Proceedings of the Society for Computation in Linguistics

Volume 1 Article 8

2018

Formal Restrictions On Multiple Tiers

Alena Aksenova
Stony Brook University, alena.aksenova@stonybrook.edu

Sanket Deshmukh
Stony Brook University, sanket.deshmukh@stonybrook.edu

Formal Restrictions On Multiple Tiers

Alëna Aksënova and **Sanket Deshmukh**
Stony Brook University
{alena.aksenova,sanket.deshmukh}@stonybrook.edu

Abstract

In this paper, we use harmony systems with multiple feature spreadings as a litmus test for the possible configurations of items involved in certain dependence. The subregular language classes, and the class of tier-based strictly local (TSL) languages in particular, have shown themselves as a good fit for different aspects of natural language. It is also known that there are some patterns that cannot be captured by a single TSL grammar. However, no proposed limitations exist on tier alphabets of several cooperating TSL grammars. While theoretically possible relations among tier alphabets of several TSL grammars are containment, disjunction and intersection, the latter one appears to be unattested. Apart from presenting the typological overview, we discuss formal reasons that might explain such distribution.

1 Introduction

Recent investigations in the field of complexity of linguistic dependencies suggest that in different parts of language, well-formedness conditions are *subregular*, i.e. they do not require the full power of regular languages. For example, see (Heinz, 2010) for phonology, (Aksënova et al., 2016) for morphotactics, and (Graf and Heinz, 2015) for syntax among others.

A fruitful subregular class for natural languages is the class of *tier-based strictly local* (TSL) languages (Heinz et al., 2011). The core intuition behind this class is to capture long-distance dependencies locally by projecting elements relevant for a certain process on a tier, therefore "ignoring" all the intervening material that is irrelevant for this process. While the learner proposed in (Jardine and McMullin, 2017) is capable of inducing tier-based strictly local grammar in polynomial time using positive data only, there are numerous attested patterns that show that in some cases, one TSL grammar is not enough (McMullin, 2016). Extracting multiple cooperating grammars might become a problem if *any* type of relation is possible among *tier alphabets*, the sets of elements over which the TSL grammar operates.

In this paper, we explore possible relations among tier alphabets in natural languages, using harmonic systems with several spreadings as the litmus test. Theoretically possible relations between the two sets of harmonizing elements are containment ({a,b,c} and {a,b}), disjunction ({a,b} and {c,d}), and intersection ({a,b} and {b,c}). Here, we show that the latter case in unattested. Surprising as it may seem, this restriction actually reduces the amount of tier alphabet configurations. For example, for a set of 10 elements, there are 511 ways to form two disjoint sets, 1022 ways to arrange them with respect to the containment relation, and 27990 ways to form two sets with a non-empty intersection. The difference is striking: in this case, by removing the intersection relations, the amount of possible tier arrangements will be reduced by 95%.

The importance of eliminating possibilities that are not related to natural language and how it makes learning easier was highlighted by (Keenan and Stavi, 1986; Szymanik, 2016). For a domain with n elements, there are 2^{4^n} possible generalized quan-

Proceedings of the Society for Computation in Linguistics (SCiL) 2018, pages 64-73.
Salt Lake City, Utah, January 4-7, 2018

tifiers. However, when we take into account such property of natural language quantifiers as conservativity, it reduces the number of options to 2^{3^n}. For example, for a domain with 2 elements, there are 65536 possible generalized quantifiers, but only 64 of them are conservative.

The range of these topics recalls the "gavagai" problem (Quine, 1969): the learner of a language converges on a meaning for a word even though there are infinitely many possibilities to assign interpretation to this word. There, as well as in the case of inducing several tier alphabets, the successful learning is achieved by eliminating multiple theoretically possible assumptions.

We introduce the subregular class of tier-based strictly local languages in Sec. 2. Sec. 3 provides typological overview of different types of systems that exhibit several feature spreadings. In Sec. 4, we give a formal explanation of why it is efficient to eliminate the intersection relation from the scope of possible relations among tier alphabets. Sec. 5 concludes the paper.

2 TSL grammars

Tier-based Strictly Local (TSL) grammars (Heinz, 2011; Heinz et al., 2011) capture non-local dependencies by projecting selected elements on a *tier* in order to achieve locality among remotely dependent units. This allows us to analyze long-distance processes and rule out illicit sequences locally over the tier, because all the intervening irrelevant material is ignored. A TSL grammar consists of a tier alphabet T – set of items to be projected on a tier, and the set of n-grams G_{TSL} that must not be presented in a tier representation of a well-formed string.

For example, consider vowel harmony in LOKAA (Niger-Congo). In this language, a non-high vowel agrees with the preceding non-high vowel in ATR, whereas other vowels and consonants are transparent for the harmony, see (1-4) from (Akinlabi, 2009).

(1) **è**sìs**ò**n '*smoke*'
(2) **ɛ̀**sísɔ̀n '*housefly*'
(3) l**è**jìm**ə̀** '*matriclan*'
(4) **ɛ́**kílìk**à** '*kind of plant*'

The agreeing items are not adjacent to each other, that makes this process long-distance. For strings in (1–3), 5-grams are needed to capture this pattern, because there are 3 intervening elements in-between the two agreeing non-high vowels. But for (4), this window size is not enough: there are 5 segments in-between *ɛ* and *a*. In this language, there is no upper bound on the amount of material separating two non-high vowels that agree with respect to the [tense] feature; therefore, only projecting a tier of non-high vowels will allow to create the required locality relation among agreeing vowels.

Tier of non-high vowels **T = {ɛ, e, o, ə, ɔ, a}**	
1.	*[α tense] [β 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

In order to analyze this pattern with a TSL grammar, its tier alphabet T must include all non-high vowels presented in this language, and the ATR spreading is captured by blocking combinations of non-high vowels disagreeing in their [tense] specification, see H_{ATR} in Table 1. Figure 1 illustrates this analysis.

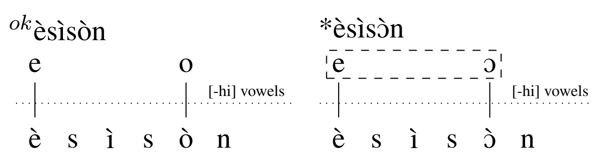

Figure 1: ATR harmony in LOKAA

The left subfigure shows the well-formed word *èsìsòn*. 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, thus the word *èsìsòn* is considered acceptable. However, its ill-formed counterpart **èsìsɔ̀n* contains two non-high vowels *è* and *ɔ̀* that disagree in [tense]. These vowels are projected, and the bigram **eɔ* is banned over the tier by the grammar H_{ATR}. Therefore the word **èsìsɔ̀n* must be ruled out.

The LOKAA harmony involved spreading of a single feature, and one tier was enough to capture the pattern. In the following section we will exemplify harmonic processes that involve multiple feature spreadings.

3 Types of multiple feature spreadings

In many languages, long-distance agreement processes involve spreading of more than one feature. The choice of items involved in a harmonic process, as well as of the harmonizing feature, varies a lot from language to language. For example, in many systems, vowel harmony in a feature such as backness (TURKISH, FINNISH) or tongue root position (MONGOLIAN, BURYAT) co-exists with labial assimilation, see (Kaun, 1995) for numerous examples of such vowel harmonies. Or it can be sibilant harmony in two features such as anteriority and voicing (NAVAJO, TUAREG). Also, in several languages it is possible to find both consonantal and vowel harmonies in features such as nasality and height (KIKONGO, KIYAKA, BUKUSU).

Further we show that in some cases, one TSL grammar is enough (Case 1) – it is possible to enforce both harmonic spreadings over a single tier. Another possibility is containment, and it is attested as well (Case 2) – there are languages in which one spreading affects a subset of items involved in another spreading. In some languages, harmonies affect two separate sets of segments, and the intersection of these two sets is empty (Case 3) – such tier alphabets are disjoint. And the only relation that appears to be typologically unattested is non-empty intersection (Case 4): to the best of our knowledge, there are no harmonies that affect two sets of elements that only partially overlap.

For the details and properties of the class of Multiple TSL (MTSL) languages, see (De Santo, 2017). We would like to highlight that this current work is preliminary, and the provided data and generalizations are drawn to the best of our knowledge.

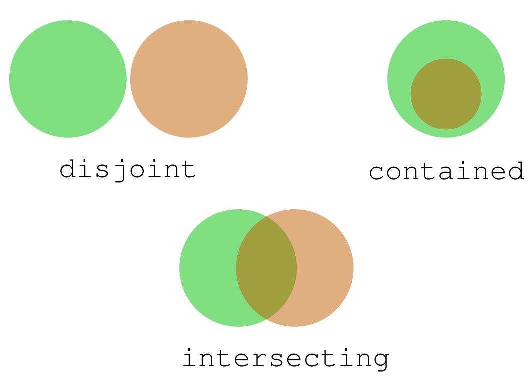

Figure 2: Theoretically possible tier alphabet relations

3.1 Case 1: single tier

Many harmonies with multiple feature spreadings can be captured with a single tier-based strictly local grammar. This 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.

Consider YAKUT (Turkic) as an example of such configuration. In this language, all vowels must agree in fronting. However, labial harmony spreads from low vowels onto both low and high ones, from high vowels to high ones, but it cannot spread from high vowels to low ones. The latter ones, in this case, function as *harmonizing blockers*: they inherit [round] specification from any preceding vowel, but block the rounding assimilation in [+high][–high] configuration, see (Sasa, 2001; Sasa, 2009).

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 examples (5-12) below from (Kaun, 1995).

(5)	**o**ɣ**o**-l**o**r	'child-PL'	***o**ɣ**o**-l**a**r
(6)	b**ö**r**ö**-l**ö**r	'wolf-PL'	*b**ö**r**ö**-l**e**r
(7)	**o**ɣ**o**-n**u**	'child-ACC'	***o**ɣ**o**-n**ɨ**
(8)	b**ö**r**ö**-n**ü**	'wolf-ACC'	*b**ö**r**ö**-n**i**
(9)	m**u**r**u**m-**u**	'nose-ACC'	*m**u**r**u**m-**ɨ**
(10)	t**ü**nn**ü**k-**ü**	'window-ACC'	*t**ü**nn**ü**k-**i**
(11)	**o**j**u**m-l**a**r	'shaman-PL'	***o**j**u**m-l**o**r
(12)	t**ü**nn**ü**k-l**e**r	'window-PL'	*t**ü**nn**ü**k-l**ö**r

Within a word, all vowels must share the same [tense] specification (5-12). High suffixal vowels agree with any preceding vowel in rounding (7-10), whereas low vowels can only inherit rounding feature from preceding low vowel (5,6), otherwise they are realized as non-rounded (11,12).

The tier alphabet T of TSL grammar that captures YAKUT pattern consists of all vowels presented in the language. H_{front} rules out sequences of vowels that disagree in fronting, whereas the part of the grammar responsible for the labial harmony ($H_{r1} \cup H_{r2} \cup H_{r3}$) 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 op-

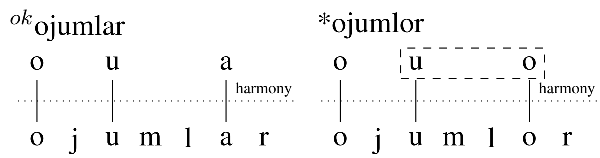

Figure 3: Fronting and labial harmony in YAKUT

erates over the tier alphabet T and its list of illicit substrings is $G_{TSL} = H_{front} \cup H_{r1} \cup H_{r2} \cup H_{r3}$.

	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.	*[+ high, α round] [+ high, β round]
	H_{r1} = {*uɨ, *üi, *iü, *ɨu, *iu, *ɨü, *ui, *üɨ}
3.	*[+ high, α round] [– high, + round]
	H_{r2} = {*üö, *uo, *iö, *ɨo, *io, *ɨö, *uö, *üö}
4.	*[– high, α round] [β round]
	H_{r3} = {*oa, *oɨ, *öi, *öe, *ao, *au,*eö, *eü, *aö, *aü, *eo, *eu, *oi, *oe, *öa, *öɨ}

Table 2: TSL grammar for YAKUT harmony

Figure 3 shows that such a grammar correctly predicts that the word *ojumlar* is well-formed with respect to the constructed TSL grammar, because the labial harmony spreads from the non-high vowel *o* to the following high vowel *u*. However, it cannot spread from a high vowel to a low one, therefore **ojumlor* is blocked as the illicit bigram **uo* is found on its vowel tier.

3.2 Case 2: tier and its sub-tier

Another possibility for the tier alphabets is to be in a set-subset relation. In this case, one harmony operates over a proper superset of items that are involved in another agreement.

In IMDLAWN TASHLHIYT[1] (Berber), affixal sibilants regressively harmonize with the stem in voicing and anteriority, see (Hansson, 2010b; McMullin, 2016). Whereas the anteriority harmony is not a subject for blockers of any kind, the voicing assimilation is blocked by any intervening voiceless obstruents. If there are no sibilants in the stem, the underspecified affixal element is realized as the voiceless anterior sibilant [s]. The data in (13-22) from (Elmedlaoui, 1995; Hansson, 2010a) illustrate the harmonic pattern using the causative prefix *s*-.

[1]The IMDLAWN TASHLHIYT generalization is presented here in a simplified way. Please refer to (McMullin, 2016) for the detailed description and discussion of the pattern.

(13) **s:**-uga ‘CAUS-evacuate’
(14) **s**-a**s:**twa ‘CAUS-settle’
(15) **ʃ**-fia**ʃ**r ‘CAUS-be.full.of.straw’
(16) **z**-bru**z:**a ‘CAUS-crumble’
(17) **ʒ**-m:**ʒ**dawl ‘CAUS-stumble’
(18) **s**-**ħ**u**z** ‘CAUS-annex’
(19) **s:**-u**kz** ‘CAUS-recognize’
(20) **sˤ**-rˤuˤ**fˤzˤ** ‘CAUS-appear.resistant’
(21) **s**-m**χ**a**z**aj ‘CAUS-loathe.each.other’
(22) **ʃ**-**qu**ʒ**:**i ‘CAUS-be.dislocated’

In (13), there are no sibilants in the root, so the prefix appears in its by-default 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] (14), [–voice, –ant] (15), [+voice, +ant] (16), and [+voice, –ant] (17). However, as mentioned before, 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 (18-22), stem-internal sibilants are voiced, but the ones in the prefix are voiceless, because of the intervening voiceless obstruents in-between them that block the agreement relation. Note that even if the voicing harmony is blocked, the anteriority one is still obeyed.

	Sibilant tier **T_{ant} = {s, z, ʃ, ʒ}**
1.	*[α ant] [β ant]
	H_{ant} = {*sʃ, *sʒ, *ʃs, *ʒs, *zʃ, *ʃz, *ʒz, *zʒ}
	Tier of sibilants and voiceless obstruents **T_{voice} = {s, z, ʃ, ʒ, ħ, k, f, χ, q}**
1.	*[+ cont, α voice] [+ cont, β 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 3: 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. This process is not local over a single tier – the locality required for the anteriority harmony cannot be achieved over a single tier, because both sibilants and voiceless obstruents are projected on the same tier.

The solution is to project two tiers. The first tier contains only sibilants (T_{ant}) and blocks their combinations that disagree in anteriority (H_{ant}): this tier enforces anteriority harmony. Both sibilants and voiceless obstruents must be projected on the second tier (T_{voice}), and the set of its illicit bigrams includes sibilants that disagree in anteriority (H_{v1}) and voiced sibilants followed by voiceless obstruents (H_{v2}). In this case, the second tier captures voicing assimilation.

Figure 4 illustrates this analysis. 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 over the voicing tier, the sibilants *ʃ* and *z* are not adjacent. The word **zukz* is also out, because the voicing grammar prohibits voiced sibilants followed by the voiceless obstruents (**zk*). Note that even though this word is ruled out, there are no violations over the anteriority tier: the voiceless obstruent *k* is not seen there.

IMDLAWN TASHLHIYT pattern requires two tiers, because the set of the elements affected by the anteriority assimilation is the proper subset of the one

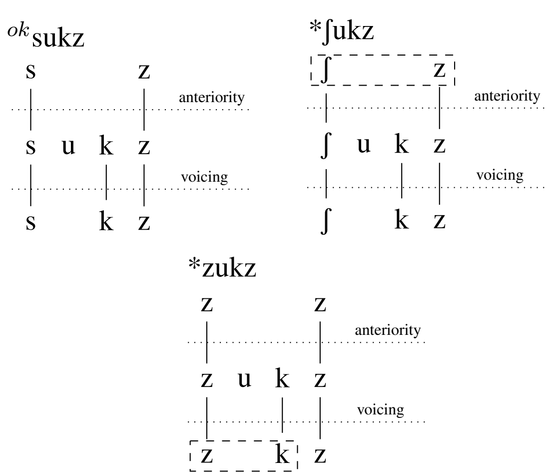

Figure 4: Sibilant harmony in IMDLAWN TASHLHIYT

taking part in the voicing harmony. One tier cannot provide the locality that is required in order to capture both spreadings.

3.3 Case 3: disjoint tiers

In some cases, two spreadings target absolutely different sets of elements: neither of the elements involved in one harmony takes a part in another agreement, and vice versa.

As an example of such a system, consider KIKONGO (Bantu). In this language, there are both consonant and vowel harmonies. Vowel harmony enforces vowels to agree in height, whereas nasal agreement turns both /d/ and /l/ into /n/ if preceded by a nasal in the stem, see (Ao, 1991; Hyman, 1998).

First, consider the height harmony that applies to vowel. In the examples below, the applicative suffix *-el, -il*, and the reversive transitive suffix *-ol, -ul* show that all vowels within a word must share the same height specification.

(23)	-s**o**mp-**e**l-	‘attach-APPL’
(24)	-l**e**ng-**e**l-	‘languish-APPL’
(25)	-t**o**mb-**o**l-	‘do-TRANS’
(26)	-l**e**mb-**o**l-	‘broom-TRANS’
(27)	-s**i**k-**i**l-	‘support-APPL’
(28)	-v**u**r-**i**l-	‘surpass-APPL’
(29)	-v**i**l-**u**l-	‘move-TRANS’
(30)	-b**u**b-**u**l-	‘bribe-TRANS’

In this language, suffixes are specified for rounding, and acquire their height specification depending on the stem vowel. In (23-26), both vowels in the stem and in the affix are non-high, whereas (27-30) contain only the high vowels.

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

Table 4: TSL grammar for KIKONGO vowel harmony

This harmony operates over the tier of vowels T_v, and the grammar must rule out all combinations of vowels that disagree in height, see Table 4.

But along with vowel harmony, this language also has a consonantal one – nasal agreement. Segments /d/ and /l/ in the affix both become /n/ if nasal consonants such as /m/ or /n/ are found in the root. See examples below from (Ao, 1991), where *-idi* is the perfective active suffix, and *-ulu* is its passive counterpart.

(31)	-suk-i**d**i-	'wash-PERF.ACT'
(32)	-**n**ik-i**n**i-	'ground-PERF.ACT'
(33)	-me**n**g-e**n**e-	'hate-PERF.ACT'
(34)	-suk-u**l**u-	'wash-PERF.PASS'
(35)	-**n**ik-u**n**u-	'ground-PERF.PASS'
(36)	-me**n**g-o**n**o-	'hate-PERF.PASS'

In (31, 34), there are no nasals in the 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/ assimilate to /n/, see (32, 35) and (33, 36) for *-idi-* and *-ulu-* respectively.

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

Table 5: TSL grammar for KIKONGO consonant harmony

Only /d/, /l/, and nasals are involved in the process, therefore those are the items that must be projected on the tier. Then the grammar H_n blocks occurrence of /d/ and /l/ after the nasals.

The two TSL grammars that capture vowel and consonantal harmonies 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. The tier alphabets are disjoint: their intersection is empty.

As the illustration, see Figure 5. Two tiers are necessary for the description of KIKONGO pattern, because only they can provide the needed locality relations among the vowels for vowel harmony, and /d/, /l/ and nasals for the nasal assimilation. The well-formed word *nikunu* is permitted because its vowel tier representation *iuu* does not violate the vowel harmony rule, and the nasal tier *nn* also satisfies the nasal assimilation. The ill-formed combinations of segments such as **io*, **uo*, and **nl* are ruled out by the two TSL grammars H_v and H_n, respectively. Note that the two vowels /i/ and /u/ are intervening between the two /n/ in the rightmost subfigure, and only the existence of the separate tier for the nasal harmony makes the two /n/ adjacent over the tier.

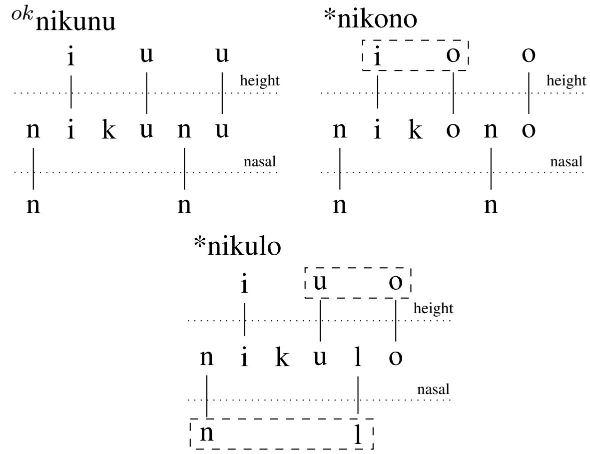

Figure 5: Vowel and nasal harmonies in KIKONGO

3.4 Case 4 (unattested): incomparable tiers

The following tier alphabet configurations were considered in this paper so far: single set (two harmonies operate over the same sets of elements), set-subset relation (one harmony operates over the proper subset of elements that are involved in another harmony), and disjoint sets (there is no item that is affected by both harmonies). The configuration that was not discussed yet is incomparable sets, i.e. a set in which the tiers are only partially overlapping. Going forward, such cases are unattested.

An example of such a system where the sets of segments that are involved in different harmonies will have non-empty intersection (excluding the proper subset case), would be the following. Imagine a pattern of a non-existent toy language YAKONGO that combines agreements from YAKUT and KIKONGO. Its alphabet includes *a, o, n*, and *d*. Vowels within a word agree in rounding, i.e. all of them are either /a/ or /o/, unless /n/ intervenes: only non-rounded vowels can follow /n/. The consonant /d/ assimilates to /n/ if it is preceded by /n/. Obviously, such pattern would require two TSL grammars, where the first one enforces the vowel harmony: T_v = {a, o, n}, H_v = {*ao, *oa, *no}. The second grammar captures the nasal assimilation: T_n = {n, d}, H_n = {*nd}. The intersection of the two tier alphabets is not empty and contains {n}.

However, to the best of our knowledge, there are no attested cases like this: if two TSL grammars are needed to capture two harmonies, their tier alphabets are either disjoint, or one is a proper subset of the other. This generalization might be surprising, but one of the possible reasons why it is the case is discussed in the following section.

4 Formal explanations of the typology

In this section, we are considering the problem of tier alphabet configurations from the formal point of view. Namely, we are discussing ways to partition sets in order to get each of the configurations discussed above. We show that if we consider all possible partitioning of a set into two subsets, then the vast majority of the resulting sets are incomparable, and it is exactly the configuration that seems to be absent from natural languages. Note that the partitioning considered here allows for replication, i.e. it allows for an item to be present in both sets obtained by partitioning of the initial set.

One of the reasons to think in this direction is related to learnability. It might be easier for a learner to converge on a particular hypothesis for the tiers if one does not need to consider all possible tier alphabet configurations. Eliminating the option of incomparable tier alphabets helps to remove the majority of guessing options from the set of hypotheses that a learner is considering. On a relevant note, (Keenan and Stavi, 1986; Szymanik, 2016) show that if we assume all possible generalized quantifiers, there are 2^{4^n} of them, where n is the size of the domain. However, if we take into account such property of all natural language quantifiers as conservativity (Barwise and Cooper, 1981), it reduces the amount of possible quantifiers to 2^{3^n}. For a domain of 2 elements, there are 65536 possible generalized quantifiers, but only 64 of them are conservative. The topic of tier alphabets and possible quantifiers share the same core idea: the importance of restricting the system in a way that natural languages restrict themselves.

The question that we are answering in the following subsections is the following: in how many ways it is possible to partition a set of n elements into 2 sets such that these sets will be in the set-subset relation, or disjoint, or incomparable.

Proper subset: if we have n elements in a set and we want to create a subset of k elements, this is equivalent to choosing k elements from a set of n, or $\binom{n}{k}$. Two of such subsets need to be excluded: $k = 0$, where one of the tier alphabets is empty, and $k = n$, where the two tier alphabets are equivalent. The amount of all other proper subsets is given by the following formula:

$$\sum_{k=1}^{n-1} \binom{n}{k} = 2^n - 2 \qquad (1)$$

For example, consider the set of 10 elements, i.e. $n = 10$. Then there are $2^{10} - 2 = 1022$ ways to form two sets that are in such containment relation.

Disjoint sets: the general case of partitioning a set of n elements into k disjoint subsets is given by Stirling Numbers of the Second Kind also denoted as $S(n, k)$, see (Knuth, 1968). It is evaluated as follows:

$$S(n,k) = \frac{1}{k!} \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} j^n \qquad (2)$$

If we want to partition the set of n elements into 2 disjoint sets, we can substitute the variable k in the expression (2) by 2, therefore getting the following formula:

$$S(n,2) = \frac{1}{2} \sum_{j=0}^{2} (-1)^{2-j} \binom{2}{j} j^n \qquad (3)$$

In this case, the number of partitions obtained from the set of 10 elements is 511, which is times less than the number of possibilities for the previous case.

Partition with intersection: in this case we want to partition a set of n elements into two sets with a non-empty intersection. This problem can be divided into two sub-problems: partitioning the set of n elements into 3 disjoint sets; and ordering the partitions to generate all possible intersections.

The solution to the first problem is the $S(n, 3)$, see (2) above. As for the second problem, let A_1, A_2 and A_3 be the three obtained partitions. Then we can create two sets with a non-empty intersection as follows: A_1A_2 and A_2A_3 where A_2 is the intersection, A_2A_1 and A_1A_3 where A_1 is the intersection, and A_1A_3 and A_3A_2 where A_3 is the intersection. Therefore for every partition, there are 3 combinations of sets that can be generated. The number of partitions given by $S(n, 3)$ needs to be multiplied by 3. The following expression calculates the number of 2 sets with incomparable intersection that can be obtained from a set with n elements:

$$3 * S(n, 3) = \frac{1}{2} \sum_{j=0}^{3} (-1)^{3-j} \binom{3}{j} j^n \tag{4}$$

For $n = 10$, this would give 27990 ways to create two sets with a non-empty intersection. This number is 95% more than the previous two combined.

Looking at the numbers of possible ways to partition a set of n elements, it is easy to notice that the biggest contribution is always made by the sets with a non-empty intersection. This fact makes us suspect that the absence of such tier alphabet configuration is due to the limitation on the computational processes: much less options need to be considered when such limit is established.

In order to illustrate the growth, consider Figures 6 and 7 below. Figure 6 shows the normal scale of growth of the amount of partitions. The green dashed line shows the disjoint partitions, the blue dotted line represents the partitions with set-subset relation, and the solid red line is representing exponentially growing number of incomparable partitions. If the number of elements in the initial set is larger than 10, the two lowest lines become nearly indistinguishable, therefore for bigger numbers it is better to consider the growth on a loglog scale, see Figure 7.

5 Conclusion

In this paper, we studied various harmonic processes involving transmission of multiple features, and used such systems as a litmus test for detecting possible tier alphabet configurations. We found out that there are 3 typologically attested cases, namely: single tier, when both harmonies operate over the same set of elements, tier containment, where one harmony operates over the proper subset of items that are involved in another assimilation, and disjoint tiers, where no the items involved in one harmony are relevant for the other one. The fourth possibility, being incomparable tier alphabets, is unattested to the best of our knowledge.

Although it might seem unexpected, in fact this restriction limits the amount of possible tier configurations a lot, as it is shown in Sec. 4. For a set of 10 elements, this limitation excludes 95% of all possible tier alphabet organizations. With the increasing number of elements in the set of items relevant for harmonic processes, this percentage grows as well.

This is just preliminary research about the typology of long-distance processes and the math behind it, and, of course, a lot is still remained unexplored. For example, here we are investigating harmonic processes, but these generalization must be checked on a variety of dissimilation processes, see (Bennett, 2013). Another route will be to investigate the

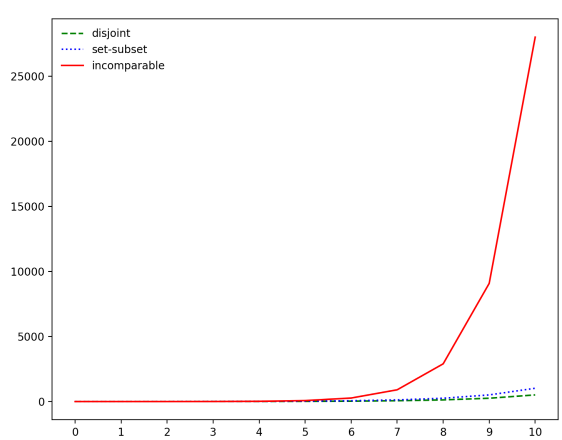

Figure 6: Growth of number of partitions of sets containing up to 10 elements (normal scale)

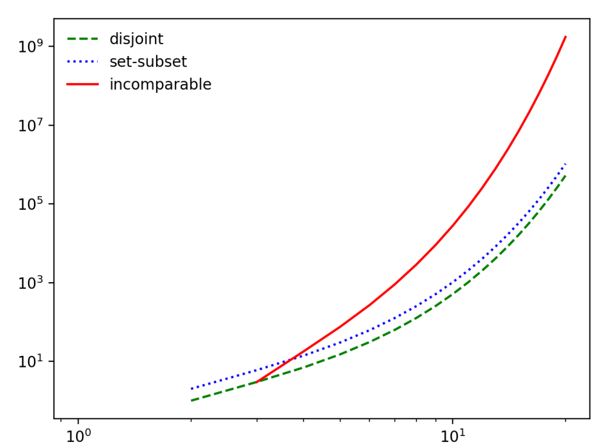

Figure 7: Growth of number of partitions of sets containing up to 20 elements (loglog scale)

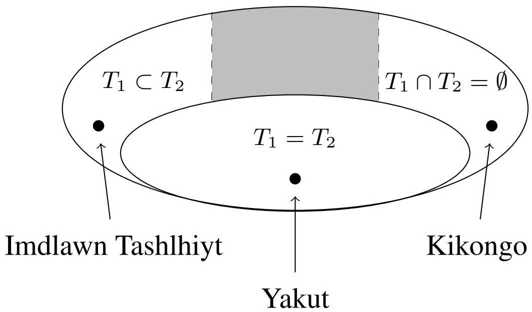

Figure 8: Attested tier alphabets relations

size n of tier alphabets that is relevant for natural languages, and check which tier alphabet configurations are available for each range of n. And, of course, more careful typological overview is needed.

However, this result can be interesting from several different perspectives. First, it reveals new typological generalization about harmonic systems and natural languages in general. Secondly, it might shed light on the issues related to the learnability of multiple tier-based strictly local grammars. And, lastly, it brings the desired naturalness to the theory of formal languages.

Acknowledgments

We thank the anonymous referees for their useful comments and suggestions. We are very grateful to our friends and colleagues at Stony Brook University, especially to Thomas Graf, Lori Repetti, Jeffrey Heinz, and Aniello De Santo for their unlimited knowledge and constant help. Also big thanks to Gary Mar, Jonathan Rawski, Sedigheh Moradi, and Yaobin Liu for valuable comments on the paper. All mistakes, of course, are our own.

References

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

Alëna Aksënova, Thomas Graf, and Sedigheh Moradi. 2016. Morphotactics as tier-based strictly local dependencies. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 121–130.

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

Jon Barwise and Robin Cooper. 1981. Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4:159–219.

William G. Bennett. 2013. *Dissimilation, Consonant Harmony, and Surface Correspondence*. Ph.D. thesis, Rutgers University.

Aniello De Santo. 2017. Pushing the boundaries of tsl languages. Manuscript. Stony Brook University.

Mohamed Elmedlaoui. 1995. *Aspects des représentations phonologiques dans certaines langues chamito-sémitiques*. Ph.D. thesis, Université Mohammed V.

Thomas Graf and Jeffrey Heinz. 2015. Commonality in disparity: The computational view of syntax and phonology. Slides of a talk given at GLOW 2015, April 18, Paris, France.

Gunnar Olafur Hansson. 2010a. *Consonant Harmony: Long-Distance Interaction in Phonology*. University of California Press, Los Angeles.

Gunnar Olafur Hansson. 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.

Jeffrey Heinz, Chetan Rawal, and Herbert G. Tanner. 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.

Jeffrey Heinz. 2010. Learning long-distance phonotactics. *Linguistic Inquiry*, 41(4):623–661.

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

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

Adam Jardine and Kevin McMullin. 2017. Efficient learning of tier-based strictly k-local languages. *Lecture Notes in Computer Science*, 10168:64–76.

Abigail Rhoades Kaun. 1995. *The typology of rounding harmony: an optimality theoretic approach.* Ph.D. thesis, UCLA.

Edward L. Keenan and Jonathan Stavi. 1986. A semantic characterization of natural language determiners. *Linguistics and Philosophy*, 9:253–326.

Donald E. Knuth. 1968. *Fundamental Algorithms*. Addison-Wesley, Reading, MA.

Kevin James McMullin. 2016. *Tier-based locality in long-distance phonotactics: learnability and typology*. Ph.D. thesis, University of British Columbia.

Willard O. Quine. 1969. Ontological relativity. In *Ontological relativity and other essays*. Columbia University Press, New York.

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

Tomomasa Sasa. 2009. *Treatment of vowel harmony in optimality theory*. Ph.D. thesis, University of Iowa.

Jakub Szymanik. 2016. *Quantifiers and Cognition: Logical and Computational Perspectives*. Springer, Switzerland.