

Most of the course up until now has shown you **finite-state** models for language processing. These are pretty powerful and well-studied, but when the time comes to talk about deeper levels of linguistic representation - syntax and semantics - they become less elegant. We're about to make an important jump into syntax, and this lecture helps to motivate that jump.

This lecture discussed the **Chomsky hierarchy** (aspects of which should have been review), and foreshadowed some of the tradeoffs implied by deciding to use a model that has more or less generative power.

You should understand the following concepts:

- Regular languages (and their relationship to FSAs and regexps)
- Context-free languages
- Context-sensitive languages (and the existence of a “mildly” context-sensitive subset that seems to be of particular interest in linguistics)
- Recursively enumerable languages
- Pumping lemmas
 - Their use in proving that languages are not finite-state (you saw the pumping lemma)
 - Their use in proving that languages are not context-free (we talked about this only in brief)
 - The correct use of intersecting with finite-state languages in these proofs
- Recursive center embedding in English
- Cross-serial dependencies in Swiss German

There exist natural language theories at every level of the Chomsky hierarchy!

Regarding models we will see in this class, moving **up** the Chomsky hierarchy gives us models that jive better with linguistics¹ and are capable of more accurately predicting linguistic phenomena and avoiding predicting things that don't happen in the language. Moving **down** the Chomsky hierarchy gives us models with faster processing time and simpler requirements for data. Unigram models may be an extreme in this regard.

¹Caveat: some linguists—particularly those who prefer theories at the “mildly” context-sensitive level—argue that going too far up the hierarchy gives *too much* generative power, and that this is a bad thing. Indeed, most linguists will agree that the theory should use all the power it needs and no more, and so it's a matter of what kinds of constraints are appropriate. This is out of scope for this class.