tuesday – September 10, 2024

Sub-regular phonology

1 Where do phonological patterns sit in the Chomsky Hierarchy?

A bit of the history

(take in strings, output strings)

1. Kaplan & Kay (1994)[1] show that finite-state transducers can model any mapping generated by a phonological rule system.
2. Subsequent work supports the idea that phonology is regular. [2]

But are we done?

1.1 Expressiveness and restrictiveness

Linguistic models (and models in general) are often evaluated against two particular criteria: expressiveness and restrictiveness:

Expressiveness: does the model generate all the things we need it to generate?

- e.g., can FSAs/FSTs generate all the kinds of patterns we see in the phonologies of the world's languages?

Restrictiveness: does the model generate only the things we need to generate?

- e.g., do FSAs/FSTs generate patterns that are never attested (documented) in the phonologies of the world's languages?

FSAs/FSTs have been argued to be sufficiently expressive as models of phonology, but are not sufficiently restrictive.

For example, both of the following phonologically-bizarre languages are regular:

- A language where every word must have an even number of vowels.
- A language where every word must have exactly three vowels.

) not human languages (too bizarre)

[1] Kaplan, R.M. & Kay, M. (1994). Regular models of phonological rule systems. *Computational Linguistics*, 20, 331–378.
[2] Suprasegmental phenomena, such as tone patterns, are often claimed to be more complex.

Final exam practice break:
Assume $\Sigma = \{C, V\}$

1. Design an FSA that accepts only those words with an even number of vowels.
2. Design an FSA that accepts only those words with exactly three vowels.

We'd like our models of phonology to avoid generating patterns like these, while still generating patterns that we do see. For this, we need a more restrictive formalism.

1.2 Learnability

Another property of regular languages that is often cited as motivation for exploring the subregular hierarchy is that they are not learnable in the limit from positive data (Gold, 1967).

Learning from positive data:

Imagine you are taking an exam for this class. I sit you down and read you a list of strings that are members of some regular language. You know that the language is regular, but you don't know its grammar. You pass the exam if you correctly tell me the grammar. The actual process goes like this:

1. I read you a single string from the language.
2. You guess what the grammar is, based on all the strings you've heard so far.
3. We repeat steps 1 and 2 until you produce the correct grammar.

If you eventually pass the final exam, it means that you've successfully learned this *language in the limit*. The *positive data* part refers to the fact that I'm not giving you any examples of strings that *aren't* in the language.

Unfortunately, regular languages are in general not learnable in this way: i.e., you are not guaranteed to ever converge upon the correct grammar.

It's important to take this argumentation with a grain of salt, since this is not intended as a model of human learning. Among other things, human learners probably do receive some amount of negative evidence.

Consider a hypothetical 3-year old child. A common mistake that English-learning children make at this age is saying "go-ed" instead of "went." (overgeneralization)

1. The adults will never say "go-ed" no matter how many times the child says it.
2. The adults may laugh or even explicitly correct the child when they make this mistake.

However, some people have also argued that this kind negative evidence is not necessary for language learning (Marcus 1993)[3], because ...

1. if noisy feedback exists it is too weak
2. no kind of noisy feedback is systematically provided to all children at all ages for all types of errors

So still, learnability should be something we consider carefully when we create our models.

[3]Marcus, G. F. (1993). Negative evidence in language acquisition. *Cognition*, 46(1), 53-85.

1.3 What parts of phonology do we want to model?

There are two broad types of phonological processes: **phonotactics** and **transformations**.

- **phonotactics**: what is a valid phonological form?

  Consider the following nonce words:

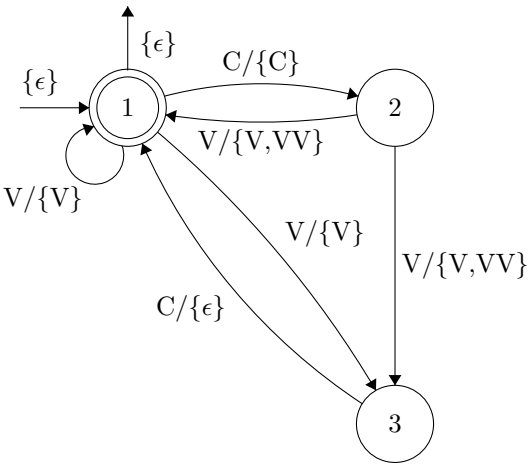
/wɪs/	/tip/
/zeɪps/	/ʃwuʒ/
/θeɪpt/	/pwʌdz/
/vlas/	/ptak/

- **Transformations**: how do we map from one phonological form(i.e., an underlying representation) to another one (i.e., a surface representation)?

  We talked about transformations in the setting of Semiring-valued FSAs: the output string semiring, also known as a finite-state transducer (FST), uses concatenation along the path and union to combine different paths.

  The following FST shows the effect of optionally lengthening vowels that follow an onset, and deleting all codas.

  (1)



We'll restrict ourselves here to considering just phonotactics: what forms could plausibly be a word in a particular language?

- Modeling phonotactics requires fewer assumptions than modeling mappings from underlying to surface forms.
- Fits better with the material we've covered in this class.

You can find a good overview of transformations and associated formalisms in Heinz (2018)[4].

[4]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.Mouton De Gruyter.

2 The subregular hierarchy

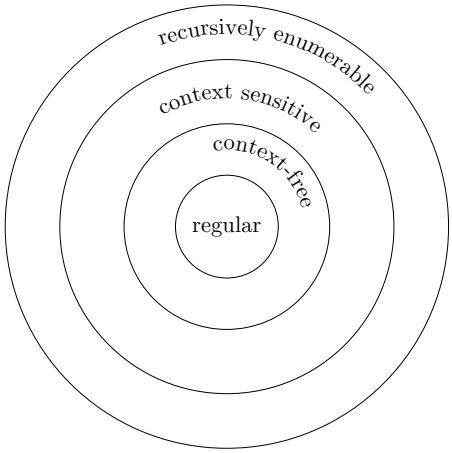
Recall the Chomsky Hierarchy:

(2)

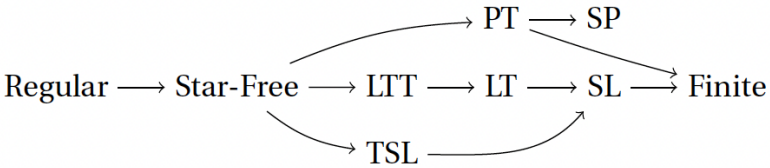
Grammar	Language	Automaton
Type-0	Unrestricted/recursively enumerable	Turing machine
Type-1	Context-sensitive	Linear-bounded automaton
Type-2	Context-free	Nondeterministic pushdown automaton
Type-3	Regular	Finite-state automaton

Chomsky Hierarchy: regular ⊆ context-free ⊆ context-sensitive ⊆ unrestricted

(3)



Where do natural language phonotactics sit?



We won't discuss all of these today, but we'll go through a few that are more commonly [5] applied to natural language phonology. Let's start from the bottom and work our way up.

2.1 Finite languages

Finite languages are languages containing only a finite number of strings. All of the grammars we've discussed in the class have been capable of generating or accepting infinite sets of strings. There are two reasons for this:

1. Natural languages do not have a finite number of possible words/utterances. [6]
2. Finite languages are not theoretically very interesting.

We can define a finite language by simply enumerating the set of all its strings. We can determine if a string is in the language by checking if it is in this set.

- There's no notion of a grammar here!

Just check if string is in set

[5] See Rogers and Pullum(2011); Rogers et al. (2013); Heinz (2018) for good overviews.
[6] "Language makes infinite use of finite means" –Wilhelm von Humboldt

2.2 Strictly local grammars

2-Strictly local

Formally, a (2)-SLG is a four-tuple (Σ, I, F, T), where

- Σ is the alphabet of symbols,
- I is a subset of Σ, specifying the allowable starting symbols,
- F is a subset of Σ, specifying the allowable final symbols, and
- T is a subset of $\Sigma \times \Sigma$, i.e. a set of pairs of symbols, specifying the allowable two-symbol sequences (or allowable "bigrams").

Notice a critical difference between SLGs and FSAs: SLGs have no states!

Practice: Define a 2-SL grammar over $\Sigma = \{a, b\}$ that generates the language $ab(ab)^*$.

$k-$Strictly local

Note that the definition defines a 2-strictly local grammar, because it keeps allowable *bigrams*. We could define a more generalized $k-$strictly local grammar, which keeps all allowable k-grams.

↪ T is a subset of $\Sigma \times \Sigma \times \Sigma \ldots \times \Sigma$.

A language is said to be **Strictly Local**(SL) if and only if there is some k for which it is $k-$strictly local.

The probabilistic version of $n-$Strictly local is known as an *n-grams language model*; see Chapter 3 in Jurafsky & Martin (2023). [7]

An important property of SL languages is that they are *closed under suffix substitution.*

> A stringset L is Strictly local if and only if there is some k such that whenever there is a string x of length k-1 and strings u_1, v_1, u_2, and v_2, such that $u_1xv_1 \in L$ and $u_2xv_2 \in L$, then $u_1xv_2 \in L$ as well.

Are k-SL grammars sufficient for phonology?

Many phonotactic restrictions are local, and can be modeled using SL grammars. However, there are a number of long-distance phonotactic restrictions that cannot be captured in a general way by SL grammars.

These patterns are often modelled with autosegmental phonology, and include things like:

- Vowel harmony
- Consonant harmony
- Certain tone and stress patterns

[7] Jurafsky, D. & Martin, J (2023). *Speech and Language Processing.* 3rd ed. draft

An example of this is Aari sibilant anteriority harmony, described by Hayward (1990).

UR	SR	Gloss
/baʔ-s-e/	[baʔse]	'he brought'
/ʃed-er-s-it/	[ʃederʃit]	'I was seen'
	*[ʃedersit]	
/ʒaːg-er-s-e/	[ʒaːgerʃe]	'it was sewn'
	*[ʒaːgerse]	

- the sibilants have to agree on their [anterior] feature (s: [+anterior]; ʃ: [-anterior])
- no occurrence of both [s] and [ʃ], no matter how far these two phones are apart.

Question: Why can't this pattern be captured by a k-SL grammar?

2.3 Strictly Piecewise languages (SP)

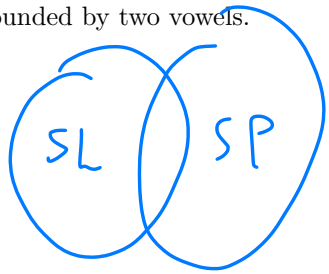
Strictly k-local grammars keep allowable k-grams. The symbols in the k-grams have to immediately follow their preceding symbols in the string.

Strictly k-piecewise grammars keep the set of subsequences up to some length k. The symbols could follow its preceding symbol at any distance in the string.

Exercise: Let us design a 2-Strictly Piecewise language for Aari sibilant harmony.

The relationship between SLs and SPs:

- There's a Strictly-Piecewise language that is not Strictly-Local
  - The Aari case and other long-distance dependencies
- There's a Strictly-Local language that is not Strictly-Piecewise.
  - ... because of the strong locality restriction in SLs
  - Think of the intervocalic voicing requirement: a stop must be voiced when it's **immediately** surrounded by two vowels.



2.4 Tier-based Strictly local grammars

Tier-based k-strictly local (k-TSL) grammars (Heinz et al., 2011) are similar to k-SL grammars, except that we erase certain symbols from the string before checking for prohibited substrings.

Intuitively, Tier-based Strictly Local grammars are like Strictly-Local grammars, specifying allowable (or forbidden) sub-strings, but target a "tier" which only contains some subset of the segments in a language.

We can think of Tier-based strictly k-local grammars as a five-tuple: $(\Sigma, T, I, F, \delta)$

- Σ is the alphabet of input symbols
- T is a subset of Σ, specifying what symbols should be included in a tier
- I is a subset of T, specifying the allowable starting symbols in the tier
- F is a subset of T, specifying the allowable final symbols in the tier, and
- δ is the set of allowable k-grams on the tier for some k

Practice: Let's design a 2-TSL language for Aari sibilant harmony.

The relationship between TSLs and SLs: TSLs properly includes SLs:

- The Aari case can be captured by a TSL: have a tier that only contains sibilants
- All SL laguages can be captured by a TSL: all symbols belong to one tier.

The relationship between TSLs and SPs: SP languages cannot be directly compared to TSLs.

- There's a Strictly Piecewise language that is not Tier-based Strictly Local
  - A low tone (L) never appears between two high tones (H), no matter how far the two high tones are apart.
- There's a Tier-based Strictly Local language that is not Strictly Piecewise
  - A word must have one and only one stressed syllable, and the stressed syllable can occur at any position (could be very far from the word edges)

Are TSL grammars enough for phonology?

TSLs are a desirable upper bound for phonological complexity. It is expressive...

- Captures long distance patterns, like Aari anteriority harmony.

But it is also restrictive...

- For example, it cannot generate languages that have an even number of vowels.

There are a few other nice properties of TSL languages.

Learnability There are algorithms that allow TSL languages to be learned in polynomial time from positive data.

This means that a learner using this algorithm is guaranteed to recover the correct [8] TSL grammar in a number of steps that is some polynomial function of the size of the input.

Psychological reality: Artificial grammar learning An artificial grammar learning (AGL) study is where we teach people simple languages in a laboratory setting, and then test how well they have learned them.

- e.g., First, I show you words from an artificial language that has vowel harmony until I think you've 'learned' the language sufficiently.
- Next, I present you with words you haven't seen before, and ask if they could be part of this language.

McMullin (2016)[9] and McMullin and Ólafur Hansson (2016)[10] show that:

- Participants are able to learn 2-TSL patterns effectively.
- When participants are given more complex patterns, they often learn 2-TSL patterns that approximate the input data.

So TSL grammars seem to be both computationally and experimentally good approximations of phonotactic complexity. The weak subregular hypothesis suggests that TSL grammars are sufficient to capture all of human phonological knowledge (Heinz, 2018).

2.5 Non-TSL segmental phonology

Although TSL appears to be sufficient to generate almost all segmental phonological processes, there are a few that have been claimed to be beyond TSL.

- Tamashek Tuareg and Imdlawn Tashlhiyt sibilant harmony (McMullin, 2016)
- Uyghur vowel harmony (Mayer and Major, 2018) [11]
- Sanskrit n-retroflexion (Graf andMayer, 2018) [12]

All of these can be captured by other subregular languages.

[8] See Jardine, A. and Heinz, J. (2016). Learning tier-based strictly 2-local languages. *Transactions of the ACL*, 4:87–98.; Jardine, A. and McMullin, K. (2017). Efficient learning of tier-based strictly k-local languages. In Drewes, F.,Martín-Vide, C., and Truthe, B., editors, *Language and Automata Theory and Applications*, 11th International Conference, pages 64–76. Springer.

[9] McMullin, K. (2016). *Tier-based locality in long-distance phonotactics: Learnability and typology*. PhD thesis, University of British Columbia.

[10] McMullin, K. and Ólafur Hansson, G. (2016). Long-distance phonotactics as tier-based strictly 2-local languages. *Proceedings of the Annual Meetings on Phonology 2014*.

[11] Mayer, C. and Major, T. (2018). A challenge for tier-based strict locality from uyghur backness harmony. In Foret, A., Kobele, G., and Pogodalla, S., editors, *Formal Grammar 2018*. Springer, Berlin, Heidelberg.

[12] Graf, T. and Mayer, C. (2018). Sanskrit n-retroflexion is input-output tier-based strictly local. In *Proceedings of the 15th SIGMORPHONWorkshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 151–160. The Special Interest Group on Computational Morphology and Phonology, Brussels, Belgium.

2.6 Non-regular segmental phonology?

Some non-regular segmental phonological patterns are predicted by our models (e.g., optimality theory) but are unattested or marginally attested. One such class of patterns is majority rules vowel harmony.

Majority rules vowel harmony: The number of input vowels of different types (e.g., front vs. back) is counted, and all vowels in the output harmonize to the majority type.

A putative example of this comes from Warlpiri (Bowler and Zymet, 2019) [13].

- Warlpiri is a Pama-Nyungan language spoken in Australia by around 3000 people.
- It has three vowels: [i a u]
- The high vowels [i u] harmonize with one another, generally across morpheme boundaries.

Disharmonic stems in Warlpiri appear to be in the process of regularizing to harmonic forms. Many of these exhibit free variation with harmonic forms. The harmonic forms appear to display majority rules harmony based on the number of vowels in the disharmonic form!

2 u's vs 1 i

kulkuri ∼ kulkuru	'middle'	ɟuwiɲti ∼ ɟiwiɲti	'wild currant"
puluki ∼ puluku	'calf'	warupiɲi ∼ waripiɲi	'grass seed'
puruɽuɟinpa ∼ puruɽuɟunpa	'beetle'	munijiki ∼ minijiki	'honey bee'

Thinking question: What about this pattern makes it non-regular?

No FSA can be constructed

We'll come back to whether we should be worried about these cases below.

not needed

3 What are we modeling when we do formal language phonology?

Computational models of phonology are different than models of syntax, because we have more access to substantive explanations for patterns we see: that is, we produce and perceive speech using our bodies, and our bodies constrain speech in many ways.

Phonological patterns might arise for a few reasons, e.g.:

- It's hard to produce a particular sound in a particular context.
- It's easy to misperceive a particular sound in a particular context.
- etc.

Formal language phonology generally abstracts away from these details, but this is not meant to imply that these details are unimportant! Instead, it approaches phonological system from a higher level. This is reminiscent of phonological conspiracies as proposed by Kisseberth (1970). [14]

[13] Bowler, M. and Zymet, J. (2019). A count effect in Walpiri vowel harmony. Poster presented at the 27th Manchester Phonology Meeting.

[14] Kisseberth, C. (1970). The functional unity of phonological rules. *Linguistic Inquiry*.

3.1 Phonological conspiracies

Yokuts Yawelmani has a general ban on CCC sequences in surface forms.

When the morphology would produce a CCC sequence, there are a few different repair strategies that are deployed:

$\emptyset \rightarrow$ V / C __CC	vowel epenthesis
C $\rightarrow \emptyset /$ CC+__	consonant deletion

These rules have something in common (they both conspire to prevent CCC clusters), but rule-based phonology has no way to capture this commonality. This was part of the motivation behind adopting Optimality Theory, which explicitly makes such constraints part of the grammar.

3.2 Computational conspiracies

Formal language models of phonology do something similar, in that they abstract away from the substance of phonology to provide a characterization of its overall complexity.

Why is this useful?

- It provides precise, quantifiable bounds on the generative power of our theories.
  - Constraint interaction in optimality theory can generate non-regular transductions, even if all individual constraints are regular (or SL or TSL).
- It allows us to factor a complex system into formal and substantive components.
  - Breaking down complex systems into manageable parts helps us study them effectively.
- Provides new, testable predictions that may not be apparent from a substantive perspective.

We should not in general interpret formal language phonology as claiming that phonology patterns arise from computational limitations on humans.

- i.e., phonology is probably mostly TSL because of the properties of the processes that create and shape phonologies and how they interact, not because people can't learn more complex patterns.

3.3 Should we be worried about supra-TSL patterns?

We need to carefully investigate the data! First, we should be skeptical of our data.

- Do we actually have a correct understanding of the pattern?
- If so, do people learn it productively? Does it seem diachronically stable? That is, the system does not break down after a long period of times.

If the answer to both these questions is yes, then we should be sure that our models predict the uncommonness of such patterns, rather than necessarily excluding them completely.