment from nasals that are already to the right of a vowel by virtue of being in coda position. Strings containing intervocalic nasals, which are presumably in onset position, are unaffected by this rule.

To model compensatory lengthening of the vowel as total assimilation, the fully underspecified segments created by rule (49) would need to take on whatever values the immediately preceding vowel segments have. There is more than one member in

---

[17]This claim relates to the controversial issue of segment 'closeness' or rules 'skipping' segments, also known as *saltations* (Hayes & White, 2015; Reiss, 1995, 1996, 2019, and references therein).

the class of segments defining the environment. In our examples, sometimes it's the features of /a/ that get filled in, and sometimes it's the features of /e/ that get filled in, and sometimes it is the features of /u/ that get filled in. We can use α-notation to fill in the correct values by composing the unification rules for each vowel feature. Of course, we need to extensionally target the derived empty segments that result from rule (49), but our unification rules will have to target *all* segments (since this is the only natural class that the empty segment belongs to). However, for all fully specified segments targeted by the rule, unification either fails or is vacuous, so the the output of the unification rules are identical to the inputs.[18]

Let's spell out this process in more detail with formal rules. If we suppose that the set of all features is $\{F_1 \ldots F_n\}$, then we can understand the process of implementing total assimilation as follows. If there were only one feature, the rule would be this:

(50) Unification with one particular feature

$$R_1 = [\ \ ] \sqcup \{\alpha F_1\}/[\alpha F_1]\underline{\quad}$$

So for each feature F_i there is a rule of this form:

(51) General form of rule

$$R_i = [\ \ ] \sqcup \{\alpha F_i\}/[\alpha F_i]\underline{\quad}$$

If there were just two features available, total assimilation would be expressed as the composition of the two rules R_1 and R_2 in either order: $R_2 \circ R_1$ or $R_1 \circ R_2$, with both orders giving identical results. Obviously, we can generalize this for rules R_1 to R_n for any number of features. If there are five features, then total assimilation involves composition of the five rules in some order, where once again the order of composition does not matter.

Since the order of composition does not matter, let's adopt an order based on the number of features in our inventory. Let's assume that the n features are ordered from 1 to n and the respective rules are applied in the same order:[19]

(52) Composition of vowel feature copy rules: $R_n \circ R_{n-1} \ldots R_2 \circ R_1$

For example, perhaps F_1 is ROUND, F_2 is HIGH, F_3 is ATR, F_4 is BACK, F_5 is LOW, and so on. We'll call the subtraction rule in (49) R_0. (Recall that this rule needs to apply first in order to derive the empty segment before applying the assimilation rules.) Now the derivation from the string /tumso/ to the surface form *tu:so* in (48g) involves something like the following changes in the target segment of subsequent rules:

(53) Derivation: $R_n \circ R_{n-1} \ldots R_2 \circ R_1 \circ R_0(\text{tumso}) \rightsquigarrow [\text{tu:so}]$

---

[18]We make the non-trivial assumption here that all underlying segments in the language are fully specified for all features. As stated, the rules below will attempt to unify the features of /a/ with the /k/ of *takso*, for example, but each rule application will be vacuous, since there are no values to fill-in on /k/, by hypothesis. The potentially more reasonable assumption that a segment like /k/ is underspecified even on the surface for vowel features like ROUND and BACK significantly complicates the analysis, a topic that will have to await further research.

[19]The logic remains the same even if only rules for some of the n features are used.

	/tumso/	
UR		
R_0: Subtract $\mathcal{V}$ from coda $[-\textsc{Nasal}]$	tu$\{\ \}$so	(coda *m* loses all features)
R_1: $[\ \ ] \sqcup \{\alpha\textsc{Round}\}$	tu$\{+\textsc{Round}\}$so	(because *u* is $+\textsc{Round}$)
R_2: $[\ \ ] \sqcup \{\alpha\textsc{High}\}$	tu$\left\{\begin{array}{l}+\textsc{Round}\\ +\textsc{High}\end{array}\right\}$so	(because *u* is $+\textsc{High}$)
R_3: $[\ \ ] \sqcup \{\alpha\textsc{Atr}\}$	tu$\left\{\begin{array}{l}+\textsc{Round}\\ +\textsc{High}\\ +\textsc{Atr}\end{array}\right\}$so	(because *u* is $+\textsc{Atr}$)
R_4: $[\ \ ] \sqcup \{\alpha\textsc{Back}\}$	tu$\left\{\begin{array}{l}+\textsc{Round}\\ +\textsc{High}\\ +\textsc{Atr}\\ +\textsc{Back}\end{array}\right\}$so	(because *u* is $+\textsc{Back}$)
R_5: $[\ \ ] \sqcup \{\alpha\textsc{Low}\}$	tu$\left\{\begin{array}{l}+\textsc{Round}\\ +\textsc{High}\\ +\textsc{Atr}\\ +\textsc{Back}\\ -\textsc{Low}\end{array}\right\}$so	(because *u* is $-\textsc{Low}$)
	$\vdots$	
SR	[tuuso] = [tu:so]	

Of course, the target for each of the unification rules is the natural class of *all* segments, but the only place we see non-vacuous effects of unification is in the ever-more-specified, formerly-coda-nasal segments, the ones that lost all their features via the first rule.

It may seem dissatisfactory that there is not a single rule of total assimilation at work when we see a process like compensatory lengthening, but rather a composition of n rules for each of the n features. However, such dissatisfaction is misguided. First, our approach treats total assimilation as just a special case of partial assimilation, and we know that there are many cases of partial assimilation—assimilation of place features and assimilation of laryngeal features, for example. In fact, treating both processes as involving the same types of rules makes a key prediction, namely there cannot be total assimilation without there also being partial assimilation (since rules that "fill in" the empty segment will also "fill in" partially underspecified segments). Second, there is an implicit mistake in wondering why all the individual assimilations happen to be ordered consecutively in a case of total (or multiple) assimilation. In fact, it is the lack of evidence that other rules intervene among the individual assimilations that creates the illusion that they constitute a single process.

7 Conclusions

Our set-theoretic treatment of segments provides mechanisms to address the problem of referring to underspecified segments, including the empty segment, while still maintaining the principle that rules target natural classes. We hypothesize a rule system that unifies underspecified segments with valued features in a piecewise fashion—one unification rule per feature. Critical for our purposes, unification is a partial operation, and it fails whenever its outcome would be inconsistent. The result of this failure is that non-vacuous unification applies only when filling in unspecified values. We have suggested that these simple formal techniques are applicable to a range of widely discussed phonological phenomena. In phonological terms these processes correspond to phenomena that go by the names of assimilation, compensatory lengthening, copy vowels and templatic morphology. By attempting to unify these various phenomena we further the goal of linguistic theory "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" (Chomsky, 2000).

References

Bale, A., Papillon, M., & Reiss, C. (2014). Targeting underspecified segments: A formal analysis of feature changing and feature filling rules. *Lingua*, *148*, 240-253.

Bale, A., & Reiss, C. (2018). *Phonology: A formal introduction*. MIT Press.

Bendjaballah, S., & Haiden, M. (2008). A typology of emptiness in templates. sounds of silence. In J. Hartmann, V. Hegedus, & H. van Riemsdijk (Eds.), *Empty elements in syntax and phonology* (p. 23-59). Amsterdam: Elsevier.

Borrelli, D. A. (2013). *Raddoppiamento sintattico in italian: a synchronic and diachronic cross-dialectical study*. Routledge.

Chomsky, N. (2000). Language as a natural object. In *New horizons in the study of language and mind* (p. 106-133). Cambridge: Cambridge University Press.

Chomsky, N., & Halle, M. (1968). *The sound pattern of English*. New York: Harper & Row.

Côté, M.-H. (2008). Empty elements in schwa, liaison and h-aspiré: The French Holy Trinity revisited. In J. Hartmann, V. Hegedus, & H. van Riemsdijk (Eds.), *Empty elements in syntax and phonology* (pp. 61–103). Amsterdam: Elsevier.

Dell, F., & Elmedlaoui, M. (1985). Syllabic consonants and syllabification in imdlawn tashlhiyt berber. *Journal of African Languages and Linguistics*, *7*(2), 105–130.

Flemming, E. (2005). Deriving natural classes in phonology. *Lingua*, *115*(3), 287–309.

Halle, M. (1964). On the bases of phonology. In J. J. Katz & J. Fodor (Eds.), *The structure of language* (p. 324-333). Englewood Cliffs, NJ: Prentice-Hall.

Hayes, B., & White, J. (2015). Saltation and the p-map. *Phonology*, *32*(2), 267–302.

Hyman, L. (1985). *A theory of phonological weight*. Dordrecht: Foris.

Inkelas, S. (1995). The consequences of optimization for underspecification. In J. Beckman (Ed.), *Proceedings of the north east linguistic society 25* (pp. 287–

302). University of Pennsylvania: Graduate Linguistic Student Association.

McCarthy, J. (1988). Feature geometry and dependency: a review. *Phonetica*, *45*, 84-108.

Mielke, J. (2008). *The emergence of distinctive features*. Oxford: Oxford University Press.

Peterson, G. E., & Harary, F. (1961). Foundations of phonemic theory. In R. Jakobson (Ed.), *Structure of language and its mathematical aspects* (Vol. 12, pp. 139–165). American Mathematical Soc.

Reiss, C. (1995). *A theory of assimilation, with special reference to Old Icelandic phonology* (Ph.D.). Harvard University, Department of Linguistics.

Reiss, C. (1996). Deriving an implicational universal in a constrained OT grammar. In K. Kusumoto (Ed.), *Proceedings of north east linguistic society* (pp. 303–318). Harvard University and MIT: Graduate Linguistic Student Association.

Reiss, C. (2003). Quantification in structural descriptions: Attested and unattested patterns. *The Linguistic Review*, *20*(2-4), 305–338.

Reiss, C. (2019). *Are saltations real?* (Ms. Concordia University)

Sagey, E. C. (1986). *The representation of features and relations in non-linear phonology* (Unpublished doctoral dissertation). Massachusetts Institute of Technology.

Shen, D. T.-C. (2016). *Precedence and Search: Primitive concepts in morphophonology*. Taipei: National Taiwan Normal University dissertation.

Trommer, J. (2015). Moraic affixes and morphological colors in Dinka. *Linguistic Inquiry*, *46*(1), 77-112.