Class 20, 6/4/2020: Mathematical Phonology; Course Summary

1. Assignments

- Read:
  - Heinz, Jeffrey, and William Idsardi (2013) What complexity differences reveal about domains in language. *Topics in cognitive science* 5: 111-131.
  - On course web site.
- Feel free to discuss progress/problems with your term project.
  - Friday office hours moved to 4, not 2.
  - Office hours continue through Finals Week
- Handout-session with me before the middle of Finals Week, Wed. 6/10.
- Term paper due Mon. after Finals Week, 6/15.

2. Extensions and incompletes

- In light of current events I am being very flexible about giving you extensions on assignments, or an Incomplete in the course.
- I am available this summer to do Ling. 219 appointments and grade your work.
- Please let me know what is needed.

MATHEMATICAL PHONOLOGY

3. About mathematics in general

- All math is built with a series of inverted pyramids that use proof to deduce ever more consequences from a set of simple assumptions (postulates). Euclid showed the way.
- Progress in mathematics follows its own logic and cannot be channeled easily, no matter how much we would like to do this.
- A wonderful skill is to penetrate the mathematical edifice and find results that could be useful to science, particularly to linguistics.
  - Only a few have this gift; most live in one world or the other.

4. The math addressed in this class: formal language theory

- A central concept is the **Chomsky Hierarchy**
- Let a language be modeled as a **set of strings**, formed from an **alphabet** of symbols.
- Then we can use math to define a **set of languages** (set of sets)
- Then we build a **hierarchy** of sets of sets, each of which **properly contains** all the languages on the next level down.

- This apparatus has given rise to a **rich set of theorems**, some of them proven by Noam Chomsky in his youth, but also a flourishing enterprise taking place largely in computer science departments.
- Startlingly, Hopcroft et al. in their text suggest that by now all the good theorems have by now been proven:
    - "in 1979 [previous edition] automata and language theory was still an area of active research. A purpose of that book was to encourage mathematically inclined students to make new contributions to the field. Today there is little direct research in automata theory as opposed to its applications."
    - We are lucky to be in a field where every line of research remains open!

5. Where to learn formal language theory like a pro

- That is, as a computer science student would learn it, not in distorted/watered down version for linguists. This can actually be very comforting!
- The textbook by Hopcroft and Ullman (1979) *Introduction to Automata Theory, Languages, and Computation* is acclaimed, and I certainly like it — very direct and focused, though not easy.
- I learn on Wikipedia that later editions, with coauthor Motwani, have more examples and are easier.

6. Another widely used text

- Jurafsky, Dan and James Martin (in progress; to become the 3rd ed.) *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*
- Grab the searchable on-line copy while you can.

7. Yet another resource

- Tim Hunter periodically teaches a beautiful course that lays out all the levels of the Chomsky hierarchy, with relevant theorems and algorithms, including the probabilistic versions.

A QUICK TOUR OF THE CHOMSKY HIERARCHY

8. Caveat

- I'm doing my best here and am doubtless making errors.

9. The very bottom

- Finite lists

10. Next up: languages defined by N-gram models

- bigram: the possibility of generating *x* as the next symbol is determined solely by the current symbol
- trigram: the possibility of generating *x* as the next symbol is determined solely by the current symbol and its predecessor
- and so on for *n* = 4, 5, …
- You can do this with charts. Here is one bigram model:

If you just heard this	you may move on to any of these
[	t, k
t	a, i
k	a, i
a	t, k,]
i	t, k,]

- ☞ What is the language generated?
- This can easily be made probabilistic; e.g. in Row 1 pick from {t, k} at random according to their relative frequency.
- There are refinements to handle zero-frequency cases; see Jurafsky and Martin.

11. Uses of n-gram models

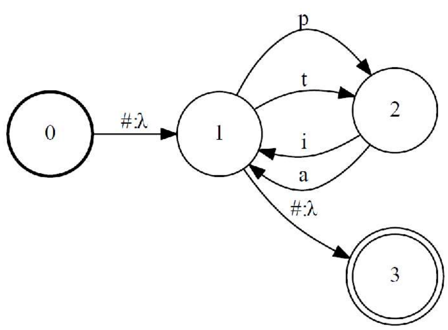
- A quick-and-unprincipled account of **phonotactics**, assigning probability to any string
  - ➢ See this ref. for a comparison of trigram models with more linguistically-informed models:
  - ➢ (2011) Robert Daland, Bruce Hayes, James White, Marc Garellek, Andreas Davis, and Ingrid Normann. Explaining sonority projection effects. *Phonology* 28:197–234. 28: 197–234.
- Stuff meant (?)for entertainment, like mimicking Shakespeare with *n*-gram models with increasing *n.* From Jurafsky and Martin's text:

1 gram	–To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have –Hill he late speaks; or! a more to leg less first you enter
2 gram	–Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow. –What means, sir. I confess she? then all sorts, he is trim, captain.
3 gram	–Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, 'tis done. –This shall forbid it should be branded, if renown made it empty.
4 gram	–King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv'd in; –It cannot be but so.

12. Regular languages

- There is an official definition of this, but let us skip it.
- Far easier to use the theorem (Hopcroft and Ullman 1979 29-35) that tells us that a regular language is one that can be generated by a **finite-state machine**.
- A finite-state machine is a cute object with arcs and states (ovals), that *accepts* a class of strings — a regular language.

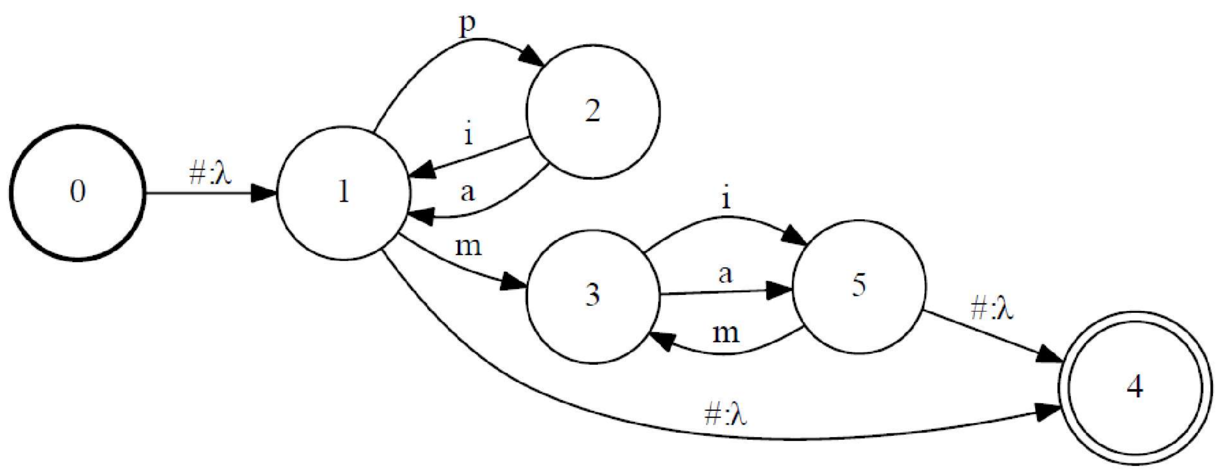
13. One example from Hayes and Jo (in resubmission)



- bold circle is a start state
- double circle is a terminal state
- ignore the #, just for starting and ending
- If you can digest a string by traveling along the arcs, than that string is part of the language.
- ☞ Characterize the language in words.

14. Regular languages are a superset of n-gram languages

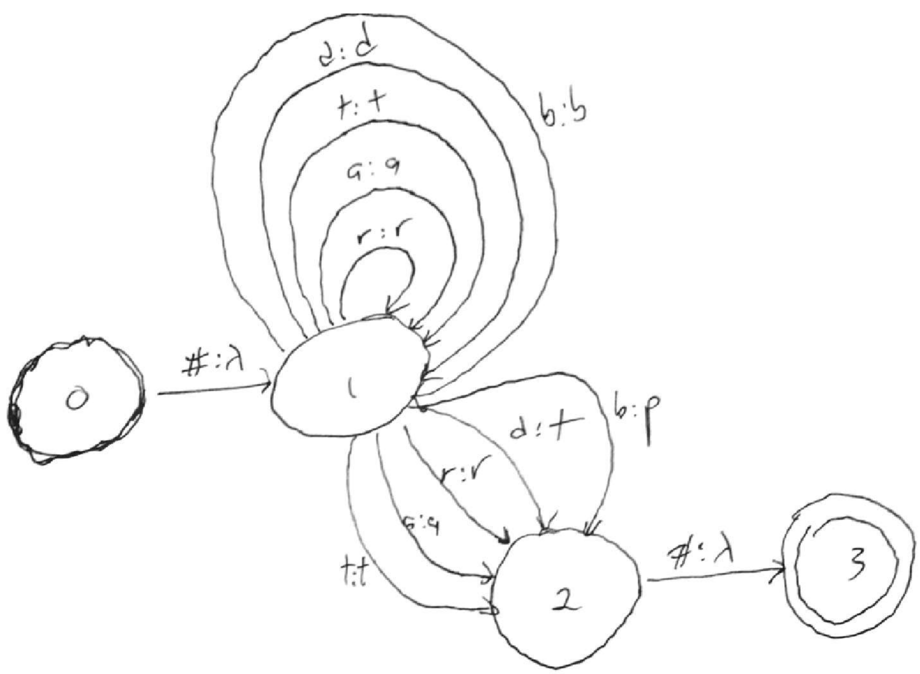
- It's like the chart in (10), but you can have multiple rows for the same symbol, *depending on how you got there*.
- So it has a stronger sense of "history".
- Show this by telling me in prose what language the finite state machine below recognizes.



- Observe that there are two states, 1 and 5, that say "I have just recognized an [i] or [a]", and they crucially are different.

15. Finite-state transducers

- Let the "symbols" of the alphabet be **changes**, like d:t, b:p, t:∅.
- Now the finite state machine is a finite state **transducer**, formalizing a (perhaps *SPE*-style) rule.
- Here is a finite-state transducer that can do Final Devoicing on a language with /d/ and /b/.
    - ➢ Note the characteristic ducking of the problem of natural classes.



16. Next up on the hierarchy: *context-free languages*

- The apparatus here is very familiar: the phrase structure grammars we teach to beginners are the sort of grammars that generate context free languages.

  S → NP VP
  NP → N PP
  NP → N
  PP → P NP
  VP → V
  N → *lots*
  P → *of*
  V → *exist*

- We have nonterminal symbols, like S, NP, VP, N V
- Terminal symbols, like *lots*, *of*, and *exist*
- A start symbol, S.
- "Context free" = e.g., that the symbol NP expands as (Det) N-bar, irrespective of where the NP occurs.

17. Simple case of a context free language that is not regular

X → aXb
X → ab

[☞ What is this (famous) language?]
[☞ Intuitively, why can it not be generated by a finite state machine?]

18. The probabilistic version of context-free languages

- Take all the rules that have X on the left side.
  - ➢ e.g. VP → V NP, VP → V AP
- Give each one a probability, such that the probabilities sum to one.
- The probability of a sentence is the product of the probabilities of the rules that generate it during the derivation.
  - ➢ Often this is exceedingly small.
- We are now a tiny bit of the way to a syntactic model that can be trained on data and returns a super-rough approximation to gradient well-formedness judgments.
- On probability and judgements see:
  - ➢ Lau, Jey Han, Alexander Clark, and Shalom Lappin. "Grammaticality, acceptability, and probability: A probabilistic view of linguistic knowledge." Cognitive Science 41, no. 5 (2017): 1202-1241.

19. Next level up: *Context-sensitive languages*

- Nobody seems to use them.
- You can't even use a tree to display their computations.

20. Relevant work in the trans-context free region

- The region is subdivided into further subset relations.
- **Multiple context free grammars** (work of Carl Pollard, Hiroyuki Seki, etc.) are popular (e.g. with Tim Hunter) as a way of formalizing grammars powerful enough to do real syntax, like with long-distance dependencies.

21. The next level up: *computably enumerable*

- Operationally, this is "generable with a Turing machine" — the upper limit of the hierarchy.
- These gets us into profound intellectual results of the 20th century (e.g. Gödel) but doesn't seem to be very relevant to linguistics.

22. The full hierarchy as stated in the readings

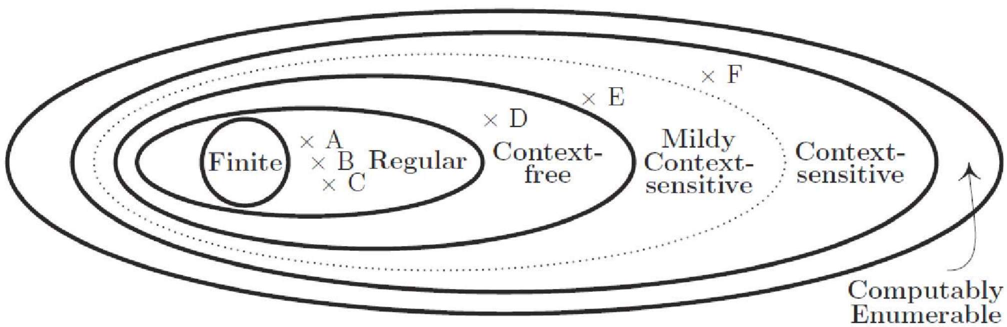


Fig. 1. The Chomksy Hierarchy. Point A represents English consonant cluster constraints (Clements & Keyser, 1983); B is sibilant harmony in Samala Chumash (Applegate, 1972); C is the unbounded stress pattern of Kwakiutl (Bach, 1975); D is English nested embedding (Chomsky, 1956); E is a dialect of Swiss German (Shieber, 1985); and F is copying constructions in Yoruba relative clauses (Kobele, 2006).

- It is fairly standard to leave out the n-grams, which fall between Finite and Regular.

23. There are many intermediate levels within the Regular class

- The Delaware School (below) has a particular interest in levels of the hierarchy that are lower than Regular; here is a diagram of such levels from the readings:

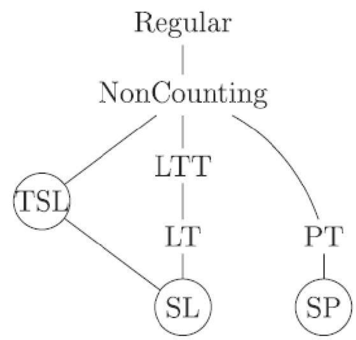


Fig. 3. Subregular classes of sets with proper inclusion relationships indicated from top to bottom. LT, locally testable; LTT, locally threshold testable; PT, piecewise testable; SL, strictly local; SP, strictly piecewise; TS, tier-based strictly local. Circled names indicate regions hypothesized to be where all individual phonotactic generalizations lie.

24. Many of the theorems of formal language theory are potentially very useful

- E.g., defining a process of "intersection" on finite state machines, such that the intersected machine recognizes strings that fall into both language define by the input machines.
- A theorem (Hopcroft and Ullman p. 135) tells us that the result is a regular language, so we can iterate the process ad infinitem — even formalize whole phonologies in one machine.

25. The structure of this material was mostly internally-generated, not a response to language data

- See (3) above on math.
- Notably, **copying** is ubiquitous in language.
  - Meaningless reduplication in phonology: *pompom*; topic of current work by Hayes and Jo.
  - morphological reduplication, which is ubiquitous
  - syntactic or phrasal reduplication; e.g. Jennifer Cole (1996) on Bengali; Greg Kobele (2006) on Yoruba
- Copying falls very high on the Chomsky hierarchy (somewhere in the context-sensitive region.
- I sometimes wonder (prompted by conversations with Edward Keenan) if it might be useful to start formal language theory all over, with axioms that permit copying as a primitive entity.

APPLICATIONS OF THESE MATHEMATICA L IDEAS TO PHONOLOGY

26. Rendering *SPE* rigorous

- Work starting with a book (1972) by C. Douglas Johnson explored the restrictions (modest) needed to make SPE phonology expressible with finite-state machines.

- A widely-admired paper by Kaplan and Kay (1994) did the full translation, complete with proof.

27. Rendering OT rigorous

- OT's Achilles' heel is the finite GEN of OT — what about the unfound winning candidate?
  - A hard-hitting paper title of Karttunen: "On the failure of pencil and paper linguistics", showing that Paul Kiparsky missed a winning candidate, revealed by Karttunen's finite-state computations.
  - Other leading scholars in the "make OT rigorous" campaign:
    — Jason Eisner, University of Pennsylvania
    — Jason Riggle, University of Chicago (Ph.D. UCLA 2004)
    — the late Daniel Albro (Ph.D. UCLA 2005) — trans-regular work, for reduplication

28. The Delaware School of Phonologyy

- This is a deliberately silly name, invented by Eric Baković, which nevertheless might stick.
- Delaware is the 2nd smallest state of the United States, located on the Atlantic coast.
- Key figure is **Jeffrey Heinz** (Ph.D. UCLA 2007), and various students he taught before leaving the University of Delaware for Stony Brook University: notably Adam Jardine and Jane Chandlee.
- The Delaware school is deeply involved in applying formal language theory to phonology, in a variety of ways.

29. Guiding work in Artificial Grammar Learning studies

- Conjecture: the computational complexity of a mapping influences the difficulty with which a particular pattern or mapping can be learned.
- See readings for one interesting case.

30. Learnability

- In principle, sound mathematical analysis can help us understand the behavior (perhaps even at the level of certain proof) of algorithms for learning phonology, a topic covered here.
  - Heinz, Jeffrey. "Learning long-distance phonotactics." *Linguistic Inquiry* 41, no. 4 (2010): 623-661.
- Such algorithms can be assessed for performances, e.g.
  - Wilson, Colin, and Gillian Gallagher. "Constraint complexity in surface-based phonotactics: A case study of South Bolivian Quechua." Linguistic Inquiry 49, no. 3 (2018): 610-623.

31. Taxonomic work

- The goal is to locate the full range of all phonological phenomena for where they lie on the extended Chomsky hierarchy.
- Method:
  - Look up a description of a phenomenon in the research literature, usually an analytical article.
  - Conduct mathematical analysis covering the published description, and figure out where it sits on the Chomsky hierarchy.
- By now, this work collectively embodies a substantial catalog of taxonomized processes, a true labor of love.

32. Consequences derived from the taxonomy: claims about the real world of phonology

- Phonology **calls on more limited computational resources** than morphology or syntax.
  - Specifically: "phonology is subregular".
  - Syntax is known to be transregular (famous paper by Schieber)
  - This theme appears repeatedly in the writings of Delawarians, e.g. in the readings.
  - The claim was picked up with enthusiasm in Chomsky and Berwick's language-and-evolution book *Why only us?* (2016): merge made syntax (hence, us) possible; phonology is a prelapsarian capacity, lacking Merge.[1]
  - Hayes and Jo's ms. disputes this, on ground of phonological reduplication.
- Some **particular areas of phonology** call upon richer computational resources than others; notably:
  - Jardine, Adam. "Computationally, tone is different." Phonology 33, no. 2 (2016): 247-283.
  - and replies by Hyman and Pater

33. A new theory of phonology?

- Delawarians tend to be OT-skeptics.
- This is fine, but it's hard to tell what they really think about issues of framework.
- What I hope will happen is that a leading Delawarian will propose an explicit framework. This will permit comparisons.
- (Contents of a framework: theory of representations; theory of components and their interaction, mechanisms to account for human ability to synthesize novel forms and to compute degrees of phonotactic well-formedness).
- Inside information suggests such a framework may be currently under construction by Delawarians.

---

[1] I would like to see this claim examined more closely: are not the hierarchies of phonological phrasing and metrical structure something we might appropriately treat with Merge?

COURSE SUMMARY

34. Setting: changes in the field of phonology

- The *SPE* era: intuitive rationalization of corpus
- Moving to: making **predictions**, about:
  - how consultant will respond to elicitation
  - experimental findings

35. MaxEnt

- How it works (the math)
- Is it intuitive?
- Algorithms (usable in analysis and learning) for finding best-fit weights
  - The concept of the objective function
- Statistical testing of constraints with the Likelihood Ratio Test

36. Theories with similar form and intent to MaxEnt

- Stochastic OT
- Noisy Harmonic Grammar
- Ways to distinguish?
  - Harmonically bound "semiwinners"? Example from Korean/Bengali phonological phrasing
  - Overgeneration in MaxEnt? Work of Anttila/Borgeson/Magri
  - Undergeneration in MaxEnt? Scary Tapping example — same constraint cannot be used for strict phonot

37. Blending effects in MaxEnt and related theories

- Moore-Cantwell's findings on syllable weight and final [i] in English.
- The blending problem:
  - How does probability of forms that violate two different constraints relate to probability of forms that violate just one of them?
- Constraints blend in particular ways, in different theories, yielding sigmoids etc.
- Some theories fail to match data well in such cases (Stochastic OT)

38. Maxent phonotactics

- We used an approach with MaxEnt, probability-as-measure-of-well-formedness, and only-markedness-constraints.
- You are all now experienced analysts in this domain due to a homework.
- Assessing the constraints: MaxEnt as an improvement over the venerable Observed/Expected statistic

39. Bias

- = Gradient forms of UG
- How it might be implemented in MaxEnt with a prior term (Wilson, later workers)
- Candidate types of bias
  - initial syllable Faithfulness (Becker and Nevins)
  - phonetic OO correspondence (Steriade, Zuraw, White)

40. Acquisition of phonology

- The Mennian vision: dual grammars and lexicons
  - theory of parent
  - method of talking
- Empirical arguments for this
- OT as a good candidate theory for child output phonology (Markedness effects, conspiracies)
- Quirks:
  - Child-specific constraints, based on
    random factors in learning to articulate
    word-sized templates like CajaC
  - Avoidance, and how to get it in OT frameworks
- Near-neutralization in child output phonology (cf. later, generative phonetics)
- Opacity in child output phonology, and how to interpret it
- Some ideas in how the child might learn the adult system.
  - Distributional segmentation (Goldwater et al.)
  - Allomorph discovery, alternation discovery
  - the problem of parallel learning

41. Learning hidden structure

- Why it is a recurring problem in the computational literature.
- Why some scholars are exterminationists.
- Expectation-Maximization as an underexplored route.

42. Paradigm Uniformity

- Manifested in historical change, experiment, acquisition, near-neutralization.
- Frameworks, with emphasis on OO-Correspondence

43. Weight as example of role of phonetics in phonology

- Gordon: weight is audible, measurable
- Ryan: weight is gradient when you look at gradient contexts

44. Generative phonetics

- mathematical regularities detectible in phonetic experimentation
- Flemming's Harmonic Grammar approach — parabolic violations
  - Lefkowitz's Theorem
- Braver and the compromise theory of Paradigm Uniformity
- Lefkowitz and the search for general mathematical laws of phonetic realization
  - random variation matches structural
  - asymmetrical compression
  - skewness

45. Gradient symbolic representations; mathematical phonology

COURSE EVALUATIONS

46. You can find this on MyUCLA

- Please take the last ten minutes to provide feedback.