The Computational Power of Harmony

Alëna Aksënova, Jonathan Rawski, Thomas Graf, Jeffrey Heinz

July 1, 2020

1 Computation and Phonology

Some of the most stunning scientific advances of the twentieth century came from the theory of computation. It is hard to think of an area in science which has not been enriched by studying a subject computationally. At the core of the theory of computation is the ability to formally define problems and processes, and ask how (or whether) they can be solved computationally, and if so, with which resources. For linguistics, the science of language, this means characterizing the cognitive processes involved in the human capacity for language. For phonology, it means asking about the nature of the phonological representations, constraints, and processes present in the phonological grammar.

For phonologists, the big questions are: what constitutes knowledge of phonological well-formedness? What representations are used to determine that knowledge? How is that knowledge acquired on sparse and under-determined data? Each of these questions forms a computational problem. What type of functions constitute phonotactics and phonological processes? What data structures are manipulated in these processes? Is there a learning algorithm that manipulates such structures to correctly and reliably induce these functions?

This chapter seeks to define necessary and sufficient conditions for each of these problems in phonology, specifically involving harmony. A function is computable if there exists a step-by-step procedure — an algorithm — that produces the correct output of the function for any given input. In linguistics, grammars can be understood to be such algorithms.

In this chapter, we distinguish two types of grammars. In the simplest case the input is a linguistic structure, such as a sentence or a string of speech segments, and the output of the grammar is a simple yes/no answer, which corresponds to the familiar notion of categorical well-formedness. (However, nothing hinges on the categorical nature of the function, since we may switch the binary output to something more gradient.) With respect to phonology, phonotactic grammars are an example this kind of grammar because they evaluate the well-formedness of surface forms. Alternatively, phonological grammars, which map underlying forms to surface forms, are another type of algorithm. Here instead of outputting some scalar value to indicate the well-formedness of the input, the grammar/algorithm outputs another linguistic structure.

Consequently, this chapter studies vowel harmony from these two computational perspectives. The first studies the conditions on the types of constraints present in vowel harmony *phonotactics*. The second studies the conditions on the types of constraints on harmony *as a process*. In each case we care about necessary and sufficient conditions on the appropriate class of grammars. From a computational perspective, while there are many similarities between functions which output well-formedness values and functions which output structures, there are also some key differences.

The takeaway is that in both the cases, the computational complexity of vowel harmony appears to fall within a well-defined class of finite-state, or regular, models (Johnson, 1972; Kaplan and Kay, 1994b). This means the memory required to compute the output is bound by a constant no matter the size of the input. Furthermore, in both cases, the computational conditions on vowel harmony can be even more strongly characterized, using familiar phonological notions of tiers, which relativizes locality conditions (Goldsmith, 1976). This is in line with Heinz (2011), who argues many phonological analyses can be made in terms of weaker, *subregular* classes.

We point out an interesting open issue in our understanding of vowel harmony processes which must take into account non-local contexts, both on the left and on the right of the triggering vowel. The last section briefly reviews how these results have informed work on computational learning models of vowel harmony.

2 Complexity of Surface Forms

In this section, we focus on the subregular class of *tier-based strictly local* (TSL) grammars that capture long-distance dependencies locally by projecting elements relevant for a certain process on a tier, therefore "ignoring" all the intervening irrelevant material. For vowel harmony, typically it is some set of vowels that are projected. This section begins by formally defining this class.

These grammars evaluate surface forms. If a surface form violates some harmony constraint, then as we illustrate, one should be able to write a TSL grammar accounting for that fact.

We show that a single tier is always enough for vowel harmonies. When there are several harmonic processes occurring within the same language, we refer to this as *double harmony*. For these cases, we observe that if more than a single tier alphabet is needed, those tier alphabets can be either in a subset-superset relation, or completely incomparable. Interestingly, the case when the tier alphabets are partially overlapping seems to be unattested.

2.1 Formal Languages: Preliminaries

We begin by reviewing the Strictly Local grammars and languages. Readers are referred to Rogers and Pullum (2011) and Rogers et al. (2013) for more details on this class and related classes.

2.1.1 Strictly Local Grammars

The core idea behind Strictly Local grammars is that well-formed strings are those which do not contain any *forbidden substrings*. The grammar can essentially be thought of as a finite list of these forbidden substrings.

Formally, a SL grammar consists of the set of k-grams (substrings of length k) G_{SL} that must not be contained within in a well-formed string of the language.

For example, in RUSSIAN, obstruents in a cluster must agree in voicing, see the data below in (1-2).

(1)	[s]-tajat^j	'PREF-melt'	*[z]-tajat^j
(2)	[z]-bit^j	'PREF-beat'	*[s]-bit^j

The telic prefix *s-* is realized as [s] if followed by a voiceless consonant (1), and as [z] if followed by a voiced one (2). In this case, the alphabet $\Sigma = \{$s, z, t, b, j, a, i$\}$. The grammar prohibits combinations of obstruents with mixed voicing: $G_{SL} = \langle$*sz, *zs, *sb, *bs, *zt, *tz, *tb, *bt$\rangle$. Such grammar then would correctly rule out the forbidden forms of the words given above, where the prefix disagrees in voicing with the following obstruent.

As another phonological process, consider vowel harmony in LOKAA (Niger-Congo). In this language, a non-high vowel agrees with the preceding non-high vowel in ATR. Furthermore, all high vowels and consonants are transparent for the harmony. That surface forms exhibit this pattern is plausibly due to a harmony process exemplified by the SPE-style rule in (A1).

$$\text{(A1)} \begin{bmatrix} -\text{cons} \\ -\text{high} \end{bmatrix} \rightarrow \begin{bmatrix} \alpha \text{ tense} \end{bmatrix} / \begin{bmatrix} -\text{consonant} \\ -\text{high} \\ \alpha \text{ tense} \end{bmatrix} \ldots \_$$

The examples below illustrating well-formed and ill-formed surface forms are from (Akinlabi, 2009). In this case, the agreeing items are not adjacent to each other. For strings such

(3)	èsìsòn	'*smoke*'	*èsìsɔ̀n
(4)	ɛ̀sísɔ̀n	'*housefly*'	*ɛ̀sísòn
(5)	lèjìmə̀	'*matriclan*'	*lèjìmà
(6)	ɛ́kílìkà	'*kind of plant*'	*ékílìkà

as (3) or (4), we need 5-grams to capture this pattern, because there are 3 intervening elements in-between the two agreeing non-high vowels. But for (6), even this window size is not enough: there are 5 segments in-between *ɛ* and *a*. Because there is no upper bound on the amount of material which separate the two non-high vowels agreeing with respect to the [tense] feature, therefore no SL grammar can be constructed to capture this pattern. (This can be proven using arguments presented in Rogers and Pullum (2011).) It is this problem that Tier-based Strictly Local grammars are a solution for.

2.1.2 Tier-based Strictly Local Grammars

Tier-based Strictly Local grammars (Heinz et al., 2011) capture non-local dependencies by projecting elements of a certain type on the *tier* in order to achieve locality among the remote units. Then a long-distance dependency can be viewed as local over the tier, because all the intervening material in-between the two dependent elements is ignored. As is done in SL grammars, forbidden substrings of segments can be ruled out over the tier. A TSL grammar consists of the tier alphabet T, and the set of k-grams G_{TSL} that must not be found on a tier representation of a well-formed string.

Lokaa vowel harmony described in the previous subsection affected arbitrarily far non-high vowels, and only those vowels. Consequently as explained, it is not possible to construct a strictly local grammar capturing this process. But the power of TSL languages allows us to project remote dependent items on a tier, and block illicit combinations over this tier.

The table below shows the active constraints on tier-adjacent non-high vowels, first as an illciit featural configuration, and second in terms of the corresponding illicit segmental bigrams, which would constitute the forbidden substrings in the TSL grammar H_{ATR}. In

Tier of non-high vowels T = {ɛ, e, o, ə, ɔ, a}	
1.	*[α tense] [$-\alpha$ tense]
	H_{ATR} = {*ɛe, *eɛ, *ɛo, *oɛ, *ɛə, *əɛ, *ɔe, *eɔ, *ɔə, *əɔ, *ɔo, *oɔ, **aə, *əa, *ao, *oa, *ae, *ea}

Table 1: TSL grammar for Lokaa harmony

this case, the tier alphabet includes all non-high vowels presented in the language, and the pattern of ATR agreement observed in surface forms is obtained simply by blocking the combinations of non-high vowels disagreeing in their [tense] specifications.

Figure in 1 illustrates this analysis. The left subfigure shows the well-formed word *èsìsòn*.

Figure 1: ATR harmony in Lokaa

The only non-high vowels (*e* and *o*) are projected on the tier, and their combination *eo* is not among those that need to be ruled out, and thus the word *èsìsòn* is considered acceptable. Another subfigure shows the ill-formed word **èsìsɔ̀n*, where the two non-high vowels *è* and *ɔ̀* disagree in their [tense] specifications. After *è* and *ɔ̀* are projected on a tier, we see that the bigram **eɔ* is banned by the grammar, and therefore this word must be ruled out.

2.1.3 Multiple TSL Languages

In many languages, segments agree in more than just a single feature. For example, in Turkish, vowels agree in backness and rounding; in Imdlawn Tashlhiyt, sibilants agree in voicing and anteriority, and in Bukusu, one agreement process involves vowels, and another one involves liquids.

The questions that we ask in this subsection is *Are there any restrictions on the relationships among the tier-alphabets in a language than has more than one long-distance feature harmony?*

There are four logical possible relations between distinct tier alphabets T_1 and T_2 as shown in Figure 2. Firstly, T_1 can be **the same** as T_2, i.e. the set of blockers and undergoers is the same for both harmonic processes (same tier). Jumping ahead, this case seems to

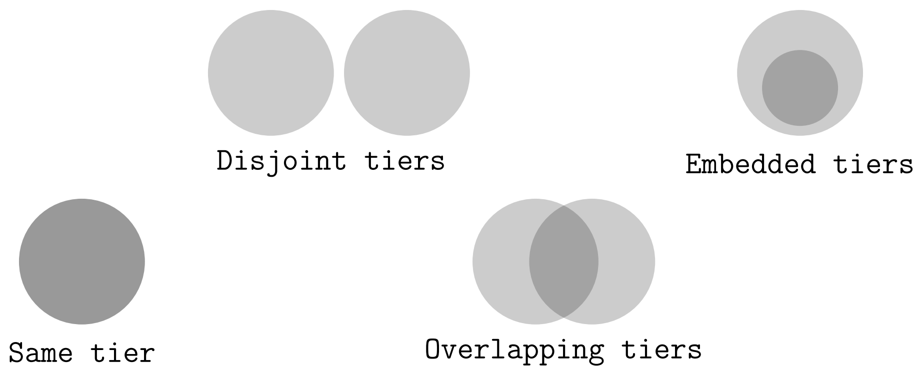


Figure 2: Possible tier relations

be always sufficient for double harmonies that only involve vowels. Alternatively, T_1 might **share no common elements** with T_2, and this is the case of independent vowel and consonant harmonies (disjoint tiers). T_1 can also be **a subset of** T_2, and it seems to be the case for some of the consonant harmonies (embedded tiers). However, the case when T_1 is only **partially overlaps** with T_2 seems to be typologically unattested (overlapping tiers).

2.2 Attested Patterns

In this section, we use the previously discussed TSL approach to capture different harmonic patterns. A harmonic process picks out a set of elements (*undergoers*) and establishes an agreement relation among them with respect to a certain feature. This agreement might be blocked: there are some items (*blockers*) that can stop this spreading, so the previous harmonizing element will not affect the following one if a blocker is found in-between them. All other segments are *transparent* for the long-distance assimilation – they do not affect and are unaffected by the harmonic process. These terms are useful even when discussing static phonotactic patterns, even though we we introduced them in the context of *processes*, and they are often used in that context as well.

A typological study of double vowel harmony patterns show that they require just a single tier, even though more than one feature is being transmitted. For sibilant harmonies, one tier is not enough. Two tiers need to be projected, and their tier alphabets are in a set-subset relation. Languages with both vowel and consonantal harmonies also require two tiers. However, in this case those tiers do not share any common element in their tier alphabets, i.e. they are disjoint. So far, there are no attested cases of intersecting alphabets, when the sets of items involved in the two spreadings only partially overlap. Out of the logical possibilities, the partially overlapping case is the most numerous!

2.2.1 Double Vowel Harmonies

The naïve expectation could be that each of the harmonic features of the double harmony will require its own tier. However, the data shows that in the case of double vowel harmony, both harmonies fit on the same tier. Note that it does not mean that undergoers and blockers are the same for both harmonies, it only means that none of the items taking part in one harmony is irrelevant for the other one. In most of the cases, one of the harmonies affects all vowels, and the other one spreads its feature only among the tier-adjacent [α high] vowels, whereas [$-\alpha$ high] ones function as blockers (TURKISH, MONGOLIAN, BURYAT, TATAR). Sometimes two features are transmitted simultaneously (KIRGHIZ). In other cases, all vowels function as undergoers with respect to both harmonies, while only a subset of those vowels can further spread both harmonic features – the others function as harmonizing blockers (YAKUT). See Kaun (1995) for numerous examples of labial feature spreadings that commonly serve as one of the two harmonies in a double harmonic system. This makes it possible to formulate two main requirements for the abstract harmonies A and B in order to fit them on the same tier:

- the set of items transparent for the harmonies A and B must be identical; and
- the blockers and undergoes for the harmony A must play some role in the harmonic process B, i.e. they must be either blockers or undergoers.

Here we show three examples of such double spreadings: fronting and rounding harmony in KIRGHIZ, where two features are transmitted together, i.e. the set of undergoers is the same for both spreadings; ATR and rounding harmony in BURYAT, where some undergoers for the ATR harmony function as blockers for the labial one; and fronting and labial harmony in YAKUT, where some harmonizing items block the spreading in certain configurations.

Kirghiz (Turkic) The rule of double vowel harmony in KIRGHIZ is to spread the features of frontness and rounding simultaneously: all vowels within a word must agree in these two features. This would be a consequence of the rule in (B1).(B1) $\begin{bmatrix} -\text{cons} \end{bmatrix} \rightarrow$

$$\begin{bmatrix} \alpha\text{ front} \\ \beta\text{ round} \end{bmatrix} / \begin{bmatrix} -\text{cons} \\ \alpha\text{ front} \\ \beta\text{ round} \end{bmatrix} \ldots \_$$

A locative affix with a non-high vowel (*-de, -dö, -da, -to*) and a genitive affix with a high vowel (*-nɨn, -din, -dün, -tun*) are used in the examples (5-12) below from (Nanaev, 1950). Here I present the data of the southern dialect of KIRGHIZ, see Herbert and Poppe (1963) and Kaun (1995) for other dialects of this language.

(5)	kɨz-da	'girl-LOC'	(9)	kɨz-nɨn	'girl-GEN'
(6)	ot-to	'fire-LOC'	(10)	ot-tun	'fire-GEN'
(7)	kim-de	'who-LOC'	(11)	kim-din	'who-GEN'
(8)	üj-dö	'house-LOC'	(12)	üj-dün	'house-GEN'

There are two harmonizing features that can be either plus or minus, therefore all vowels within a word have a choice among the following four possibilities: [+front, +round], [+front, –round], [–front, +round], and [–front, –round]. For each of these options, there is an example in the data above that shows that affixes agree with the root in these two parameters. For instance, fronted rounded vowel in *üj* 'house' spreads its feature specifications both to non-high (8) and high (12) affixes. Table2 shows the feature configurations that must be avoided in the well-formed vowel sequence of KIRGHIZ words: any bigram where the two vowels disagree in fronting or rounding must be blocked.

Vowel tier **T = {a, ɨ, e, i, o, ö, u, ü}**	
1.	*[α front] [β front]
	H_{front} = {*ai, *ae, *aö, *aü, *oi, *oe, *oö, *oü, *ɨi, *ɨe, *ɨö, *ɨü, *ui, **ue, *uö, *uü, *ia, *io, *iɨ, *iu, *ea, *eo, *eɨ, *eu, *öa, *öo, *öɨ, *öu, **üa, *üo, *üɨ, *üu}
2.	*[α round] [β round]
	H_{round} = {*oa, *oɨ, *ua, *uɨ, *öi, *öe, *üi, *üe, *ao, *au, *iö, *iü, **ɨo, *ɨu, *eö, *eü, *oe, *oi, *öa, *öɨ, *ue, *ui, *üa, *üɨ, *eo, *io. **aö, *ɨö, *eu, *iu, *aü, *ɨü}

Table 2: TSL grammar for Kirghiz harmony

Both spreadings operate over the same set of segments, therefore the tier alphabet T consists of all the vowels. To enforce the frontness harmony, combinations of vowels disagreeing in their [front] specification must be blocked by H_{front}. In addition to it, substrings with vowels of different [round] specification also need to be ruled out by H_{round}. Note that only one TSL grammar is needed to capture this pattern. Its tier alphabet is T, and the grammar G_{TSL} is a combination of H_{front} and H_{round} sets of the banned bigrams.

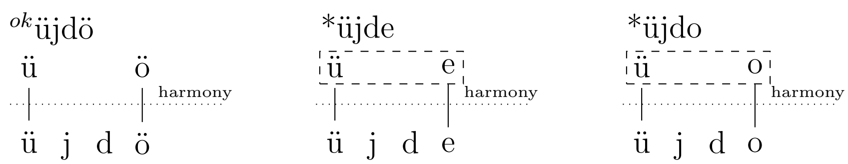


Figure 3: Frontness and labial harmony in Kirghiz

Consider Figure 3 as the graphic example of the grammar constructed above in action. The left subfigure represents the well-formed word *üjdö*. Both vowels are projected on a tier, and nothing is blocked, because the combination *üö* is permitted: both *ü* and *ö* agree in their fronting and rounding features. The middle subfigure shows a violation of the rounding harmony, wherefore bigram **üe* is ruled out by the H_{round} part of the grammar. Likewise, the fronting harmony violation (see the right subfigure) is ruled out by H_{front}, because **üo* is listed as an ill-formed substring. In Kirghiz, the set of undergoers for one harmonic spreading is exactly the same as for the other one, so, as a result, only one TSL grammar is needed to capture the desired pattern.

Buryat (Mongolian) In Buryat, all vowels within a word must agree in ATR. All tier-adjacent non-high vowels agree in rounding, unless there is an intervening high vowel that blocks this assimilation. As a result, the only possible positions in which rounded non-high vowels can be found are the initial syllables, and in other positions if they are licensed by the labial spreading (Poppe, 1960). The set of transparent items is the same for both harmonies, and includes the only non-rounded high vowel /i/ and all consonants, see (van der Hulst and Smith, 1987; Skribnik, 2003; Svantesson et al., 2005) for details. This set of well-formed surface forms can be understood as the consequence of the SPE-style rules (C1-3). (C1)

$$\begin{bmatrix} -\text{cons} \end{bmatrix} \rightarrow \begin{bmatrix} \alpha\ \text{tense} \end{bmatrix} / \begin{bmatrix} -\text{cons} \\ \alpha\ \text{tense} \end{bmatrix} \ \ldots \text{ or /i/} \quad \_$$

(C2) $$\begin{bmatrix} -\text{cons} \\ -\text{high} \end{bmatrix} \rightarrow \begin{bmatrix} \alpha\ \text{round} \end{bmatrix} / \begin{bmatrix} -\text{cons} \\ -\text{high} \\ \alpha\ \text{round} \end{bmatrix} \ \ldots \text{ or /i/} \quad \_$$

(C3) $$\begin{bmatrix} -\text{cons} \\ -\text{high} \end{bmatrix} \rightarrow \begin{bmatrix} -\text{round} \end{bmatrix} / \begin{bmatrix} -\text{cons} \\ +\text{high} \end{bmatrix} \ \ldots \text{ or /i/} \quad \_$$

In the examples below, the harmony process is illustrated using the following two affixes: the causative suffix *-ʊ:l, -u:l*, where the vowel is specified as high, therefore agrees with the stem only in ATR; and the perfective affix *-a:d, -ɔ:d, -e:d, -o:d* with a non-high vowel that agrees with the preceding segment in ATR and, if that segment is also non-high, in rounding. In (13), the non-high perfective affix agrees with the non-high root vowel in ATR

(13)	ɔr-ɔ:d	'enter-PERF'	*ɔr-a:d
(14)	ɔr-ʊ:l-a:d	'enter-CAUS-PERF'	*ɔr-ʊ:l-ɔ:d
(15)	to:r-o:d	'wander-PERF'	*to:r-e:d
(16)	to:r-u:l-e:d	'wander-CAUS-PERF'	*to:r-u:l-o:d
(17)	mɔrin-ɔ:	'horse-POSS'	*mɔrin-a:
(18)	o:rin-go:	'group-POSS'	*o:rin-ge:

and rounding: both vowels are lax and rounded. But adding the high causative affix in-between them, as in (14), results in the blocking of the labial spreading: the perfective affix no longer agrees with the stem in rounding, because they are separated from each other by an intervening high vowel. Examples (15, 16) show the same effect for the tense root. Examples (17) and (18) show the transparency of the intervening vowel /i/.

Table 3 factors the TSL grammar for Buryat into its major parts. The set of elements involved in each of these two spreadings is the same, so the tier alphabet T consists of all relevant vowels for harmony. H_{ATR} blocks combinations of vowels that disagree in tense, and $H_{r1} \cup H_{r2}$ enforce the tier-adjacent non-high vowels to harmonize in rounding, and also block rounded non-high vowels after the high ones. Again, only one TSL grammar is needed: its tier alphabet is T, and $G_{TSL} = H_{ATR} \cup H_{r1} \cup H_{r2}$.

Vowel tier (except /i/) **T = {a, e, ɔ, o, ʊ, u}**	
1.	*[α tense] [β tense]
	H_{ATR} = {*ɔo, *oɔ, *ʊu, *uʊ, *ae, *ea, *ʊo, *oʊ, *ʊe, *eʊ, *ɔu, *uɔ, **ɔe, *eɔ, *oa, *ao, *au, *ua}
2.	*[– high, α round] [– high, β round]
	H_{r1} = {*ɔa, *aɔ, *eo, *oe, *ao, *oa, *eɔ, *ɔe}
3.	*[+ high] [– high, + round]
	H_{r2} = {*ʊɔ, *uo, *uɔ, *ʊo}

Table 3: TSL grammar for BURYAT harmony

Figure 4 illustrates. The first line of the Figure 4 shows that the tier-adjacent non-high vowels must agree in rounding, therefore the tier bigram *oo* is allowed, whereas illicit substrings such as **oe* are ruled out by the grammar H_{r1}. But if there is an intervening high vowel in-between them, as on the second line of the figure 4, labial harmony does not happen: occurrence of the rounded non-high vowel after the high one (**uo*) is also blocked by the grammar. As this analysis shows, Buryat double vowel harmony also requires just a single tier, because both spreadings operate over the same alphabet.

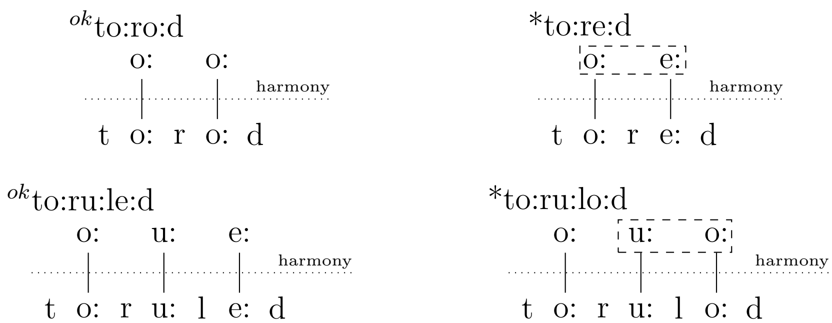


Figure 4: ATR and labial harmony in Buryat

Yakut (Turkic) In Yakut, all vowels must agree in fronting. Labial harmony spreads from the low vowels onto both low and high ones, from the high vowels to the high ones, but it cannot spread from the high vowels to the low ones. The latter ones in this case function as *harmonizing blockers*: they harmonize with any previous vowel, but block the rounding assimilation in [+high][–high] configuration. See (Sasa, 2001, 2009) for details and an OT account. This pattern is the consequence of the rules outlined in (D1-4). (D1) $\begin{bmatrix} -\text{cons} \end{bmatrix}$

$$\rightarrow \begin{bmatrix} \alpha\ \text{front} \end{bmatrix} / \begin{bmatrix} -\text{cons} \\ \alpha\ \text{front} \end{bmatrix} \ldots \_$$

$$\text{(D2)} \begin{bmatrix} -\text{cons} \\ +\text{high} \end{bmatrix} \rightarrow \begin{bmatrix} \alpha\ \text{round} \end{bmatrix} / \begin{bmatrix} -\text{cons} \\ \alpha\ \text{round} \end{bmatrix} \ldots \_$$

$$\text{(D3)} \begin{bmatrix} -\text{cons} \\ -\text{high} \end{bmatrix} \rightarrow \begin{bmatrix} \alpha\ \text{round} \end{bmatrix} / \begin{bmatrix} -\text{cons} \\ -\text{high} \\ \alpha\ \text{round} \end{bmatrix} \ldots \_$$

$$\text{(D4)} \begin{bmatrix} -\text{cons} \\ -\text{high} \end{bmatrix} \rightarrow \begin{bmatrix} -\text{round} \end{bmatrix} / \begin{bmatrix} -\text{cons} \\ +\text{high} \end{bmatrix} \ldots \_$$

The accusative affix *-(n)ü, -(n)u, -(n)ɨ, -(n)i* with a high vowel and the plural marker *-lor, -lör, -lar, -ler* with a non-high vowel demonstrate this pattern, see the examples (19-26) below from (Kaun, 1995).

High suffixal vowels agree with any preceding vowel in both fronting and rounding (21-24), whereas low vowels while always agree in fronting, and agree in rounding only when the preceding vowel is low (19, 20), otherwise they are unrounded (25, 26).

Table 4 factors the TSL grammar for Yakut into its major parts. The tier alphabet T consists of all Yakut vowels. H_{front} rules out sequences of vowels that disagree in fronting,

(19)	oɣo-lor	‘child-PL’	*oɣo-lar
(20)	börö-lör	‘wolf-PL’	*börö-ler
(21)	oɣo-nu	‘child-ACC’	*oɣo-nɨ
(22)	börö-nü	‘wolf-ACC’	*börö-ni
(23)	murum-u	‘nose-ACC’	*murum-ɨ
(24)	tünnük-ü	‘window-ACC’	*tünnük-i
(25)	ojum-lar	‘shaman-PL’	*ojum-lor
(26)	tünnük-ler	‘window-PL’	*tünnük-lör

whereas $H_{r1} \cup H_{r2} \cup H_{r3}$ blocks occurrence of a rounded low vowel if it is preceded by a high one, and also any other combination of vowels that disagree in their labial features. The obtained TSL grammar operates over the tier alphabet T and its list of illicit substrings is $G_{TSL} = H_{front} \cup H_{r1} \cup H_{r2} \cup H_{r3}$.

<table>
<tr><th colspan="2">Vowel tier
T = {a, ɨ, e, i, o, ö, u, ü}</th></tr>
<tr><td rowspan="2">1.</td><td>*[α front] [−α front]</td></tr>
<tr><td>H_{front} = {*ai, *ae, *aö, *aü, *oi, *oe, *oö, *oü, *ɨi, *ɨe, *ɨö, *ɨü, *ui, **ue, *uö, *uü, *ia, *io, *iɨ, *iu, *ea, *eo, *eɨ, *eu, *öa, *öo, *öɨ, *öu, **üa, *üo, *üɨ, *üu}</td></tr>
<tr><td rowspan="2">2.</td><td>*[+ high, α round] [+ high, −α round]</td></tr>
<tr><td>H_{r1} = {*uɨ, *üi, *iü, *ɨu, *iu, *ɨü, *ui, *üɨ}</td></tr>
<tr><td rowspan="2">3.</td><td>*[+ high] [– high, + round]</td></tr>
<tr><td>H_{r2} = {*üö, *uo, *iö, *ɨo, *io, *ɨö, *uö, *üö}</td></tr>
<tr><td rowspan="2">4.</td><td>*[– high, α round] [−α round]</td></tr>
<tr><td>H_{r3} = {*oa, *oɨ, *öi, *öe, *ao, *au,*eö, *eü, *aö, *aü, *eo, *eu, *oi, **oe, *öa, *öɨ}</td></tr>
</table>

Table 4: TSL grammar for YAKUT harmony

Figure 5 shows that under this analysis, the word *murumu* is well-formed – the labial harmony spreads among high vowels. It also spreads from the non-high vowel *o* to the following high vowel *u* in the word *ojumlar*. However, it cannot spread from a high vowel to a low one, so **ojumlor* is blocked as the illicit bigram **uo* is found on its vowel tier. As in the previous cases, a single tier is enough to capture YAKUT double vowel harmony.

2.2.2 Double vowel harmonies: summary

In the previous three subsections, we showed three examples of double vowels harmonies, and for all of them, only one TSL grammar was needed in order to capture the pattern. It was possible because neither of the vowels taking part in one spreading was irrelevant for the other harmony. Briefly, the discussed patterns were the following.

1. **Kirghiz:** all vowels within a word harmonize in both [front] and [round] features, so all vowels are undergoers for the both spreadings.

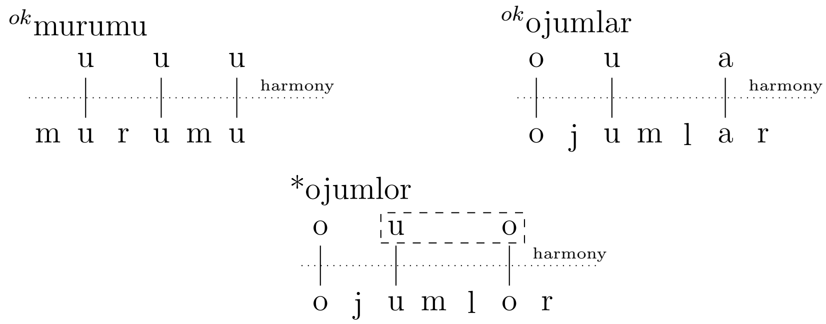


Figure 5: Fronting and labial harmony in YAKUT

2. **Buryat:** all vowels within a word harmonize in [tense], and all non-high vowels agree with respect to the [round] feature, whereas the intervening high vowels block this spreading. As the result, all vowels are undergoers for the [tense] harmony, additionally, non-high vowels are undergoers for the labial one, and the high vowels function as blockers in the latter case.

3. **Yakut:** all vowels harmonize in [front] and [round] except for the configuration when a high vowel precedes the non-high one: in this case, the labial spreading is blocked, whereas the fronting harmony is unaffected.

2.2.3 Double Sibilant Harmonies

In the previous section, we showed that one tier is enough for double vowel harmonies. Although there are harmonies that also require just a single tier, there are cases when it is impossible to avoid projection of two tiers. Here we motivate this with an example from sibilant harmony. In this case, their tier alphabets are in the set-subset relation.

Imdlawn Tashlhiyt (Berber) In IMDLAWN TASHLHIYT, affixal sibilants regressively harmonize with the stem in voicing and anteriority, see (Hansson, 2010b; McMullin, 2016). Whereas the anteriority harmony is not blocked by anything, the voicing one is blocked by any intervening voiceless obstruents. If there are no sibilants in the stem, the sibilant is realized as [s]. The SPE-style rules giving rise to this pattern are given below in (F1-4).

(F1) $$\begin{bmatrix} +\text{ cons} \\ +\text{ cont} \end{bmatrix} \rightarrow \begin{bmatrix} \alpha\text{ anterior} \end{bmatrix} / \ \_\ \ldots \begin{bmatrix} +\text{ cons} \\ +\text{ cont} \\ \alpha\text{ anterior} \end{bmatrix}$$

(F2) $$\begin{bmatrix} +\text{ cons} \\ +\text{ cont} \end{bmatrix} \rightarrow \begin{bmatrix} \alpha\text{ voice} \end{bmatrix} / \ \_\ \ldots \begin{bmatrix} +\text{ cons} \\ +\text{ cont} \\ \alpha\text{ voice} \end{bmatrix}$$

(F3) $$\begin{bmatrix} +\text{ cons} \\ +\text{ cont} \end{bmatrix} \rightarrow \begin{bmatrix} -\text{ voice} \end{bmatrix} / \ \_\ \ldots \begin{bmatrix} +\text{ cons} \\ -\text{ sonorant} \\ -\text{ voice} \end{bmatrix}$$

(F4) $\begin{bmatrix} +\text{cons} \\ +\text{cont} \end{bmatrix} \rightarrow$ [s] / elsewhere

The data in (32-41) from (Elmedlaoui, 1995; Hansson, 2010a) illustrates the harmonic pattern using the causative prefix *s-*. In (32), there are no sibilants in the root, and the prefix

(32)	s:-uga	'CAUS-evacuate'
(33)	s-as:twa	'CAUS-settle'
(34)	ʃ-fiaʃr	'CAUS-be.full.of.straw'
(35)	z-bruz:a	'CAUS-crumble'
(36)	ʒ-m:ʒdawl	'CAUS-stumble'
(37)	s-ħuz	'CAUS-annex'
(38)	s:-ukz	'CAUS-recognize'
(39)	sˤ-rˤuˤfˤzˤ	'CAUS-appear.resistant'
(40)	s-mχazaj	'CAUS-loathe.each.other'
(41)	ʃ-quʒ:i	'CAUS-be.dislocated'

appears in its by-default, underlying form *s-*. In all other examples, this prefix agrees with the sibilant in a root in its voicing and anteriority, therefore the possible feature specifications are [–voice, +ant] (33), [–voice, –ant] (34), [+voice, +ant] (35), and [+voice, –ant] (36). However, the anteriority harmony in this language does not have blockers, whereas the voicing spreading is blocked by any intervening voiceless obstruent such as /ħ/, /k/, /f/, /χ/, or /q/. In (37-41), the sibilant in the root is voiced, but the one in the prefix is voiceless because of the intervening voiceless obstruents in-between them.

Sibilant tier $\mathbf{T}_{ant}$ **= {s, z, ʃ, ʒ}**	
1.	*[α ant] [$-\alpha$ ant]
	H_{ant} = {*sʃ, *sʒ, *ʃs, *ʒs, *zʃ, *ʃz, *ʒz, *zʒ}
Tier of sibilants and voiceless obstruents $\mathbf{T}_{voice}$ **= {s, z, ʃ, ʒ, ħ, k, f, χ, q}**	
1.	*[+ cont, α voice] [+ cont, $-\alpha$ voice]
	H_{v1} = {*sz, *zs, *ʃz, *ʃʒ, *ʒʃ, *sʒ, *ʒs, *zʃ}
2.	*[+ cont, + voice] [– sonor, – voice]
	H_{v2} = {*zħ, *zk, *zf, *zχ, *zq, *ʒħ, *ʒk, *ʒf, *ʒχ, *ʒq}

Table 5: TSL grammars for IMDLAWN TASHLHIYT harmony

One tier is not enough, because there is no limit on the number of voiceless obstruents in-between the two sibilants agreeing in anteriority, therefore this process is not local – the locality required for the anteriority harmony cannot be achieved over a single tier that contains both sibilants and voiceless obstruents; see the appendix B explaining this problem in a more formal way. The only solution is to project two tiers, putting only sibilants on one of them (T_{ant}) and blocking the bigrams disagreeing in anteriority (H_{ant}); and projecting both sibilants and voiceless obstruents on the second one (T_{voice}), and blocking sibilants disagreeing in anteriority (H_{v1}) and voiced sibilants followed by voiceless obstruents (H_{v2}).

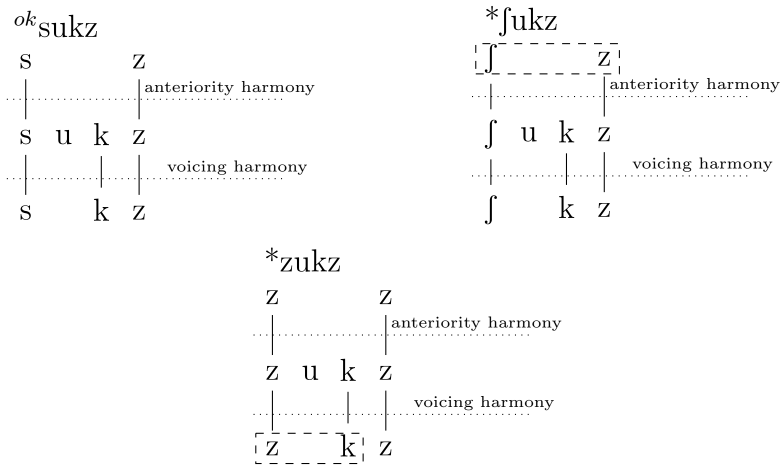


Figure 6: Sibilant harmony in Imdlawn Tashlhiyt

Figure 6 shows this analysis of the Imdlawn Tashlhiyt double sibilant harmony. The word *sukz* is well-formed, because the anteriority grammar allows for the *sz* combination: they both agree in anteriority, and the voicing tier is satisfied with the bigrams *sk* and *kz*. However, **ʃukz* is ruled out because the **ʃz* combination is banned over the anteriority tier: note that the sibilants *ʃ* and *z* are not adjacent over the tier of the voicing harmony. The word **zukz* is also out, because the voicing grammar prohibits voiced sibilants followed by the voiceless obstruents (**zk*), and this blocker is not seen over the anteriority harmony tier. Imdlawn Tashlhiyt pattern requires two tiers, because the set of the elements taking part in one long-distance assimilation is different from the one involved in the other dependency: two tiers are required, and their alphabets are in the set-subset relation.

2.2.4 Vowel Harmony and Consonant Harmony

Before, we showed double harmonies only among vowels or only among consonants. Here, the two harmonies target different sets of elements – one of them is consonantal harmony, and the second one is vowel assimilation.

Kikongo (Bantu) There are both consonant and vowel harmonies in Kikongo. Vowel harmony enforces vowels to agree in height, whereas nasal agreement turns /d/ and /l/ into /n/ if preceded by a nasal in the stem, see (Ao, 1991; Hyman, 1998) and the rule in (G1). The data below is adopted from (Hyman, 1998). In these examples, the applicative suffix *-el, -il*, and the reversive transitive suffix *-ol, -ul* illustrate the vowel harmony in height.

(G1) $\begin{bmatrix} -\text{cons} \end{bmatrix} \rightarrow \begin{bmatrix} \alpha \text{ high} \end{bmatrix} / \begin{bmatrix} -\text{cons} \\ \alpha \text{ high} \end{bmatrix} \dots \_$

In this language, suffixes are specified for rounding, and acquire their height specification depending on the stem vowel. In (42, 43) and (46, 47), both vowels in the stem and in the affix are non-high, whereas (44, 45) and (48, 49) contain the high ones.

This harmony operates over the tier of vowels T_v, and the grammar must rule out all combinations of vowels that disagree in height. But along with vowel harmony, this language

(42)	-somp-el-	'attach-APPL'	(46)	-tomb-ol-	'do-TRANS'
(43)	-leng-el-	'languish-APPL'	(47)	-lemb-ol-	'broom-TRANS'
(44)	-sik-il-	'support-APPL'	(48)	-vil-ul-	'move-TRANS'
(45)	-vur-il-	'surpass-APPL'	(49)	-bub-ul-	'bribe-TRANS'

Vowel tier $\mathbf{T}_v = \{\mathbf{e, o, i, u}\}$	
1.	*[α hi] [β hi]
	H_v = {*oi, *ou, *ei, *eu, *io, *ie, *uo, *ue}

Table 6: TSL grammar for KIKONGO vowel harmony

also has a consonantal one – nasal feature spreading formalized in (G2). Segments /d/ and /l/ in the affix both become /n/ if nasal consonants /m/ or /n/ are present in a root. See examples below from (Ao, 1991), where *-idi* is the perfective active suffix, and *-ulu* is its passive counterpart.

(G2) /d/, /l/ → [n] / [m], [n] _

(50)	-suk-idi-	'wash-PERF.ACT'
(51)	-nik-ini-	'ground-PERF.ACT'
(52)	-meng-ene-	'hate-PERF.ACT'
(53)	-suk-ulu-	'wash-PERF.PASS'
(54)	-nik-unu-	'ground-PERF.PASS'
(55)	-meng-ono-	'hate-PERF.PASS'

In (50, 51), there are no nasals in a root, so the consonant in the affix is unchanged – it remains /d/ and /l/ respectively. However, when there are nasals /n/ or /m/ in the stem, both affixal /d/ and /l/ change to /n/, see (51, 54) and (52, 55) respectively.

$\mathbf{T}_n = \{\mathbf{n, m, d, l}\}$	
1.	*d [+ nasal], *l [+ nasal]
	H_n = {*nd, *nl, *md, *ml}

Table 7: TSL grammar for KIKONGO consonant harmony

Only /d/, /l/, and nasals are involved in the process, therefore those are projected over the tier, and the grammar H_n blocks occurrence of /d/ and /l/ after the nasals. The two tier-based strictly local grammars have absolutely different tier alphabets T_v and T_n, and cannot be combined together, because nasals can occur in-between vowels, as well as vowels in-between nasals.

Figure 7 illustrates such an analysis for the KIKONGO pattern. Two tiers are necessary for the description of this pattern, because only those can provide the needed locality relations among vowels for vowel harmony, and /d/, /l/ and nasals for the nasal assimilation. The well-formed word *nikunu* is permitted because its vowel tier string *iuu* does not violate the vowel harmony rule, and the nasal tier *nn* also satisfies the nasal harmony. Such ill-formed

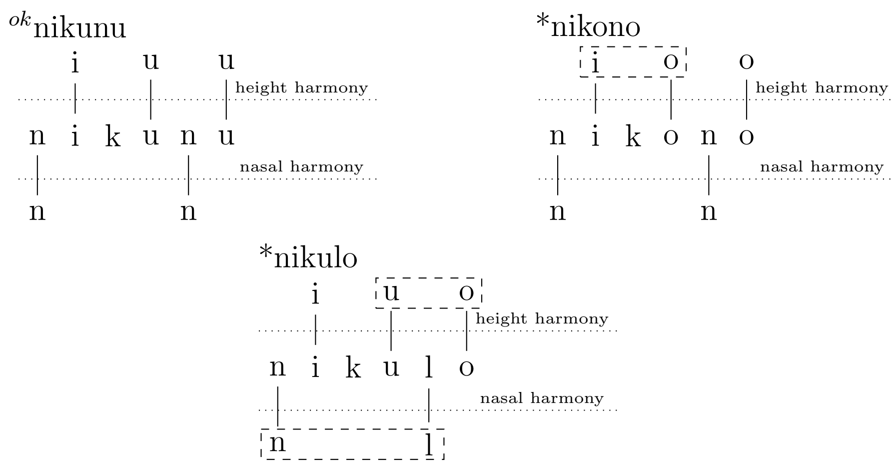


Figure 7: Vowel and nasal harmonies in Kikongo

combinations of segments as **io, *uo*, and **nl* are ruled out by the corresponding grammars. Note that the two vowels /i/ and /u/ are intervening between /n/ and /l/ in the rightmost subfigure, and only existence of the two separate tiers creates locality needed for the TSL analysis of this pattern.

2.2.5 Interim Summary

Table 8 summarizes the harmonic patterns in the different languages in this section where dependencies exist among more than one feature. The table briefly describe the surface pattern, and also its type in terms of the tier alphabet(s). **S** stands for a single tier – only one TSL grammar is required to capture both dependencies. **E** stands for embedding, i.e. one tier alphabet is a subset of the other. **D** means that the two tier alphabets are disjoint.

To our knowledge, there are no examples of the fourth logically possible type, where the two tiers partially overlap (**P**). As Aksënova and Deshmukh (2018) show, if this absence were formulated as a linguistic hypothesis on phonologically possible tiers, it would drastically reduce the predicted tier combinations. They write "For a set of 10 elements, this limitation excludes 95% of all possible tier alphabet organizations." In other words, the attested types {**S, E, D**} constitute only a small fraction of the four logically possible types {**S, E, D, P**}.

Type	Language	Description	References
S	Kirghiz	All vowels agree both in fronting and rounding.	Kaun (1995)
S	Kachin Khakass	All vowels agree in fronting. Tier-adjacent high vowels agree in rounding.	Korn (1969)
S	Yakut	All vowels agree in fronting. High vowels agree with any preceding vowel, non-high ones agree only with preceding non-high vowel.	Sasa (2001)
S	Karachay, Azerbaijani, Tuvan, Uyghur	All vowels agree in fronting. High vowels agree with any preceding vowel in rounding.	Kaun (1995) Comrie (1981) Krueger (1977)
S	Kazakh	All vowels agree in fronting. In fronted words, all vowels agree in rounding. In non-fronted ones, only high vowels agree with the preceding vowel in rounding.	Korn (1969)
S	Shor	All vowels agree in fronting. In fronted words, all vowels agree in rounding. In non-fronted ones, tier-adjacent vowels agree in rounding only if they agree in height.	Korn (1969)
S	Buryat, Mongolian	All vowels agree in ATR. Non-high vowels agree in rounding, high vowels block it. Transparent element for both harmonies is /i/.	Poppe (1960) Svantesson et al. (2005)
S	Even, Oroch, Ulcha	All vowels agree in RTR. Tier-adjacent non-high vowels agree in rounding.	Comrie (1981)
S	Tuareg	All sibilants agree both in voicing and anteriority.	Hansson (2010b)
E	Imdlawn Tashlhiyt	Sibilants agree in voicing and anteriority. Voiceless obstruents block the voicing harmony and are transparent for the anteriority one.	Hansson (2010a)
D	Kikongo, Kiyaka	Vowels agree in height. Consonants /d/ and /l/ assimilate to /n/ if preceded by a nasal.	Ao (1991) Hyman (1998)
D	Bukusu	Vowels agree in height. Liquid /l/ assimilates to /r/ if preceded by /r/.	Odden (1994)

Table 8: A summary of the reviewed phonotactic patterns with multiple long-distance dependencies.

3 Complexity of Transformations

3.1 Harmony as a Process

Whereas the previous section analyzed long-distance dependencies in terms of constraints on surface forms, this section analyzes long-distance assimilation as a phonological *process*. Modern generative analysis conceives of the phonological grammar as a function which maps an abstract lexical representation (the 'underlying form') to a more concrete—but still abstract— phonological representation (the 'surface form'). In this sense, vowel harmony refers to a systematic pattern of pronunciation in which certain features of vowels which are different at the underlying level are the same at the surface level.

As an example, Table 9 shows two allomorphs of the Turkish genitive suffix, [-in] and [-un] (Nevins, 2010, p.32). The allomorph is predictable based on the front/back dimension of the preceding vowel: if the preceding vowel is front then [-in] occurs, but if it is back then

[-un] occurs We may consider a generative phonological analysis of the Turkish forms above,

	noun	genitive	gloss
a	ip	ip-in	rope
b	el	el-in	and
c	son	son-un	end
d	pul	pul-un	stamp

Table 9: Turkish genitive allomorphs

which posits the underlying form of the genitive suffix to be /-un/ and a mapping f which derives the surface forms as shown below.

w	$f(w)$
/ip-un/	[ip-in]
/el-un/	[el-in]
/son-un/	[son-un]
/pul-un/	[pul-un]
...	...

Following Heinz and Lai (2013), we can abstract these examples of vowel harmony into the following map from underlying to surface form: $/+-/ \rightarrow [++]$. Here underlying and surface forms each contain two vowels, and consonants are left out as they are 'off the tier'. The mapping shows the binary value of the feature F for each vowel, which is an abstraction of any of the features which participate in the harmony. At the underlying level, the values of the first and second vowel are $/+/$ and $/-/$, respectively. At the surface level, however, both vowels bear the value $[+]$ for feature F.

What classes of underlying to surface mappings f adequately characterize harmonic processes? Heinz and Lai (2013) analyze several types of logically possible vowel harmony maps discussed in the phonological literature, summarized in Table 10. Here we discuss these various types of patterns. As mentioned, these patterns have abstracted away from consonants and just focus on the vowels 'on the tier.' There is work on generalizing the TSL class to string-to-string functions (Burness and McMullin, 2019), which does not make this assumption, but in this section we focus on the conditions needed to characterize the processes on the tier. Readers are referred to Heinz (2018) for more detailed formal discussion.

3.2 Subregular Function Hierarchy and Unattested Maps

One plausible hypothesis is that the nature of harmony processes is sufficiently described by porting the notion of Regularity from formal languages to string-to-string functions. This is the condition imposed by Kaplan and Kay (1994a) when they describe the maps generated by SPE-style phonological rules as being regular maps, provided the rules do not apply to their original locus. The regularity condition says that certain logically possible harmony processes are unattested by virtue of being non-regular.

One logically possible but unattested map occurs if unfaithful vowels are always fewer in number than the faithful ones, called Majority Rules (MR) harmony. In constraint-interaction terms, it is the optimal outcome of two very simple violable constraints: a

	Input w	LA(w)	PH(w)	RH(w)	DR(w)	SG(w)	MR(w)	CU(w)	SC(w)
a.	$/+--/$	$[++-]$	$[+++]$	$[---]$	$[+++]$	$[+++]$	$[---]$	$[+--]$	$[---]$
b.	$/-++/$	$[--+]$	$[---]$	$[+++]$	$[+++]$	$[---]$	$[+++]$	$[-++]$	$[+++]$
c.	$/---/$	$[---]$	$[---]$	$[---]$	$[---]$	$[---]$	$[---]$	$[---]$	$[---]$
d.	$/-+-/$	$[---]$	$[---]$	$[---]$	$[+++]$	$[---]$	$[---]$	$[-+-]$	$[+++]$
e.	$/+-+/$	$[+++]$	$[+++]$	$[+++]$	$[+++]$	$[+++]$	$[+++]$	$[+++]$	$[---]$
f.	$/+-\boxminus/$	$[++\boxminus]$	$[++\boxminus]$	$[--\boxminus]$	$[++\boxminus]$	$[+-\boxminus]$	$[--\boxminus]$	$[--\boxminus]$	$[-+\boxminus]$
g.	$/+\ominus-/$	$[+\ominus-]$	$[+\ominus+]$	$[-\ominus-]$	$[+\ominus+]$	$[+\ominus+]$	$[-\ominus-]$	$[+\ominus-]$	$[-\ominus-]$

Table 10: Example mappings of underlying forms (w) given by local assimilation (LA), progressive harmony (PH), regressive harmony (RH), dominant/recessive harmony (DR), sour grapes harmony (SG), and majority rules harmony (MR), circumambient unbounded harmony (CU), and stem control (SC). Symbols [+] indicates a [+F] vowel and [−] indicates a [−F] vowel where "F" is the feature harmonizing. Symbols [$\boxminus$] and [$\ominus$] are [−F] vowels that are opaque and transparent, respectively. The stem control column assumes only the second element in the input is the stem. (Heinz and Lai, 2013)

markedness constraint banning successive vowels with different values of feature F (Agree(F)) outranking the faithfulness constraint Ident(F). Baković (2000, 26) explains the term this way:

> When Agree[F] is dominant, it winnows the candidate set down to basically two candidates, one with all [αF] segments and the other with all [-αF] segments. If IO-Ident[F] gets the next crack at the evaluation process, it will choose the one of these candidates that is least deviant from the input, regardless of the stem/affix or +/- distinctions. In other words, what ends up mattering is the relative percentages of [αF] and [-αF] vowels in the input: the underlyingly greater number of [-αF] vowels in [a map where $/+--/ \rightarrow [---]$] gangs up on the lesser number of [αF] vowels, yielding the problematic effect that I call 'majority rule.'

MR vowel harmony is not regular because the difference between the number of [αF] and [-αF] vowels can grow without limit as the length of the input words increase, thus violating the constant memory bound that the computational Regularity condition imposes.

While Regularity forms a sufficient condition that rules out some unattested but logically possible forms, is it a necessary condition? Another logically possible, but unattested type of vowel harmony process has been called Sour Grapes (SG) (Padgett, 1995; Wilson, 2003). Sour Grapes is like progressive harmony except that vowels only harmonize if no opaque vowels occur later in the word. If an opaque vowel occurs somewhere after the initial vowel, then non-neutral vowels between it and the initial vowel will not harmonize (a sour grape ruins the bunch).

Sour Grapes is in fact a regular function. In order to concretely determine the necessary and sufficient conditions for a theory of harmony processes, we need to again ask about fine-grained subclasses, this time of the regular functions. This section describes the hierarchy of such subclasses, as shown in Figure 8, and attested harmony processes that lie within them.

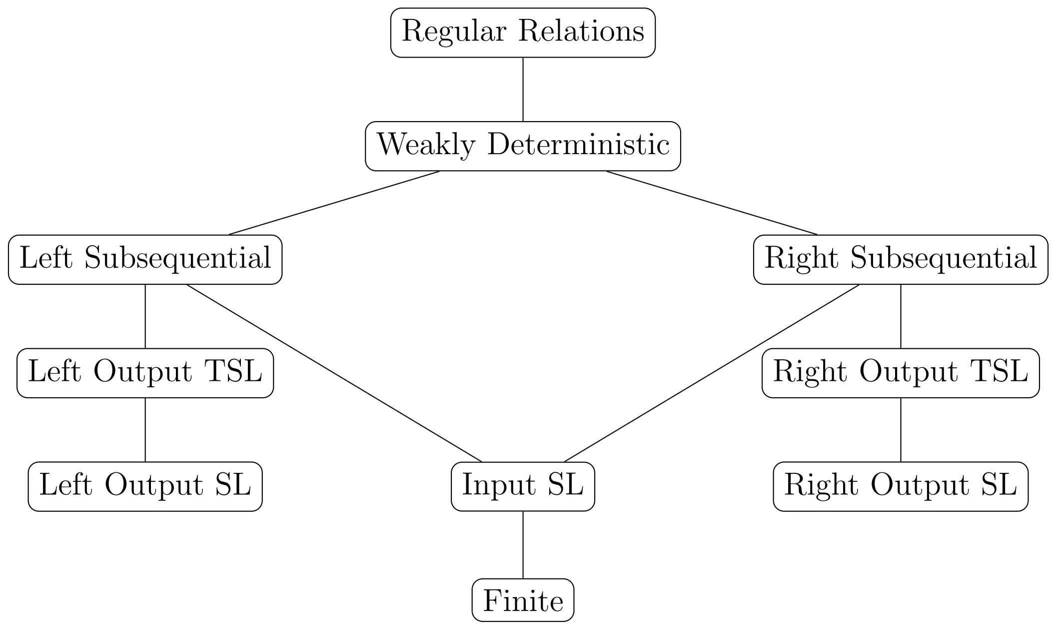


Figure 8: A hierarchy of subregular functions

3.3 Input & Output Strictly Local Functions

Let us begin at the simplest subclasses. Strictly Local functions generalize the conditions which characterize Strictly Local stringsets. Recall the defining property of Strictly Local stringsets: wellformedness of any position in a string can be determined by checking the $k-1$ previous symbols in the string. For Input Strictly Local functions, every element in the input string corresponds to a string of symbols in the output string. For any input symbol x its output string u will only depend on x and the previous $k-1$ elements of x in the input string, as shown in Figure 9 where $k = 2$. It should be easy to see that this holds regardless of whether one is reading the string from left to right or vice versa.

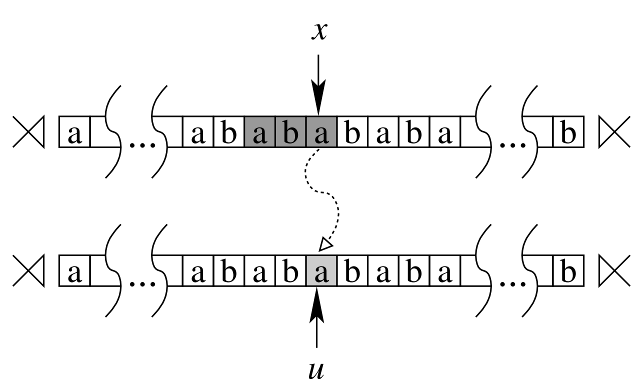
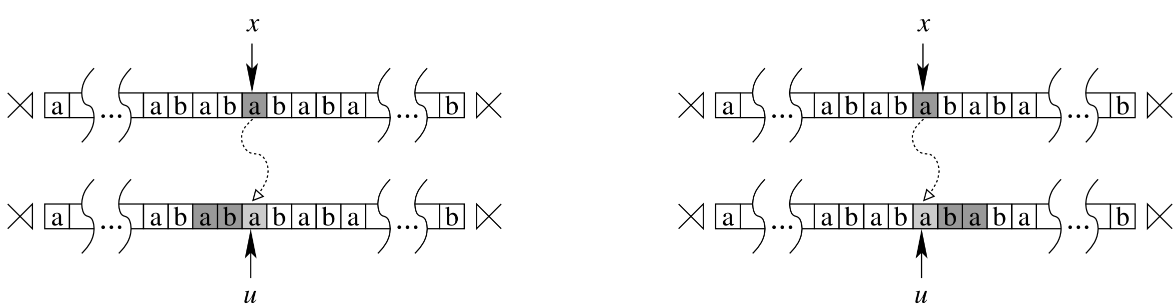


Figure 9: ISL function

One type of harmonic process defined by an ISL function can be called LOCAL ASSIMILATION (LA), in the sense that only the following vowel is affected. Local Assimilation is ISL with $k = 3$, since if the previous two elements are $/\# +/$ or $/\# -/$ (where # is a word edge) then if the current input element x is a non-neutral vowel, the output element u will be [+] or [-] respectively.

Another type of process is one where the first or last vowel determines the features of the other vowels in the word, called PROGRESSIVE HARMONY (PH) and REGRESSIVE HARMONY (RH), respectively. In this case, neutral vowels resist harmonizing and either are skipped (transparent) or force subsequent vowels to harmonize with them (opaque). Following Heinz and Lai (2013), we use the symbols [$\ominus$] and [$\boxminus$] to represent [-F] vowels that are transparent and opaque, respectively, and we use [$\oplus$] and [$\boxplus$] to represent [+F] vowels that are transparent and opaque. Various examples are given in Table 10.

Progressive and Regressive Harmony cannot be described by an ISL function, for a simple reason. Progressive harmony defines, for all numbers k, maps like $/ + -^k - / \rightarrow [+ +^k +]$ and $/ - -^k - / \rightarrow [- -^k -]$. Such a map cannot be Input Strictly Local for any k. This is because whether the last input element surfaces as $[+]$ or $[-]$ depends on an input element which is more than k input elements away, violating the bounded input window property of ISL functions. Chandlee (2014) shows that maps of these kind inhabit the Left and Right Output Strictly Local functions (LOSL and ROSL). Intuitively, whereas the context for an ISL function to write a symbol was a bounded chunk of the *input* string, in an LOSL or ROSL function the context is in the *output*, as shown in Figure 10.



(a) Left OSL function

(b) Right OSL function

Figure 10: Left and Right OSL functions

Burness and McMullin (2019) synthesize the Left and Right OSL functions with the Tier-based SL formal languages to obtain Left and Right Tier-based Output SL functions.

3.4 Subsequential Functions

What if we put the consonants back in? Specifically, what if an unbounded number of consonants are allowed to intervene between the harmonizing vowels? If so, then the Progressive and Regressive Harmony maps will not be LOSL nor ROSL, respectively, for a similar reason why PH is not ISL. For the PH case, for all numbers k, $/ + C^k - / \rightarrow [+C^k+]$ and $/ - C^k - / \rightarrow [-C^k-]$. Such a map cannot be either LOSL or ROSL, because whether the last input element surfaces as [+] or [-] depends on an output element which is more than k output elements away.

Accommodating this increase in memory requires a move up into the class of Subsequential functions, of which there are Left and Right variants. Informally, for Subsequential functions, any of the infinitely many possible input strings is classified as belonging to exactly one of finitely many regular stringsets. For any input element x of that string, the output u only depends on x and the regular stringset to which its preceding input string belongs. Figure 11 illustrates this property of subsequential functions.

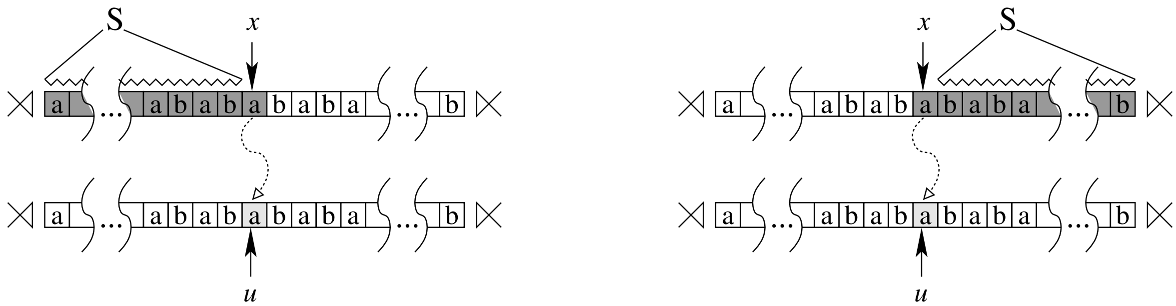


Figure 11: Left and Right Subsequential functions

The key property of Subsequential functions is similar to ISL functions of being able to 'look ahead' by outputting an empty string. The difference is that left subsequential functions can look into the right context of the input element *only* some finite distance, and vice-versa for right subsequential. The bound k on how far they can look ahead procedurally means it can only remember finitely much information about the input. This means it is not possible to remember the exact preceding input string. For this reason Subsequential functions include the Tier-based OSL and ISL functions.

3.5 Weakly Deterministic Functions

Some analyses of vowel harmony posit that vowels in a word harmonize to a particular feature value, if that feature is present somewhere in the word. This has been termed DOMINANT/RECESSIVE (DR) harmony, since the feature F appears to have a dominant value (which vowels harmonize to) and a recessive value (which they don't). The dominant/recessive map is neither left nor right subsequential. Following a similar argument as before, for all numbers k, $/ -^k + -^k / \rightarrow [+^k + +^k]$ and $/ -^k - -^k / \rightarrow [-^k - -^k]$. This cannot be left subsequential since whether the first k input elements all surface as [+] or [−] depends whether the next element is [+] or not. Even though these functions could output λ for the first k input elements, if the [+] comes next, they will still have to output $k+1$ [+] symbols. However, this is impossible: k can be any number and left subsequential functions can only classify the preceding input string into one of finitely many categories. A similar argument holds for right subsequential functions.

To accommodate patterns like DR (and SC), Heinz and Lai (2013) introduce a class of functions they call Weakly Deterministic. These are functions that can be composed of a left subsequential and right subsequential function without the introduction of new alphabetic symbols. They show that the dominant/recessive (DR) and stem-control (SC) maps are properly weakly deterministic.

3.6 Nondeterministic Regular Functions

The main takeaway from Heinz and Lai (2013) is that DR and SC maps are more computationally complex than PH and RH maps, yet weaker than fully regular maps. Nondeterministic regular functions are the composition of a left and right subsequential function, as long as the intermediate string can use additional alphabetic symbols (Elgot and Mezei, 1965).

Sour Grapes can be described as the composition of a left subsequential function and a right subsequential function, but crucially the intermediate form appears to require the use of an additional alphabetic markup symbol. For this reason, Heinz and Lai conjecture that the unattested SG map is not weakly deterministic.

The definition of weak determinism has been criticized (McCollum and Essegbey, 2019), and there is further debate about the best way to generalize Left and Right OSL functions to allow bidirectional harmony patterns without obtaining the class of non-deterministic regular functions. Research into this issue has shown that its properties may be relaxed or strengthened to accommodate some more interesting cases (Lamont et al., 2019; Smith and O'Hara, 2019).

Circumambient unbounded harmony (CU), following Jardine (2016), is also like dominant/recessive harmony in that only one value of the feature triggers the harmony, but requires two /+F/ triggers which must surround the affected vowels but perhaps arbitrarily far from them. 'Circumambient' refers to two surrounding triggers and the term 'unbounded' refers to the absence of a bound on the distance between the two triggers.

Since the presence of CU patterns is important evidence with respect to the weak determinism analysis of harmony, it is worth going through two examples in some detail. One example comes from Advanced Tongue Root (ATR) harmony in Tutrugbu (McCollum and Essegbey, 2019). The CU property of the Tutrugbu pattern can be seen in the comparison between the pairs of forms below (see McCollum et al. (2020) for data and THE cu analysis). The ATR feature spreads leftward from roots to prefixes, and [-high] prefix vowels undergo harmony (56-57) unless the initial prefix vowel is [+high], in which case harmony is blocked by [-high] vowels (58-59).

56.	/a-tí-ba-bá/	[a-tí-ba-bá]	'3S-NEG-FUT-come'
57.	/a-tí̄-ba-ʃē/	[e-tí̄-be-ʃē]	'3S-NEG-FUT-grow'
58.	/ɪ-ba-bá/	[ɪ-ba-bá]	'1S-FUT-come'
59.	/ɪ-ba-ʃē/	[ɪ-ba-ʃē]	'1S-FUT-grow'

McCollum et al. (2020) also show that Turkana features CU vowel harmony patterns (Dimmendaal, 1983; Baković, 2000). Here, the ATR feature spreads bidirectionally from the root, as shown in the examples below. Affix vowels alternate in accordance with the ATR value of root vowels, but the behavior of /ɑ/ depends if it is on the left or right of a dominant vowel: left /ɑ/ is opaque, and right /ɑ/ alternates between the [+low, −ATR] vowel [ɑ] and the [−low, +ATR] vowel [o]. [a] comes after [−ATR] roots and [o] after [+ATR] roots (seen in the epipatetic vowel).

In addition to dominant root vowels, there are dominant suffixes in Turkana. Some of these are [−ATR], and when a dominant [−ATR] suffix accompanies a [+ATR] root, the

60. /ɑ-k-ɪnɔk-ɑ-kɪn-ɪ/ [ɑ-k-ɪnɔk-ɑ-kɪn-ɪ] 'GER-K-light.fire-EPI-DAT-V'
61. /ɑ-k-ibus-ɑ-kin-i/ [ɑ-k-ibus-o-kin-i] 'GER-K-drop-EPI-DAT-V'

epipatetic vowel between them surfaces as neither [+low, −ATR] [ɑ] nor [−low, +ATR] [o], but instead [−low, −ATR] [ɔ]. In (62) the vowel surfaces as [o] after the [+ATR] root /ibus/ if no dominant [−ATR] suffix vowel follows. In (63), a dominant [−ATR] suffix occurs to the right of an 'epipatetic' vowel (Dimmendaal, 1983), which surfaces as [ɔ].

62. /e-ibus-ɑ-kin-ɪ/ [e-ibus-o-kin-i] '3-drop-EPI-DAT-V'
63. /ɛ-ɪbʊs-ɑ-kɪn-ɑ/ [ɛ-ɪbʊs-ɔ-kɪn-ɑ] 'GER-K-drop-EPI-DAT-VOI'

Crucially, McCollum et al. (2020) show that the root's ATR value alone cannot determine the surface suffixal low vowel, since the root may precede the suffix at any distance, but also depends on the presence or absence of a dominant suffix, which may follow at any distance. Since in both of these languages this information can be an unbounded distance away in both directions, ATR harmony in Tutrugbu and Turkana are unbounded circumambient.

Despite Jardine (2016)'s claim that such segmental CU harmony patterns are rare, McCollum et al. (2020) reference other unbounded circumambient segmental patterns in addition to Tutrugbu and Turkana. The Tutrugbu pattern of harmony detailed below is also attested in Tafi (Bobuafor et al., 2013), a related language. The Turkana pattern described above is also attested in Karimojong, a related Nilotic language (Novelli, 1985; Lesley-Neuman, 2012), and similar patterns are found in another Nilotic language, Toposa (Schröder and Schroeder, 1987), in the Bantu language Liko (Wit et al., 2015) and in the Dogon language Bondu-So (Hantgan and Davis, 2012).

Computing CU patterns appears then to require the power of non-determinism, moving us past the weakly deterministic boundary into the non-deterministic regular function region (Heinz and Lai, 2013; Jardine, 2016).

To illustrate this nondeterminism property, consider McCollum et al. (2020)'s analysis of the Turkana ATR rightward spread from the root in (60)–(63). Reaching the epipatetic vowel presents three options: map input /ɑ/ faithfully to output [ɑ]; map /ɑ/ to [o]; or map /ɑ/ to [ɔ]. The faithful map is chosen if the root is [−ATR], while the others depend on the root value of ATR and the presence of a dominant suffix. With a [+ATR] root or a dominant [+ATR] suffix, [o] surfaces, but with a [+ATR] root and a dominant [−ATR] suffix, [ɔ] surfaces. While the flavor of the CU property differs between Turkana, Tutrugbu, and Tafi, the realization of a suffix low vowel in Turkana is demonstrably non-deterministic.

There are two lessons to be taken from this discussion of the computational power of harmony processes. The first is that the necessary and sufficient conditions for what determines the class of harmony patterns in human languages appears to hover around weak determinism, but is still demonstrably subregular. The data that contradict the claim of strict weak determinism are accompanied by analyses which alter and refine the definition, rather than rejecting it. This will hold true of any future data that bears on the question, and brings up issues of representation and computation that all phonologists can contribute to.

The second lesson is that the computational nature of these processes still holds regardless of whether one is interested in the behavior of *all* harmony processes. Progressive Harmony is still (tier-based) OSL whether one wants it to be or not. It is also OSL whether one is modeling it with rewrite rules or constraint-interaction or any other model. This is true for any of the patterns and subregular function classes discussed. The individual characteristics of these function classes are important intrinsic properties of the particular harmony maps, and it is exactly these properties which can enable successful learning of these patterns, as outlined in the next section.

4 Learning

One of the contributions that computational analyses of language patterns enable is a direct tie to plausible and provably correct learning algorithms. The grammar classes discussed above provide target hypothesis classes for learning, to which you can ask how successfully (if at all) something can in principle or in practice induce a grammar from a pattern inhabiting that class. While there are many ways to characterize learning problems in phonology, (Heinz and Rawski, ming) they are always characterized as functions which map data to grammars. Learners are always evaluated in terms of the class of grammars they are guaranteed to learn under defined criteria of what it means to 'learn'. In the case of learning languages, the SL, SP, and TSL grammars are very efficiently learnable (Lambert et al., ress). The MTSL grammars are also learnable (McMullin et al., 2019), though with the caveat that when tiers interact as shown above, the type of interaction drastically affects learnability (Aksënova and Deshmukh, 2018). Experimentally, Lai (2015) and McMullin and Hansson (2019) have shown using artificial grammar learning experiments that adult learners attend to hypotheses consistent with these classes in contrast to competing hypotheses.

Similar evidence exists regarding the learning of processes. In particular, the properties of the subsequential class ensure that it is feasibly learnable (Oncina et al., 1993). Gildea and Jurafsky (1996) used this result to adapt their algorithm for use in phonology, and much work has focused on the learnability of the ISL and OSL subclasses (Chandlee, 2014; Chandlee et al., 2015; Jardine et al., 2014). Intriguingly, recent work on recurrent neural sequence-to-sequence architectures appear limited to learning only the subsequential class of structures (Nelson et al., 2020). The learnability of ISL functions has also prompted ways of switching the data structure away from strings to incorporate autosegmental and feature-geometry properties, which may provide a more unified way of thinking about harmony (Chandlee and Jardine, 2019; Rawski and Dolatian, 2020). Experimentally, Finley (2008) clearly established that subjects are able to learn progressive harmony patterns, but not Majority Rules patterns, with less conclusive results on learning Sour Grapes due to confounds in the opaque and transparent vowels (see Finley (2015)).

5 Conclusion

The computational analysis of harmony today bears much fruit, just as as any other subject studied computationally. These insights and issues are prescient whether one is discussing harmony using constraints on well-formedness, or as transformations from underlying and surface forms. In each case, the properties of harmony define necessary and sufficient conditions on the class of functions characterising the phonological knowledge to compute it.

These properties and conditions drive successful learning, whether as an algorithm or in the laboratory. Further work in this area will continue to shed light on the nature of phonology, and long-distance dependencies such as those found in harmony will undoubtedly continue to play an important role.

References

Akinlabi, A. (2009). Neutral vowels in lokaa harmony. *Canadian Journal of Linguistics*, 59(2):197–228.

Aksënova, A. and Deshmukh, S. (2018). Formal restrictions on multiple tiers. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2018*, pages 64–73.

Ao, B. (1991). Kikongo nasal harmony and context-sensitive underspecification. *Linguistic Inquiry*, 22(1):193–196.

Baković, E. (2000). *Harmony, Dominance and Control*. PhD thesis, Rutgers University.

Bobuafor, M. et al. (2013). *A grammar of Tafi*. LOT: Utrecht.

Burness, P. and McMullin, K. (2019). Efficient learning of output tier-based strictly 2-local functions. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 78–90, Toronto, Canada. Association for Computational Linguistics.

Chandlee, J. (2014). *Strictly Local Phonological Processes*. PhD thesis, The University of Delaware.

Chandlee, J., Eyraud, R., and Heinz, J. (2015). Output strictly local functions. In Kuhlmann, M., Kanazawa, M., and Kobele, G. M., editors, *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*, pages 112–125, Chicago, USA.

Chandlee, J. and Jardine, A. (2019). Autosegmental input strictly local functions. *Transactions of the Association for Computational Linguistics*.

Comrie, B. (1981). *Languages of the Soviet Union*. Cambridge University Press, Cambridge, England.

Dimmendaal, G. J. (1983). *The Turkana Language*. Walter de Gruyter GmbH & Co KG.

Elgot, C. C. and Mezei, J. E. (1965). On relations defined by generalized finite automata. *IBM Journal of Research and Development*, 9(1):47–68.

Elmedlaoui, M. (1995). *Aspects des représentations phonologiques dans certaines langues chamito-sémitiques*. PhD thesis, Université Mohammed V.

Finley, S. (2008). *The formal and cognitive restrictions on vowel harmony*. PhD thesis, Johns Hopkins University, Baltimore, MD.

Finley, S. (2015). Learning non-adjacent dependencies in phonology: Transparent vowels in vowel harmony. *Language*. in press.

Gildea, D. and Jurafsky, D. (1996). Learning bias and phonological-rule induction. *Computational Linguistics*, 24(4):497–530.

Goldsmith, J. A. (1976). *Autosegmental phonology*, volume 159. Indiana University Linguistics Club Bloomington.

Hansson, G. (2010a). *Consonant Harmony: Long-Distance Interaction in Phonology*. Number 145 in University of California Publications in Linguistics. University of California Press, Berkeley, CA. Available on-line (free) at eScholarship.org.

Hansson, G. O. (2010b). Long-distance voicing assimilation in berber: spreading and/or agreement? In *Proceedings of the 2010 annual conference of the Canadian Linguistic Association*, Ottawa, Canada. Canadian Linguistic Association.

Hantgan, A. and Davis, S. (2012). Bondu-so vowel harmony: a descriptive analysis with theoretical implications. *Studies in African Linguistics*, 41(2):2–26.

Heinz, J. (2011). Computational phonology part II: Grammars, learning, and the future. *Language and Linguistics Compass*, 5(4):153–168.

Heinz, J. (2018). The computational nature of phonological generalizations. In Hyman, L. and Plank, F., editors, *Phonological Typology*, Phonetics and Phonology, chapter 5, pages 126–195. De Gruyter Mouton.

Heinz, J. and Lai, R. (2013). Vowel harmony and subsequentiality. In Kornai, A. and Kuhlmann, M., editors, *Proceedings of the 13th Meeting on the Mathematics of Language (MoL 13)*, pages 52–63, Sofia, Bulgaria.

Heinz, J., Rawal, C., and Tanner, H. G. (2011). Tier-based strictly local constraints for phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 58–64, Portland, USA. Association for Computational Linguistics.

Heinz, J. and Rawski, J. (forthcoming). History of phonology: Learnability. In Dresher, E. and van der Hulst, H., editors, *Oxford Handbook of the History of Phonology*, chapter 32. Oxford University Press.

Herbert, R. and Poppe, N. (1963). *Kirghiz manual*. Uralic and Altaic Series 33. Indiana University Publications, Bloomington.

Hyman, L. (1998). Positional prominence and the 'prosodic trough' in yaka. *Phonology*, 15:14–75.

Jardine, A. (2016). Computationally, tone is different. *Phonology*, 32(2):247–283.

Jardine, A., Chandlee, J., Eyraud, R., and Heinz, J. (2014). Very efficient learning of structured classes of subsequential functions from positive data. In Clark, A., Kanazawa, M., and Yoshinaka, R., editors, *Proceedings of the Twelfth International Conference on Grammatical Inference (ICGI 2014)*, volume 34, pages 94–108. JMLR: Workshop and Conference Proceedings.

Johnson, C. D. (1972). *Formal aspects of phonological description.* The Hague: Mouton.

Kaplan, R. and Kay, M. (1994a). Regular models of phonological rule systems. *Computational Linguistics*, 20(3):331–378.

Kaplan, R. M. and Kay, M. (1994b). Regular models of phonological rule systems. *Computational Linguistics*, 20(3):331–378.

Kaun, A. R. (1995). *The typology of rounding harmony: an optimality theoretic approach.* PhD thesis, UCLA.

Korn, D. (1969). Types of labial vowel harmony in the turkic languages. *Anthropological Linguistics*, 11:98–106.

Krueger, J. (1977). *Tuvan manual.* Indiana University Publications, Bloomington, Indiana.

Lai, R. (2015). Learnable vs. unlearnable harmony patterns. *Linguistic Inquiry*, 46(3):425–451.

Lambert, D., Rawski, J., and Heinz, J. (in progress). Typology emerges from simplicity in representations and learning. *Journal of Language Modelling.* Under review.

Lamont, A., O'Hara, C., and Smith, C. (2019). Weakly deterministic transformations are subregular. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 196–205.

Lesley-Neuman, D. (2012). Morpho-phonological levels and grammaticalization in karimojong: A review of the evidence. *Studies in African Linguistics*, 41(1):100.

McCollum, A. G., Baković, E., Mai, A., and Meinhardt, E. (2020). Unbounded circumambient patterns in segmental phonology. *Phonology.* to appear.

McCollum, A. G. and Essegbey, J. (2019). Initial prominence and progressive vowel harmony in tutrugbu. *Phonological Data and Analysis.*

McMullin, K., Aksënova, A., and De Santo, A. (2019). Learning phonotactic restrictions on multiple tiers. *Proceedings of the Society for Computation in Linguistics*, 2(1):377–378.

McMullin, K. and Hansson, G. Ó. (2019). Inductive learning of locality relations in segmental phonology. *Laboratory Phonology: Journal of the Association for Laboratory Phonology*, 10(1).

McMullin, K. J. (2016). *Tier-based locality in long-distance phonotactics?: learnability and typology.* PhD thesis, University of British Columbia.

Nanaev, K. (1950). *Uchebnik kirgizskogo yazika.* Kirgizgosizdat, Frunze.

Nelson, M., Dolatian, H., Rawski, J., and Prickett, B. (2020). Probing rnn encoder-decoder generalization of subregular functions using reduplication. *Proceedings of the Society for Computation in Linguistics (SCiL)*, pages 31–42.

Nevins, A. (2010). *Locality in Vowel Harmony.* The MIT Press, Cambridge, MA.

Novelli, B. (1985). *A grammar of the Karimojong language*, volume 7. Reimer.

Odden, D. (1994). Adjacency parameters in phonology. *Language*, 70(2):289–330.

Oncina, J., García, P., and Vidal, E. (1993). Learning subsequential transducers for pattern recognition tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:448–458.

Padgett, J. (1995). Partial class behavior and nasal place assimilation. In Suzuki, K. and Elzinga, D., editors, *Proceedings of the 1995 Southwestern Workshop on Optimality Theory.*

Poppe, N. (1960). *Buriat grammar. Uralic and Altaic series.* Indiana University, Bloomington.

Rawski, J. and Dolatian, H. (2020). Multi-input strict local functions for tonal phonology. *Proceedings of the Society for Computation in Linguistics*, 3(1):245–260.

Rogers, J., Heinz, J., Fero, M., Hurst, J., Lambert, D., and Wibel, S. (2013). Cognitive and sub-regular complexity. In Morrill, G. and Nederhof, M.-J., editors, *Formal Grammar*, volume 8036 of *Lecture Notes in Computer Science*, pages 90–108. Springer.

Rogers, J. and Pullum, G. K. (2011). Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information.*

Sasa, T. (2001). Yakut vowel harmony: an optimality theory account. *Turkic Languages*, 5:270–287.

Sasa, T. (2009). *Treatment of vowel harmony in optimality theory.* PhD thesis, University of Iowa.

Schröder, H. and Schroeder, H. (1987). Vowel harmony in toposa. *Afrikanistische Arbeitspapiere*, 12:27–35.

Skribnik, E. (2003). Buryat. In Janhunen, J., editor, *The Mongolic Languages*, chapter 5, pages 102–128. Routledge, London.

Smith, C. and O'Hara, C. (2019). Formal characterizations of true and false sour grapes. *Proceedings of the Society for Computation in Linguistics*, 2(1):338–341.

Svantesson, J.-O., Tsendina, A., Karlsson, A., and Franzen, V. (2005). *The Phonology of Mongolian.* Oxford University Press, Oxford.

van der Hulst, H. and Smith, N. (1987). Vowel harmony in khalkha and buriat (east mongolian). *Linguistics in the Netherlands*, pages 81–89.

Wilson, C. (2003). Analyzing unbounded spreading with constraints: marks, targets, and derivations. Unpublished manuscript, UCLA.

Wit, G. d. et al. (2015). *Liko phonology and grammar: a Bantu language of the Democratic Republic of the Congo*. LOT, Utrecht.