String Phonology

Ollie Sayeed, Christ’s College, Cambridge, ohbs2@cam.ac.uk - first draft for ConSOLE XXV

1. Outline

I’m going to present a model of phonology, which I call String Phonology, which maps strings to strings. In line with substance-free and evolutionary phonology, the model doesn’t try to limit phonology to ‘natural’ rules; instead, it’s supposed to be maximally general, performing any sort of computation on strings given the pieces of machinery that are needed to account for attested rules.

In section 2, I discuss some preliminaries in methodology around how to pick a theory of phonology from the evidence. In section 3, I introduce the model; in section 4, I define the operations that produce allowable strings; and in section 5, I explain the ‘delete-and-unify’ model of how rules act on strings. In section 6, I discuss what sorts of rules are possible and impossible under this model.

2. How to decide how phonology works

The job of phonological theory, I take it, is to come up with a formal model of the mental system speakers use to carry out phonological computation. In other words, we’re *not* just trying to come up with a sort of notation system for describing languages; phonology is a mentalistic enterprise, and we're making claims about minds.

This goal is generally shared, in theory, by generative linguistics; e.g. Chomsky (2004):

“*I do not know why I never realized that clearly before, but it seems obvious, when you think about it, that the notion language is a much more abstract notion than the notion of grammar. The reason is that grammars have to have a real existence, that is, there is something in your brain that corresponds to the grammar. That’s got to be true. But there is nothing in the real world corresponding to language.*”

What we should aim for as theoretical linguists is the *grammar* - the *system* that generates human language - and not just a description of the set of languages that happen to exist (Hale and Reiss, 2008). But in practice, most generative linguists don’t do this. When it comes to evaluating grammars, phonological theories tend to treat being ‘restrictive’ as a virtue - to fit the output of the grammar tightly to the set of languages that exist.

In particular, almost all phonologists seem to think it’s a good thing to restrict the grammar to only generating rules that are phonetically ‘natural’. Natural Phonology (Donegan and Stampe, 1979) proposes an innate set of universal ‘processes’, designed to be phonetically sensible fortitions and lenitions. Optimality Theory (Prince and Smolensky, 1993) proposes an innate set of universal constraints, CON, which represent universally ‘marked’ output configurations; the only way languages are allowed to differ is in their ranking of these constraints. In the SPE model (Chomsky and Halle, 1968), there are constraints on the grammar to rule out particular sorts of language that would otherwise be allowable, on the grounds of naturalness: there's a constraint against asymmetrical vowel systems, for example. Autosegmental Phonology (Goldsmith, 1976) imposes a strong restriction on rules to only be able to spread 'natural classes' of phonological features. In Feature Geometry (Clements, 1985; Halle, 1995), this restriction becomes even stronger, with rules only able to spread certain privileged natural classes of features. The reasoning, in each case, is explicit: "X is unattested, and so we should add restrictions to the grammar to rule it out". If some process doesn't exist, it must be impossible.

An alternative tradition recognizes the fact that this mindset fails to do what its proponents all agree linguistics should do: characterize mental systems. The fact that some type of process isn't attested doesn't mean the grammar is unable to compute it: that would be conflating attested languages with possible languages.

Instead, there are principled *extralinguistic* reasons why some linguistically computable processes might not appear in the languages of the world (Blevins, 2004; Newmeyer, 2005; Ohala, 2005; Hale and Reiss, 2008). Phonetically unnatural rules are less likely to appear diachronically - for phonetic reasons - and so less likely to become phonologized into grammars. There's no need to assume the phonological component is also unable to *compute* them; that would duplicate our explanations with no good reason. The idea of 'markedness' has no new explanatory power. As well as being unnecessary, the 'substanceful' approach is empirically wrong: there are well-attested 'unnatural' rules (Anderson, 1981; Mielke, 2008), meaning the mindset of biasing the architecture against them is a doomed enterprise.

Kaplan (1987) warns against the 'Substance Seduction':

"*But the problem is that, at least in the current state of the art, they don't know which generalizations and restrictions are really going to be true and correct, and which are either accidental, uninteresting or false. The data just isn't in; indeed, the definitive data may in fact be psychological and not linguistic in nature. So if we try to restrict our formalisms by taking substance into account, what we think is true of possible languages, we're apt to make a number of mistakes, some of which have undesirable consequences. Premature identification of substantive generalizations may lead to grammatical descriptions that complicate, or even defy, formal specification.*"

Here, the 'uninteresting' generalizations and restrictions (to theoretical linguists interested in a model of the *grammar*) are the extralinguistic explanations in terms of phonetics, etc. Really, it's a 'grammar-external' theory, about language change and acquisition, that should make predictions about what patterns we should see across the world's languages. It represents a deliberate category mistake to conflate two sorts of fact - 'attested' and 'possible', or 'marked' and 'biased against', or 'natural' and 'biased in favour of' - into a theory of the grammar.

In sum, we should follow the 'substance-free' school and reject the philosophy that biases the grammar towards attested languages. Conflating the grammatical system with all the other extralinguistic factors that interact in producing the set of attested languages means that most generative phonologists fail on their own terms: they provide a good formal system for describing the languages of the world, but a bad one for a theory of humans' mental capacity for phonology.

The question phonologists then face is how to use the data in front of them to come up with a formal system that can account for the facts without being unnecessarily tightly tied to the data, without going to the other extreme and having an 'anything goes' theory. I offer a sensible-sounding methodological principle for deciding on a level of machinery that abides by the substance-free philosophy:

The Cheapskate's Principle: Propose as little machinery as possible, but put as few constraints on that machinery as possible.

In other words, propose the machinery we need to account for the languages we can see, but avoid constraining that machinery without good reason. It might seem paradoxical to avoid proposing unattested machinery but to embrace unattested processes that can run on our current machinery, but the logic follows from Ockham's razor and minimizing assumptions. It would be an extra assumption to assume the grammar has access to more machinery than we can see at work; but it would also be an extra assumption to ban our machinery from generating patterns it would otherwise be able to compute.

3. A picture of phonology

The model of phonology I'm proposing here, which I call String Phonology, is meant to hit the 'middle level' defined by the Cheapskate's Principle: we give phonology access to just the machinery it needs to generate attested grammars, but we don't put any further constraints on that machinery. Rather than building concerns about preferred (or 'natural') rules into the grammar, we end up with a very general system that maps strings to strings.

Without justifying it here[1], I'm taking it that phonology consists of:

- a formal system of *symbols*, qualitatively different things from the muscle movements (or the acoustic signal) that make up speech
- 'underlying' and 'surface' levels of representation using the same alphabet of symbols
- ordered rules, not constraints, defining the mapping between these two levels of representation
- symbols that encode phonological 'features', which combine to create composite objects like 'segments' and 'strings'.

[1] I'm not claiming these bullet points are logically necessary - they do have to be justified somewhere. Any generative phonology textbook, like Kenstowicz and Kisseberth (1979), will have some good (and some bad) arguments.

I'm going to give a set-theoretic description of phonological representations and the rules that act on them.

For a 'set theory of phonology', we want to be able to define as much as possible from some collection of primitive objects using set-theoretic operations; the goal is that everything else can be built out of these primitive sets of objects by the normal operations of subsethood, intersection, union, complementation, pair-set building, power set.

3.1. The alphabet

String Phonology has six sorts of primitive object that appear in representations:

- a finite set V of *features*;
- an infinite set I of *timing slots*;
- an infinite set of variables;
- a 's*pace'* symbol '…'
- a set of *boundary* symbols: word boundary, phrase boundary, etc.[2]
- a set of *operation* symbols like OR, NOT, and AND.

3.1.1. Features and segments

The elements of V, features, are themselves sets of *feature values*: the feature [voice] is the set {[+voice], [$-$voice]}, for example.[3][4] For each feature $F \in V$, I take it there exists a set of variable symbols [αF], [βF], [γF] (see section 4.5).

Define a *segment* S to be the image of a (partial) choice function on V - in other words, a set of feature values, containing at most one value from a given feature. In this definition, a segment can't be both [+voice] and [$-$ voice], but it could be not specified as either. Call the set of segments Σ.

Rather than writing them out explicitly, we can refer to sets (or 'bundles') of features using their own symbols as shorthand, like /b/ for the set {[+voice], [+labial], [$-$ sonorant], [+consonant], …}. A segment doesn't need to contain a value of every feature; we can use /B/ to refer to the segment {[+labial], [$-$ sonorant], [+consonant], …}, which is like /b/ except for not containing either value of the feature [voice].

[2] I'm following Samuels (2009) in assuming a flat phonology as the null hypothesis; in principle there could be symbols like 'syllable boundary' or 'clitic group boundary' within this framework.

[3] I'm leaving it open as to whether every feature has to have exactly two values, or we can have *n*-ary features for any *n* (like one-valued place features or three-valued height features, say; see Gnanadesikan (1997) for arguments for the latter).

[4] In this model, the labels like + and - are just a phonologists' convention, following the tradition of the generative notation; they aren't meant to be parts of the representation in the grammar.

3.1.2. Strings

Define a *simple string* W to be a function from I to Σ. In other words, a simple string is a set of ordered pairs of timing slots[5] $i \in I$ and segments $S \in \Sigma$, such that each timing slot is mapped to at most one segment. There are infinitely many timing slots, so there's no arbitrary highest length that a string is allowed to be.

These timing slots need to be ordered in time (see Raimy, 2000 for discussion of the precedence relation in phonology). Define a relation $<$ as a strong total order of I, meaning that:

i) for all $i, j \in I$, it holds that $i < j$, $j < i$, or $i = j$;
ii) there's no $i \in I$ such that $i < i$;
iii) for all $i, j, k \in I$, if $i < j$ and $j < k$, then $i < k$.[6]

For two ordered pairs $(i,S_1),(j,S_2) \in W$, $i < j$ means that S_1 is articulated before S_2 in the pronunciation of W. The simple string *cat*, then, is a set $\{(i_1,/\text{k}/),(i_2,/\text{æ}/),(i_3,/\text{t}/)\}$.[7] The set of *strings* is the closure of the set of simple strings under the string operations, as I'll define in the next section.

3.1.3. The need for strings

The reason for proposing strings is that in general, phonology isn't restricted to rules that map single segments to single segments. There are attested rules that are either impossible or awkward to account for in a model that only allows one-segment-to-one-segment changes:

No segments to one:
- Insertion (e.g. English [r]-insertion; Wells, 1982)

One segment to none:
- Deletion (e.g. Portuguese vowel deletion; Silva, 1997)

One segment to many:
- Diphthongization (e.g. Spanish /o/ $\rightarrow$ [we]; Albright et al., 2001);
- Consonant gemination (e.g. Italian Raddoppiamento Sintattico; Napoli and Nespor, 1979);
- Vowel lengthening (e.g. Chickasaw final lengthening; Gordon and Munro, 2007);

---

[5] These aren't the same as the timing slots of autosegmental phonology; String Phonology is linear.

[6] It would be convenient to just inject I into the natural numbers $\mathbb{N}$ and then define $<$ as the inverse image of the standard ordering on $\mathbb{N}$ - but then the timing slots' values in $\mathbb{N}$ would be too 'close' to do arbitrarily many insertions of new timing slots, which we'll need to do to capture insertion rules in this model.

[7] We can take it that long vowels and geminate consonants consist of multiple identical segments (with no OCP in this theory). This doesn't in itself capture the difference between affricates and clusters like the Polish *czy* and *trzy*, or between 'monomoraic' and 'bimoraic' diphthongs (as in Old English; Lass 1994). These distinctions will need to be encoded in some other way. See section 4.3.2 for discussion of long and short vowels in Portuguese.

- Affrication (e.g. Ha'ili Arabic /k/ $\rightarrow$ [ts]; Alrasheedi, 2015)

Many segments to one:
- Monophthongization (e.g. Sanskrit /ai/, /au/ $\rightarrow$ [e], [o]; Zwicky, 1965);
- Degemination (e.g. Swedish degemination; Andersson, in press.);
- Vowel shortening (e.g. Luganda final vowel shortening; Hyman and Katamba, 1990);
- Cluster simplification (e.g. Swedish /rt/ $\rightarrow$ [ʈ])

Many segments to many:
- Metathesis (e.g. Faroese /sk/ $\rightarrow$ [ks]; Buckley, 2011)

String Phonology extends the Bale, Papillon and Reiss (2014) concept **(**see section 5.1.2**)** to account for these rules, by giving phonology the power to map whole strings to strings.

3.2 What a rule looks like

The above is the 'static' phonology - representations are bundles of features paired with timing slots. The 'dynamic' phonology is in the set of rules, which are mappings from the set of strings to itself. In this model, a rule R is an ordered triple (T, E, C):

- a *structural description* T (a string)
- an *environment* $E = (E_L, E_R)$ (an ordered pair of strings; a 'left-hand environment' and a 'right-hand environment')
- a *structural change* C (another ordered pair of strings).[8]

The interpretation is that any string that 'matches' the string in the target T and is flanked by the pair of strings E changes in a way defined by the strings in C, which we'll make more precise in the next section. For example, say we have a rule that devoices final obstruents at word ends. The structural description defines the set of segments acted on by the rule:

$T = \{(i, \{[-\text{ son }]\})\}$

This rule will act on any segment specified as [– son] (linked to an arbitrary timing slot i). The right-hand environment is the word end:

$E_R = \{(i, \#)\}$

The rule will act on any [$-$ son] segment, when followed immediately by the word boundary symbol #. The left-hand environment in this rule is empty; any string of symbols to the left of the string targeted by the structural description will satisfy the environment condition. We'll come to what the structural change does in section 5.

---

[8] All of these strings are *finite*; we obviously don't let the grammar store an infinite amount of information by hand about each phonological rule.

3.3. String subsumption

Here, I give a technical definition of what it means for a string to 'match' another string: intuitively, we mean that the first string W_1 can be laid on top of W_2 such that their overlapping segments don't mismatch.

Say a simple string $W_1 : I \to \Sigma$ *subsumes* another simple string $W_2 : J \to \Sigma$ iff there are injections $L : W_1 \to W_2$, $M : I \to J$, and $N : \Sigma \to \Sigma$ such that:

i) for all $(i,S) \in W_1$, $L(i,S) = (M(i),N(S))$;
ii) for all $(i_1,S_1),(i_2,S_2) \in W_1$, if $i_1 < i_2$ under the ordering on I, then $M(i_1) < M(i_2)$ under the ordering on J;
iii) for all $i_1,i_2 \in I$, if there is a j in J such that j is a member of an ordered pair in W_2 and $M(i_1) < j < M(i_2)$, then j is in the image of I under M;
iv) for all S such that $(i,S) \in W_1$, $S \subseteq N(S)$.

In other words, L maps each timing slot-segment pair (i,S) in the first string to a new pair $(M(i),N(S))$ in the second string, at a new timing slot $M(i)$ with a new segment $N(S)$. The mapping M is order-preserving: if i_1 is before i_2 in the first string, their counterparts under M will be in the same order in the second string. There are no 'gaps' in the mapping: W_1 has to be laid over a contiguous subpart of W_2. And there are no mismatches: every feature bundle S of the first string is compatible with its counterpart $N(S)$ in the second string, in that there are no features in S that aren't in $N(S)$.

In sum, the map M is an order-preserving map between the sets I, J of timing slots with no gaps, meaning that M lays W_1 on top of a contiguous subpart of W_2; and N is a map such that every segment S of W_1 is a subset of its image $N(S)$ in W_2. This defines subsumption for *simple* strings; as I introduce more operations on strings, I'll give their semantics in terms of how the non-simple strings they produce subsume other strings. For a rule R to apply to a string W, we require that the structural description T of R subsumes W - in other words, that T can be mapped onto a subpart of W such that each segment in T is compatible with its corresponding segment of W.

4. Operations on strings

We said above that the set of strings is the closure of the set of simple strings under the string operations; in other words, strings are either simple strings or objects you can produce by performing operations on simple strings. This is an example of the Cheapskate's Principle, proposing a minimal amount of machinery but not imposing any limits on that machinery: I propose that strings are closed under concatenation, OR, and perhaps NOT and AND, because there are empirical examples of rules that seem to need these operations.[9]

4.1 Concatenation

The set of strings is closed under concatenation: I take it that there's no longest possible string, and that there are no restrictions (in the representational system) on what symbols can co-occur in a string. For any two strings X and Y, the concatenation XY (all the segments and timing slots of X followed by all the segments and timing slots of Y is a string. To express this fact about linear order in terms of timing slots, we just require the condition, for all timing slots i in X and j in Y, that $i < j$ under the ordering $<$ on I. We don't need to propose a 'string-splicing' operation that would interleave segments of X and Y.

If we just define pairings of timing slots and feature bundles as primitive, then the set of simple strings emerges as the closure of the set of such pairings under concatenation.

4.2. OR

Most often, the set of segments targeted by the structural description of a rule will be a 'natural class', in the sense of being exactly the set of segments defined by their sharing some set of features as subsets. Final devoicing of obstruents targets that set of features with [$-$son] as an element.

In cases where the set of segments targeted is a non-natural class, we need the phonology to have access to an OR operation acting on strings. In terms of subsumption, the semantics of OR are that a string $A = B$ OR C subsumes a string W iff B subsumes W or C subsumes W (including the case where both subsume W).

4.2.1. In structural descriptions

In Evenki (Nedjalkov, 1997; Mielke, 2008), for example, the consonants /v s g/ nasalize after nasals:

/oron-vA/ $\rightarrow$ [oron-mo] 'the reindeer (acc. def.)'
/oron-vi/ $\rightarrow$ [oron-mi] 'one's own reindeer'
/ŋanakin-si/ $\rightarrow$ [ŋanakin-ni] 'your dog'
/oron-gatʃin/ $\rightarrow$ [oron-ŋatʃin] 'like a/the reindeer'

There's no single set of features that /v s g/ share to the exclusion of all other segments in Evenki, and yet they seem to be phonologically active. Having an OR operation means the

[9] We shouldn't think of these as derivational operations, in that (unlike rules) the phonology never has to execute them in real time - they're abstract relations between strings, telling us that if X and Y are strings, then $O(X,Y)$ is also a possible string given some operation O.

phonology can express the fact that a rule applies to /v/ OR /s/ OR /g/. The relevant structural description is a disjunction of three simple strings:

$T = \{(i, /\mathrm{v}/)\}$ OR $\{(i, /\mathrm{s}/)\}$ OR $\{(i, /\mathrm{g}/)\}$

Any classic examples of unnatural classes, like the *ruki* rule retroflexing /s/ before /r u k i/ (Zwicky, 1965), will involve a use of OR in this model. For theories where all the grammar can do is spread a single natural class of features from one segment to another, these rules are problematic.

4.2.2. In environments

We can have the operation OR applying to environments, as well as structural descriptions; this has the same purpose as the brace notation in traditional rule-based phonology.

For example, at one point, Old English had two vowel shortening rules (Kiparsky, 1968):

$\mathrm{V} \rightarrow [-\text{ long}]$ / _ CCC
$\mathrm{V} \rightarrow [-\text{ long}]$ / _ $\mathrm{CCVC_0VC_0\#}$

If we take it that speakers make the generalization that these are the same process, the brace can encode a disjunction of multiple environments in the same rule:

$\mathrm{V} \rightarrow [-\text{ long}]$ / _ CC $\{\mathrm{C}, \mathrm{VC_0VC_0\#}\}$

In String Phonology, this is just another instance of OR: we can informally express the environment for shortening as $\{(i, \mathrm{CC})\}$, concatenated with the string $\{(j, \mathrm{C})\}$ OR $\{(j, \mathrm{CCVC_0VC_0\#})\}$.[10][11]

To want to encode these processes in the grammar as a single rule - nasalization of /v/ along with nasalization of /s/, etc. - we would need to be sure that these aren't just synchronically accidental pairs of rules with the same structural description and structural change. Kiparsky argues in that in the Old English case, the two rules change in the same way in Middle English, suggesting a single change to a complex rule. We use this argument to justify any generalized kind of rule: we propose a rule devoicing final obstruents, not separate rules devoicing /b/, /d/, and /g/. The logic is the same here.

The symbol OR is also the way to express the rules for which traditional generative notation uses parentheses, for defining optional parts of rules. In Karok (Bright, 1957; Kenstowicz and

---

[10] I say 'informally' because this expression informally maps a timing slot to a whole string; the proper representation in String Phonology will be multiple timing slots long.

[11] We need to say the set of strings is closed under concatenation, not just the set of simple strings (as we noted in the last section), because we see here we can concatenate non-simple strings.

Kisseberth, 1979), /s/ palatalizes when preceded by /i/, with an optional intervening consonant:

/ni-skak/ → [ni-ʃkak] ‘I jump'
/ni-ksah/ → [ni-kʃah] ‘I laugh'

In other words, we want palatalization to happen preceded by /i(C)/: i.e. by /i/ or by /iC/. In the terms of String Phonology, the environment is $\{(i, /\mathrm{i}/)\}$ concatenated with the string $\{(j$, [+cons]$)\}$ OR $\varnothing$: either a consonant, or the empty string. (The empty string has no timing slots, so the set of pairs of timing slots with segments is the empty set $\varnothing$.)

By the Cheapskate’s Principle, given we need an operation OR, we should propose OR; but given we have the machinery of OR, we should avoid putting constraints on OR without positive reason to. I say the set of strings is closed under string operations like OR just because it would represent an extra assumption to add any constraints.

4.3. AND and NOT

Phonology also seems to have access to the operations AND and NOT, parallel with OR. The operation AND would take two strings X and Y and produce a third string X AND Y; the semantics of AND are that X AND Y subsumes a string Z iff X subsumes Z and Y subsumes Z. The operation NOT would turn a string X into a string NOT X, where we define NOT X to subsume any string except those subsumed by X (see the short squib from Zwicky, 1970).[12]

Because the set of strings is closed under any string operations, having NOT and AND also gives us composite operations like AND NOT, NOT…OR, etc.[13]

4.3.1. On segments

The neatest use of NOT comes from ‘concave’ (or ‘L-shaped’) classes, where a rule targets some set of segments equivalent to a natural class with a smaller natural class missing.

In some varieties of Palestinian Arabic (McCarthy, 1997; Flemming, 2005), pharyngealization harmony applies to all segments except “high front vowels, palatal glides, and palato-alveolar consonants”; in other words, all segments except those that are [+high, -back]:

---

[12] We have a ‘segmental’ type of AND, in that a feature bundle [+nasal, +voice] is a logical conjunction of two bundles [+nasal] and [+voice]. This is a separate issue from whether there’s an AND operation parallel to OR, applying to whole strings.

[13] We should note that by the usual De Morgan’s Laws in logic, having all three of OR, NOT, and AND is redundant; X AND Y is equivalent to NOT (NOT X OR NOT Y), and dually, X OR Y is equivalent to NOT (NOT X AND NOT Y). I’ll use all three symbols for ease of reading, but if these representations are psychologically real, there is some fact of the matter about which way rules actually tend to be encoded.

/tˤuubak/ $\rightarrow$ [tˤuubɑk] 'your blocks'
/tˤaal/ $\rightarrow$ [tˤɑɑl] 'long (pl.)'
/tˤiinak/ $\rightarrow$ [tˤiinak] 'your mud'
/sˤajjad/ $\rightarrow$ [sˤɑjjad] 'hunter'

Taking pharyngealization to be encoded by a feature [RTR], McCarthy uses a highly-ranked markedness constraint *[RTR, +high, – back] with the effect of specifically banning pharyngealized high back segments. Pharyngealization targets every segment, then, *except* these ones.

The model in this paper has no constraints to block particular outcomes - so intuitively, the right expression of the structural description of pharyngealization is NOT $\{(i,$ [+high, – back]$)\}$. Again, it would be a fallacy to assume that diachrony or phonetic motivation affects how a rule is expressed synchronically; but it's interesting that this rule has a NOT-like origin, rather than an OR-like one as Schaffhausen Swiss German did.

If we want a rule to be blocked by a feature bundle among segments that are themselves specified by some features, we need the power of AND as well as NOT.

In Kinyamwezi (Maganga and Schadeberg 1992, Mielke 2008), high vowels except /ɪ/ are desyllabified before other vowels (with lengthening of the following vowel); all other vowels delete.

/mi-enda/ $\rightarrow$ [mjeenda] 'clothes'
/mu-iβa/ $\rightarrow$ [ŋwiiβa] 'thief'
/kʊ-iβa/ $\rightarrow$ [kwiiβa] 'steal'
/a-lɪ-eŋha/ $\rightarrow$ [aleeŋha] 'he is bringing'

The right structural description for desyllabification *isn't* just NOT $\{(i,$/ɪ/$)\}$, because that would target all other vowels as well; we need 'all high vowels except /ɪ/', which in this formalism is $\{(i,$[+high, – cons]$)\}$ AND NOT $\{(i,$/ɪ/$)\}$. This matches a segment so long as it matches [+high, – cons] and doesn't match /ɪ/.

4.3.2. On strings

In the case of rules involving AND and NOT that target single segments, we could also use OR: trivially, any class of segments can be built up from a union of smaller natural ones. This *isn't* so in the case of strings. Given some string X, there's no way in general to express the infinite set of strings that don't match X as a union of expressible strings - we need the symbol NOT, if there are rules that target everything except certain strings.

One string process that needs to involve AND NOT comes from Lisbon Portuguese (Carvalho, 2011, Spahr, 2016; this analysis is from Andersson, in press). In unstressed syllables, singleton /a/ raises to /ɐ/, but geminate /aa/ shorterns to /a/:

/amiga/ $\rightarrow$ [ɐmigɐ] ‘friend’
/kaza/ $\rightarrow$ [kazɐ] ‘house’
/a amiga/ $\rightarrow$ [amigɐ] ‘the friend’
/kaza azul/ $\rightarrow$ [kazazul] ‘blue house’

The rules are in a counterfeeding order, in that the shortening /aa/ $\rightarrow$ [a] doesn’t feed /a/ $\rightarrow$ [ɐ]; the interesting rule here is the first one, which needs to target /a/ without targeting /aa/. We said earlier that geminate vowels were just represented as a sequence of two timing slots $\{(i_1,\text{/a/}),(i_2,\text{/a/})\}$, which ought to match the structural description of a rule targeting $\{(i,\text{a})\}$.

In String Phonology, the rule targets the strings $\{(i,\text{/a/})\}$ AND NOT $\{(i_1,\text{/a/}),(i_2,\text{/a/})\}$; in other words, strings that match /a/, but don’t match /aa/.[14]

4.4. The space symbol

We said earlier that representations can also contain an ‘anything goes’ symbol, the ellipsis ...; the reason for this is that the phonology sometimes needs to refer to non-contiguous strings as parts of rules.

In Coeur d'Alene (Fitzgerald, 1997), a stressed vowel will lower if a pharyngeal or uvular consonant follows anywhere in the word, not necessarily immediately afterwards:

/spumalqs/ $\rightarrow$ [spomalqs] ‘fur coat'
/stʕewʕəwpus/ $\rightarrow$ [stʕawʕəwpus] ‘tear drop’

Supposing pharyngeals and uvulars are unified to the exclusion of other segments by being [– high, +back, +cons], the left-hand environment of the lowering rule will need to encode the fact that a vowel lowers if a [– high, +back, +cons] bundle occurs anywhere after it. The space symbol means we can express this string:

$\{(i_1,...),(i_2,[-\text{high, +back, +cons}])\}$

Under the ordering on timing slots, $i_1 < i_2$, so this expresses the fact that the pharyngeal or uvular appears to the right of the optional material. The semantics of ... are that the pair $\{(i,...)\}$ subsumes *any* string.

Strings containing the space symbol can be input to string operations. The Sanskrit *ruki* retroflexion mentioned earlier was blocked by an /r/ anywhere later in the word: in other

[14] These examples all involve AND used together with NOT; I don’t know of any rules in any language that use AND on its own, where a string needs to satisfy two properties not expressible using a single target string. An example would be ‘if the string contains a [+high] segment and a [-back] segment (which may or may not be the same segment), add an epenthetic schwa’. Any absence of these rules wouldn’t be worrying, though, because AND comes from free anyway by De Morgan’s Laws given that we have OR and NOT.

words, we say the rule has the right-hand environment $\{(i_1,/\text{r}/)\}$ OR $\{(i_1,/\text{u}/)\}$ OR $\{(i_1,/\text{k}/)\}$ OR $\{(i_1,/\text{i}/)\}$ AND NOT $\{(i_1,...),(i_2,/\text{r}/)\}$.

4.5. Variables

The final types of symbol that appear in phonological representations are the variable symbols, of the form [α F], [βF], [γ F] for each feature F. The semantics are the usual: the second use of a variable (in the structural change) is defined to be coreferential with the first use (in the structural description or the environment). For subsumption purposes, variables are treated as as disjunctions over the possible values of a feature. In other words, [α F] matches any value of F, or even an absence of a specification for F.

The reasoning is that the phonology has the power to key parts of the output to parts of the input - what the structural change involves can depend on what the structural description and environment are.

As in traditional generative phonology, 'alpha' rules are needed to capture generalizations like place assimilation, where the segment produced by the rule is a function of the environment. In lots of languages, for example, /n/ assimilates in place to the following obstruent; in String Phonology, the rule would have right-hand environment E_R and structural change C_2 as follows:[15]

$$E_R = \{(i,\{[+\text{cons}, -\text{son}, \alpha\,\text{place}]\})\}$$
$$C_2 = \{(i,\{[\alpha\,\text{place}]\})\}$$

In other words, nasals take on the same place, [α place], as the consonants that follow them.

5. How rules act on strings

We've said that a rule can act on a string iff the structural description subsumes the target of the string and the left and right environments subsume the strings to the left and right of the target, respectively.

How a rule then acts on the string is governed by the structural change, a pair of strings $C = (C_1, C_2)$. So far, I've only mentioned static facts about representations and conditions for rule application: in this model, what the phonology actually *does* consists of a two-step operation. Following Poser (2004) and Bale, Papillon and Reiss (2014), I take it rules use a 'delete-and-unify' process. First, we delete the string C_1 from the target string; then we unify the target string with C_2.

[15] See section 5 for this notation:

This is the only reason I've chosen to separate out the structural description from the environment, rather than each rule applying a single string E_LTE_R; where T starts determines *where* in the target string the structural change C applies to. In principle, we could copy all the environment specifications into the structural change and match them to their counterparts in the target string using variables as indices, so mapping the whole string E_LTE_R into $E_LC(T)E_R$. In that case, we'd treat terms like ‘structural description’ and ‘environment’ as practically convenient labels, rather than as actual parts of the model.

5.1. Delete-and-unify

5.1.1. Poser’s Chumash sibilant harmony

The tradition of treating rules as consisting of deletion followed by unification started with Poser’s (1982, 1993) discussion of harmony in Chumash.

In Chumash, we have a right-to-left harmony rule that spreads the feature [ant] rightwards from sibilant to sibilant (data simplified from Poser’s original):

/hasxintilawaʃ/ $\rightarrow$ [haʃxintilawaʃ] ‘his former gentile'
/siʃhuleqpeyus/ $\rightarrow$ [sishuleqpeyus] ‘they two want to follow it'

As well as Sibilant Harmony, sibilants are affected by a rule of Pre-Coronal Palatalization (PCP), which patalizes /s/ into /ʃ/ before non-strident coronals /t l n/:

/snanʔ/$\rightarrow$ [ʃnanʔ] ‘he goes'
/stepuʔ/ $\rightarrow$ [ʃtepuʔ] ‘he gambles'

When /ʃ/ is produced by PCP, it feeds Sibilant Harmony:

/sislusisin/ $\rightarrow$ [siʃlusisin] $\rightarrow$ [ʃiʃlusisin]

But instances of /ʃ/ produced by PCP don’t themselves undergo Sibilant Harmony:

/stiyepus/ $\rightarrow$ [ʃtiyepus] (but not $\rightarrow$ [stiyepus] again)

We have an ordering paradox: PCP needs to precede Sibilant Harmony to feed it, but PCP also seems to counterfeed Sibilant Harmony, which would require following it. Poser proposes that what looks like a single process of harmony is in fact two stages - deletion of the [ant] feature on all non-rightmost sibilants, and then spreading of the [ant] feature from the rightmost one - rather than a single feature-changing process. This explains how PCP can intervene:

1. Sibilant Harmony 1: deletion
2. PCP
3. Sibilant Harmony 2: spreading

PCP counterfeeds the first stage, in that sibilants produced by PCP don't undergo deletion of [ant], but it feeds the second stage, in that [ant] spreads from sibilants produced by PCP. Poser hints that *all* feature-changing rules should be analysed using these two stages, given processes of deletion and spreading are shown to exist in this case; String Phonology is one of several models to take up this suggestion.

5.1.2. Bale, Papillon and Reiss (2014)

Bale, Papillon and Reiss (2014; 'BPR') take up Poser's suggestion that all feature-changing rules should be analysed as feature deletion followed by feature-filling rules, giving an argument from Turkish to justify the existence of the latter.

In Turkish (Inkelas and Orgun, 1995), there are three sorts of stop series:

Non-alternating voiceless:
  [sanat] 'art', [sanatım] '1sg.poss.'
Non-alternating voiced:
  [etyd] 'etude', [etydym] '1sg.poss.'
Alternating:
  [kanat] 'wing', [kanadım] '1sg.poss.'

In the third type, we have a voiceless stop finally, but a voiced one medially. We can't explain the alternation with a rule of final devoicing - because [etyd] doesn't devoice - nor a rule of medial voicing - because [sanat] doesn't voice. Inkelas proposes that in the alternating type, the stop is underlyingly /D/, a coronal stop unspecified for [voice] - and that the values for [voice] are then filled in by context-sensitive feature-filling rules. Effectively, having an underlying difference between /D/, /t/, and /d/ gives us a three way opposition.

Here, we have the second half of the 'delete-and-unify' process on its own; unifying without deleting anything, which means the rule only applies to underspecified segments. We take it that the union of a value of [voice] with another value of voice is undefined, so these context-sensitive rules don't apply to specified segments.

BPR also give a phonetic argument from Korean that we can have the first half of the process, 'delete', without a 'unify' part. All coronal obstruents in Korean (Kim, 1979) neutralize word-finally to a coronal stop [t] with some contextual variability, which BPR analyse as a case of 'derived underspecification': a phonological rule deletes their distinguishing features, but fails to unify them with any further feature bundles.

So we need both feature-filling and feature-deletion rules. The BPR claim, following Poser, is that there's in fact only one sort of rule in the grammar - which, all else being equal, we should want. We don't have any evidence that we should give the phonology a separate feature-changing operation on top of these two.

The BPR model is successful at explaining the mechanism of the rules it explains - but it doesn't have a notion of a string, and this means it can only account for rules that map single segments to single segments. Because it deletes and unifies sets of features, it has no way of expressing any rules that might change the number of segments or move segments around. That's why we need the string-based extension outlined in this paper.

5.2. Examples

Here, I give some explicit examples of deletion and unification in practice.[16]

5.2.1. Step 1: deletion

It's possible to (vacuously) 'delete' a feature that isn't there - deleting the feature [+voice] from the set $\{[+\text{ATR}],[-\text{high}]\}$ can be defined to just give the same set back, for example - so rather than requiring that the deleted material C_1 subsumes the target string W, we just require that it can be mapped onto it in a way satisfying the second and third points under the definition of subsumption above.

To model deletion of segments, we specify that some *timing slots* from C_1 are part of the material to be deleted, and then have some convention that a segment in the surface representation without a timing slot isn't pronounced.

If a rule applies to a target string of length 3 and deletes [$-$voice] from the first segment, deletes [+high] and [$-$back] from the second one, and deletes the entire third segment, its C_1 would look like this:

$$C_1 = \{(i_1,\{[\text{-voice}]\}),(i_2,\{[+\text{high}],[-\text{back}]\}),(i_3,\varnothing)\}$$

The deletion part of the rule will map i_1, i_2, and i_3 onto the target string in a way that preserves the ordering under $<$ on I, and then deletes [$-$voice] from the segment at the image of i_1, [+high] and [$-$back] from the segment at the image of i_2, and then the whole timing slot at the image of i_3.

It then doesn't matter what features are or aren't deleted from the feature bundle at the image of i_3, because those will never be pronounced, so in this formulation I've said none of them are deleted. I also take it we delete the image of i_3 from the set of timing slots, so it doesn't interfere with any future subsumption relations by counting as an intervening segment.

[16] I haven't specified a *direction* for these rules to apply in: it's standardly assumed (see Kenstowicz and Kisseberth, 1979) that rules can apply iteratively, and either left-to-right or right-to-left. We suppose that each rule is specified in the grammar with a direction to apply in.

In sum, given a target string $\{(j_1,S_1),(j_2,S_2),(j_3,S_3)\}$, deleting $\{(i_1,[-\text{voice}]\}),(i_2,\{[+\text{high}],[-\text{back}]\}),(i_3,\varnothing)\}$ gives a new string:[17]

$$(j_1,S_1-\{[-\text{voice}]\}),(j_2,S_2-\{[+\text{high}],[-\text{back}]\}),(\varnothing,S_3)\}$$

5.2.2. Step 2: unification

The second part of the rule involves 'unifying' the string C_2 with the target string W, once the material from C_1 has all been deleted. As with the mapping from C_1 to W, we map the timing slots of C_2 by some function U onto the timing slots of the new W - and at each timing slot, we take the union of each set of feature values at each i in C_2 with its corresponding set at each $U(i)$ in W.

If a rule unifies takes a string of length three and unifies $\{[-\text{ATR}]\}$ with the first slot, nothing with the second, and $\{[-\text{voice}],[+\text{ATR}]\}$ with the third, its C_2 looks like this:

$$C_2=\{(i_1,\{[-\text{ATR}]\}),(i_2,\varnothing),(i_3,\{[-\text{voice}],[+\text{ATR}]\})\}$$

If we unified this with a string $\{(j_1,S_1),(j_2,S_2),(j_3,S_3)\}$, the output of the rule would be the string:[18]

$$\{(j_1,S_1\cup\{[-\text{ATR}]\}),(j_2,S_2),(j_3,S_3\cup\{[-\text{voice}],[+\text{ATR}]\})\}$$

5.2.3. 'Autosegmental' phenomena

String Phonology has no separate 'tiers' of representation, unlike autosegmental phonology (ASP; Goldsmith, 1976), so it might seem as if autosegmental phenomena are difficult to account for. The main tenets of ASP are that all phonological operations can be reduced to spreading and delinking of association lines between various tiers, and these association lines are subject to the No Crossing Constraint (NCC): lines can't cross. This is claimed to be useful for representing 'autonomous' behaviour of features like tone, in that they associated from their own tier separate from the melody.

---

[17] The minus sign - is set subtraction in the normal sense.

[18] Sometimes U might produce a timing slot with a set containing two values from the same feature. Depending on preference, we might say:

- that this is a valid segment that happens not to be pronounceable at the interface with the articulators;
- or that this bundle of features isn't a 'segment', by stipulating segments don't contain multiple values of the same feature (as we did earlier);
- or just that the output of the unification operation is undefined in those cases where it would produce a segment with multiple values from the same feature (as BPR assume for Inkelas' Turkish example).

This architecture is far too restrictive, as we've seen - in trying to bias the grammar towards 'natural' single-segment-to-single-segment spreading, which fails to account for unnatural rules and rules that act on strings - so ASP isn't a good solution. Given the existence of rules that clearly don't just involve spreading, we have less motivation for giving a special status to rules that happen to look like they do.

As (what looks like) spreading is just a special case of string-string mapping, String Phonology can easily account for the facts that ASP was designed to account for. Take the classic example of Mende tones (Leben, 1973), where adding the underlyingly toneless ending *-ma* 'on' is taken to cause spreading of the preceding tone of the pattern:

Tonal pattern	Tonal pattern when added to *-ma*
H	HH
HL	HL
LH	LH
HH	HHH
LL	LLL
HL	HLL
LHL	LHL

In this model, the relevant environment for unifying the toneless ending with a high tone can be expressed with the space symbol and the AND NOT operations:

$$E_L = \{(i_1,\{[\,\mathrm{H}\,]\}),(i_2,...)\} \text{ AND NOT } \{(i_1,\{[\,\mathrm{H}\,]\}),(i_2,...),(i_3,\{[\,\mathrm{L}\,]\}),(i_4,...)\}$$

And the vice versa case for L being blocked by an intervening H. In other words, spreading of H happens in the environment H … _, but not in the environment L … _. This model doesn't make spreading look 'natural', in that the NCC has no formal status in the grammar and has to be encoded in the rule. But given the clear diachronic origins of the NCC in languages - phonetic coarticulation happens with neighbouring segments, not distant ones - it would be a mistake to duplicate our explanations by encoding it in the architecture of the grammar.

6. Generative power: what's in and what's out

We've now seen String Phonology in action: a schema for phonological grammars designed to be a maximally general model for mapping strings to strings, accounting for attested rules, but not biasing the computation towards 'natural' rules or rules that act on single segments.

In this section, I list some types of rules that this model predicts to be computationally impossible, and some more types it predicts to be possible - though as I commented in the first section, it’s not a grammatical model in itself that makes predictions about what patterns we should see, but a theory of language change and acquisition.

6.1. Some impossible rule types

6.1.1. Bypassing rules

BPR note that their model, based on subsumption, can’t generate ‘by-passing’ rules, where specified segments can move ‘past’ underspecified segments. In a model with both feature-changing and feature-filling rules, we could have a derivation as follows (their example):

1. $[+\text{cor}, -\text{ cont}, +\text{ voice}] \rightarrow [-\text{ voice}]\ /\ \_\ ]_{\sigma}$
2. $[+\text{cor}, -\text{ cont}] \cup [+\text{ voice}]\ /\ \_\ ]_{\sigma}$

This would map an underlying /d/ to [t] in a coda by changing its [+voice] feature directly to [– voice], and then map underlying /D/ to [d] by filling in a [+voice] feature; so /d/ ‘bypasses’ the underspecified /D/ in the middle. In a delete-and-unify model, fully specified /d/ passes through a stage of being identical to /D/, after its [+voice] feature is deleted and before its new [– voice] feature is unified; so it’s impossible for a single rule to affect /d/ but not /D/.

6.1.2. Targeting underspecified segments

In a subsumption-based theory, if /X/ contains a subset of the features of /Y/, it shouldn’t be possible for a feature-changing rule - i.e. a rule with both deletion and unification parts - to target X but not Y. Any structural description or environment Z that subsumes X will also subsume Y, in that there can’t be any features in both X and Z but not Y. This makes a prediction: it shouldn’t be possible for a rule to target only underspecified segments, in either its structural description or its environment. The only rules that should be able to are feature-filling rules, which specifically fill in the feature that the underspecified segments are underspecified for.

As McCarthy and Taub (1992) point out, this view of rule application is problematic for analyses of coronals as underspecified for place (e.g. in Mascaro’s 1976 analysis of Catalan), given the existence of phonological rules that target classes that need to be expressed as ‘coronal'. Tapping in AmE targets /t d/; affrication before /r/ also targets /t d/; glottalling in Cockney English targets /t/; etc.

6.1.3. Exchange rules

BPR also explain that a model based on a delete-and-unify process can’t generate ‘polarity rules', or ‘exchange rules', as single rules. In a model that allows alpha variables to invert the value of a feature, a single rule could map /t/ to [d] and /d/ to [t]:

$[+\text{cor}, -\text{ cont}, \alpha\text{ voice}] \rightarrow [-\alpha\text{ voice}]$

Taking [– (– voice)] to be equivalent to [+voice], this gets us a rule mapping [+voice] to [– voice] and [– voice] to [+voice]. As BPR put it, as with bypassing rules, the fact that all rules involve deletion means that /t/ and /d/ necessarily merge as /D/ in any rule that affects the voicing on both of them; so the feature-filling rules that happen afterwards have no memory which were underlying /t/ and which were /d/.

Fitzpatrick et al. (2004) give three candidate examples of exchange rules. One, from Spanish, is just a morphological 'rule' - and so needn't be computable by the phonological component. The second is from Zok, but (as they acknowledge) a gap in the data means that not all parts of the exchange rule are there. Their third example is from Brussels Flemish. When underlying non-low back vowels appear in closed syllables, they shorten and invert in height:

/sxu:mtə/ → [sxomtə] 'shame'
/vlu:ms/ → [vloms] 'Flemish'
/vo:tʃə/ → [vutʃə] 'little foot'
/mo:t/ → [mut] 'must'

They say this involves an exchange rule [α high] → [$-\alpha$ high]. But the fact that the vowels also shorten is crucial: it means that a model that can map strings to strings can account for the shortening and height change in one go. We could just have a single rule mapping /u:/ → [o] in the relevant environment, and another rule mapping /o:/ → [u], with no crossover. This doesn't involve an exchange rule.[19]

6.1.4. Alpha rules

In Fitzpatrick et al.'s (2004) computational hierarchy of 'alpha' rules, exchange rules are 'Type I':

Type	Example	Computational requirements
Type I	α F → $-\alpha$ F	negation
Type II	C → $-\alpha$ F / _ α F	limited-scope feature-value variables
Type III	C → $-\alpha$ F / _ α G	unlimited-scope feature-value variables

This model can't get either Type II or Type II processes in a single rule.

[19] In some of these cases, as BPR point out, it's strictly possible to generate the correct input-output mappings if we let multiple rules do some 'juggling' (Sayeed, 2016). Even in the case of apparent exchange rules, we could map /t/ to some other segment /X/, map /d/ to /t/, and then finally map /X/ to /d/. If a child has a grammar that works by our model, they'd be *forced* to propose a derivation like this during acquisition to account for the facts in the input.

One place where Type II processes are claimed to exist is in tone polarity, as in Kɔnni (Cahill, 2004), where an underlyingly toneless syllable takes on the opposite tone to the one neighbouring it. In this model, we have to treat these as separate rules, rather than a single rule $[\alpha \text{H}] \rightarrow [-\alpha \text{H}]$. This isn't necessarily problematic. Given that there are only two logically possible tonal rules with two tones (H $\rightarrow$ L and L $\rightarrow$ H), any grammar will contain one, both, or neither; in the situation where a grammar happens to contain both at once, it looks like tone polarity. Tone polarity rules are context-sensitive feature-filling, not feature-changing, so we don't have the same problem as with exchange rules: a tone can map to L in the environment H _, and to H in the environment L _, with no worry about bypassing or information loss.

The only example Fitzpatrick et al. give of Type III processes in natural language is from the Proto-Indo-European stop inventory (Fortson, 2004). Out of the three stop series **T*, **D*, and **D^h^*, all combinations exist in PIE roots except **TeD^h^*, **DeD*, and **D^h^eT*. But this example is from the static phonotactics of PIE, which in general will be the output of some historical change - and not encoded inside the structural description, environment, or structural change of any synchronic rule of the grammar. It's an interesting generalization, but not one that needs to be represented by our system. This type of process also depends on a potentially problematic commitment to the idea that we can identify the '+' end of one feature with the '+' end of another. It represents an extra assumption about the grammar to say that our informal descriptions of features as having names like 'back' (not 'front') or 'voice' (not 'lack of voice') correspond to real features of the grammar.

6.1.5. Mirror-image rules

Mirror-image rules are 'adirectional', in that the environment of the rule refers to the segments *adjacent to* (rather than to the left or right of) the target. Kenstowicz and Kisseberth (1979) give an example from Lithuanian, where both /ʃ+s/ and /s+ʃ/ surface as [ʃ]; the claim is that Lithuanian has a rule deleting /s/ adjacent to /ʃ/. Again, String Phonology is forced to propose separate rules: the representational machinery for generating mirror-image rules doesn't follow from anything else in the grammar.

6.2. Other impossible rules

Looking beyond more obvious rule types, String Phonology definitely isn't 'anything goes': there are all kinds of other imaginable rules that aren't possible:

- 'Reverse the segments of every word.'
- 'Add consonants to the end of the word until it has the same number as the beginning of the word.'
- 'If the first /b/ is closer to the end of the word than the beginning, devoice it to /p/.'
- 'Change every vowel to the vowel that appears most often in the underlying form.' (In other words, 'majority rules' vowel harmony.)
- 'If the word contains a number of /q/s that's divisible by seven, delete the final consonant.'

- ‘If there are no clusters with an odd number of consonants, voice the last segment of the word.’

Majority rules vowel harmony is claimed to be attested in Warlpiri (Bowler, 2013), which this model predicts should turn out to be analysed differently, along with those advocates of Optimality Theory who'd prefer OT to stay computable by a finite-state automaton (Bowman, 2011).

We rule these out on the basis of an independently justified model of rule application - because the minimum of machinery demanded by the Cheapskate’s Principle can’t generate them - rather than for reasons of computational power, naturalness, or any other considerations.

6.3. Some possible rule types

String Phonology does, however, allow these equally silly-looking rules:

- ‘If there is a cluster with five consonants, voice the last segment of the word.’

An environment specification can contain five [+cons] features anchored to timing slots on one side and a word boundary on the other, while the structural change voices the segment.

- ‘Move any word-initial fricative and add it to the end of the word.’

Given a string of the form [+cons, +cont, – son] X - in other words, a fricative concatenated with any string - map it to X [+cons, +cont, – son].

- ‘Count the number n of instances of /q/ in the word, and remove all but the last n mod 7.’

Given a rule that deletes a sequence of seven /q/s, applying iteratively from either left to right, we’ll end up with n mod 7 instances of /q/ left over.

- ‘If a word contains a labial and a velar, switch their place specifications so that the labial comes first.’

The environment specification can be of the form [labial] ... [dorsal] OR [dorsal] ... [labial], targeting strings containing both a labial and a velar; it maps both of these schemas to [labial] ... [dorsal].

- ‘If every labial comes before every velar, add an epenthetic schwa to the end of the word.’

Stating that every labial comes before every velar is equivalent to stating that no velar comes before any labial; in other words, this rule should accept exactly the complement of strings in which the latter happens, so its environment specification should be of the form NOT [dorsal] ... [labial].

- 'Reverse the order of the segments if the word is less than seven segments long.'

The grammar has no way to represent the reversal of a general string, which is why 'reverse the order of the segments in every word' is ruled out; but reversing the order of a given number of segments is possible, just by using variable symbols to 'manually' map α β ... ω to ω ... β α.[20]

We aren't limiting the model to only be able to compute rules that look (intuitively) 'sensible'; we're making a principled decision to propose the least machinery possible, but to put the fewest constraints on that machinery. By the Cheapskate's Principle, we take it that phonology can do anything using the machinery given to it.

Two of these are in fact real rules, attested in child language acquisition. Ringe and Eska (2013) comment anecdotally, in explaining the apparently sporadic long-distance place metathesis in the Tocharian word for 'tongue': "*At about the age of 2 a daughter of one of the authors evolved a rule that if a word contained both a bilabial and a velar articulation, the bilabial must precede; laryngeal features, however, were not affected, so that 'camel' (for instance) was pronounced* [pʰaŋi] *and 'grape' was pronounced* [breɪk]." And Leonard and McGregor (1991) report that a child they call 'W' innovated a rule moving word-initial fricatives to the end of the word. The word 'fine' came out as [ainf], 'zoo' as [uz], 'snow' as [nos], 'spoon' as [buns], and 'Snoopy' as [nupis]. (We see the same kinds of rules in language games, e.g. Pig Latin.)

If the rules we see in the languages of the world are 'natural' for diachronic reasons, then rules with no diachronic origin - rules innovated by children during acquisition, or in language games like Pig Latin - should show a wider range; and it does seem to be true, that 'crazy' rules surface more often in acquisition than in adult language. The message from these acquisition cases backs up the substance-free/evolutionary point; it's a mistake to limit the phonology to more common or natural rules. The fact that String Phonology predicts all these silly-looking processes isn't a disadvantage of the model.

7. Conclusion

If we take seriously the idea that phonological theories describe mental systems, then it's a category mistake to try to bias those theories towards types of rules that are common, or even attested. Without good reason to depart from the null hypothesis that the phonological component can compute anything given the machinery we know it needs, we should want a quite general theory of phonology. I've presented String Phonology, an attempt at giving phonology minimal machinery, while putting no constraints on that machinery. Explanations about common and uncommon rules will come from understanding diachrony, not from stipulating biases in the system.

---

[20] Obviously, this'd make for an unwieldy grammar, and we might even think no learner would ever postulate all of these... but it's within the power of the system.

Thanks to Samuel Andersson, Jim Blevins, Elliot Ford, Maxime Papillon, Bridget Samuels, Shanti Ulfsbjorninn, and Bert Vaux for helpful comments and discussion on all of these issues, and to Daniel Heydecker and Adam P. Goucher for mathematical comments.

References

Albright, Adam, Argelia Andrade, and Bruce Hayes. 'Segmental environments of Spanish diphthongization.' UCLA Working Papers in Linguistics 7 (2001): 117-151.
Alrasheedi, Eisa S. 'Affrication in Ha'ili Arabic: A rule-based approach.' International Journal of Linguistics 7.4 (2015): 27-41.
Anderson, Stephen R. 'Why phonology isn't natural.' Linguistic inquiry (1981): 493-539.
Andersson, Samuel. 'Sieves and Herrings: For Distinctive Vowel Length in Swedish.' In press.
Andersson, Samuel. 'Safety in Numbers: Geminates in Lisbon Portuguese'. In press.
Bale, Alan, Maxime Papillon, and Charles Reiss. 'Targeting underspecified segments: A formal analysis of feature-changing and feature-filling rules.' Lingua 148 (2014): 240-253.
Blevins, Juliette. Evolutionary phonology: The emergence of sound patterns. Cambridge University Press, 2004.
Bowler, Margit. 'Majority Rules effects in Warlpiri vowel harmony', Central Australian Linguistic Circle (CALC), UCLA. 2013.
Bowman, Samuel R. Vowel Harmony, Opacity, and Finite-State OT. Diss. The University of Chicago, 2011.
Bright, William. The Karok language. University of California Publications in Linguistics, 13. Berkeley and Los Angeles: University of California Press. 1957.
Buckley, Eugene. 'Metathesis.' The Blackwell companion to phonology 3 (2011): 1380-1470.
Cahill, Michael. 'Tone polarity in Kɔnni nouns.' Studies in African Linguistics 33.1. 2004.
Carvalho, Joaquim Brandão de. Contrastive hierarchies, privative features, and Portuguese vowels. Revista de estudos linguisticos da Universidade do Porto 6:51–66. 2011.\
Chomsky, Noam. 'Three models for the description of language.' IRE Transactions on information theory 2.3 (1956): 113-124.
Chomsky, Noam. The Generative Enterprise Revisited: Discussions with Riny Huybregts, Henk van Riemsdijk, Naoki Fukui and Mihoko Zushi. Walter de Gruyter, 2004.
Chomsky, Noam and Halle, Morris. The Sound Pattern of English. Harper & Row, New York. 1968.
Church, Alonzo. 'A formulation of the simple theory of types.' The journal of symbolic logic 5.02 (1940): 56-68.
Clements, G. N.. 'The Geometry of Phonological Features.' Phonology Yearbook 2, 225-252. 1985.
Donegan, Patricia J., and David Stampe. 'The study of natural phonology.' Current Approaches to Phonological Theory (Conference on the Differentiation of Current Phonological Theories, Bloomington, Sept. 30-Oct. 1, 1977), ed. Daniel A. Dinnsen, pp. 126-173. Bloomington: Indiana University Press. 1979.
Fitzgerald, Susan. 'Historical aspects of Coeur d'Alene harmony.' International journal of American linguistics 63.3 (1997): 362-384.
Fitzpatrick, Justin, Andrew Nevins, and Bert Vaux. 'Exchange rules and feature-value variables.' 3rd North American Phonology Conference, Concordia University, Montréal, Québec. 2004.
Flemming, Edward. Deriving natural classes in phonology. Lingua 115: 287–309. 2005.
Fortson, Benjamin. Indo-European language and culture: An introduction. Oxford, 2004.
Gnanadesikan, Amalia Elisabeth. Phonology with ternary scales. Diss. University of Massachusetts Amherst, 1997.
Goldsmith, John. Autosegmental phonology. Unpublished Ph.D. dissertation, Massachusetts Institute of Technology. 1976.
Gordon, Matthew, and Pamela Munro. 'A Phonetic Study of Final Vowel Lengthening in Chickasaw.' International Journal of American Linguistics 73.3 (2007): 293-330.
Hale, Mark, and Charles Reiss. The phonological enterprise. Oxford University Press, 2008.
Halle, Morris. 'Feature geometry and feature spreading.' Linguistic inquiry(1995): 1-46.
Hopcroft, John E., Rajeev Motwani, and Jeffrey D. Ullman. 'Automata theory, languages, and computation.' International Edition 24. 2006.
Hyman, Larry M., and Francis X. Katamba. 'Final vowel shortening in Luganda.' Studies in African Linguistics 21.1 (1990): 1.
Inkelas, S. ,Orgun, C.O. Level ordering and economy in the lexical phonology of Turkish. Language71(4), 763--793. 1995.
Johnson, C. Douglas. Formal Aspects of Phonological Description. Mouton. 1972.

Joshi, Aravind Krishna. ‘Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions?.’ (1985): 139159.
Kaplan, R.M. Three seductions of computational psycholinguistics. Linguistic Theory and Computer Applications 149–188. 1987.
Kaplan, Ronald M., and Martin Kay. ‘Regular models of phonological rule systems.’ Computational linguistics 20.3 (1994): 331-378.
Kenstowicz, Michael, and Charles Kisseberth. Generative phonology: Description and theory. Academic Press, 1979.
Kim, C.W. Neutralization in Korean revisited. Stud. Linguist. Sci.9,147--155. 1979
Kiparsky, Paul. Linguistic universals and linguistic change. In E. Bach and R. Harms (Eds.), Universals in linguistic theory. New York: Holt. 1968.
Koskenniemi, Kimmo. ‘Two-level morphology: A general computational
model for word-form recognition and production.’ Publication No. 11,
Department of General Linguistics, University of Helsinki. 1983.
Lass, Roger. Old English: a historical linguistic companion. Cambridge University Press, 1994.
Leben, William. Suprasegmental phonology. Unpublished Ph.D. dissertation, Massachusetts Institute of Technology. 1973.
Leonard, Laurence B. & Karla K. McGregor. Unusual Phonological Patterns and their Underlying Representations: A Case Study.
Journal of Child Language 18, 261-271, 1991.
Maganga, Clement, and Thilo C. Schadeberg. Kinyamwezi: Grammar, Texts, Vocabulary. Cologne: Ko¨ppe. 1992.
Mascaro J. Catalan phonology and the phonological cycle. Ph.D. dissertation, MIT; 1976.
McCarthy, J.J.. Process-specific constraints in Optimality Theory.
Linguistic Inquiry 28, 231-251. 1997.
McCarthy J., Taub A. Review of Paradis & Prunet 1991. Phonology (1992); 9.2:363-370.
Mielke, Jeff. The emergence of distinctive features. Oxford University Press, 2008.
Napoli, Donna Jo, and Marina Nespor. ‘The syntax of word-initial consonant gemination in Italian.’ Language (1979): 812-841.
Nedjalkov, Igor. Evenki. New York: Routledge. 1997.
Nevins, Andrew. Locality in vowel harmony. Vol. 55. Mit Press, 2010.
Newmeyer, Frederick J. Possible and probable languages: A generative perspective on linguistic typology. Oxford University Press on Demand, 2005.
Ohala, John J. ‘Phonetic explanations for sound patterns.’ A figure of speech: A festschrift for John Laver (2005): 23.
Poser, William J. Phonological representation and action-at-a-distance. in H. van der Hulst & N.R. Smith (eds.) The Structure of Phonological Representations. Dordrecht: Foris. 121-158. 1982.
Poser, William J. Are Strict Cycle effects derivable? in Ellen Kaisse & Sharon Hargus (eds.) Studies in Lexical Phonology. San Diego: Academic Press. 315-321. 1993.
Poser, William J. ‘On the status of Chumash sibilant harmony.’ Ms., University of Pennsylvania, Philadelphia. 2004.
Prince, Alan and Smolensky, Paul. Optimality Theory: constraint interaction in generative grammar (Technical Report). Rutgers Center for Cognitive Science, New Brunswick, NJ. 1993.
Raimy, Eric. The phonology and morphology of reduplication. Vol. 52. Walter de Gruyter, 2000.
Ringe, Don, and Joseph F. Eska. Historical linguistics: toward a twenty-first century reintegration. Cambridge University Press, 2013.
Samuels, Bridget D. The structure of phonological theory. Diss. Harvard University Cambridge, Massachusetts, 2009.F
Sayeed, Ollie. ‘Phonological time travel', paper at LAGB Conference, University of York, September 2016.
Shieber, Stuart M. ‘Evidence against the context-freeness of natural language.’ The Formal complexity of natural language. Springer Netherlands, (1985). 320-334.
Silva, David James. "The variable deletion of unstressed vowels in Faialense Portuguese." Language Variation and Change 9.03 (1997): 295-308.
Spahr, Christopher. ‘Confronting the European Portuguese central vowel distinction.’ The Canadian Journal of Linguistics/La revue canadienne de linguistique 61.2 (2016): 211-217.
Wells, John C. Accents of English. Cambridge University Press, 1982.
Zwicky, Arnold M. Topics in Sanskrit phonology. Diss. Massachusetts Institute of Technology, 1965.
Zwicky, Arnold M. ‘Class complements in phonology.’ Linguistic inquiry 1.2 (1970): 262-264.