Grammar Complexity

Jurafsky and Martin, pp. 477-498

Miriam Butt
October 2002

Complexity

- How Complex is a given Problem?
- What formal mechanisms best model this complexity?

Natural Language: used to be thought of as a sort of "code". That is hard, but regular.

Now: mind-bogglingly complex.

But: is it an unsolvable problem?

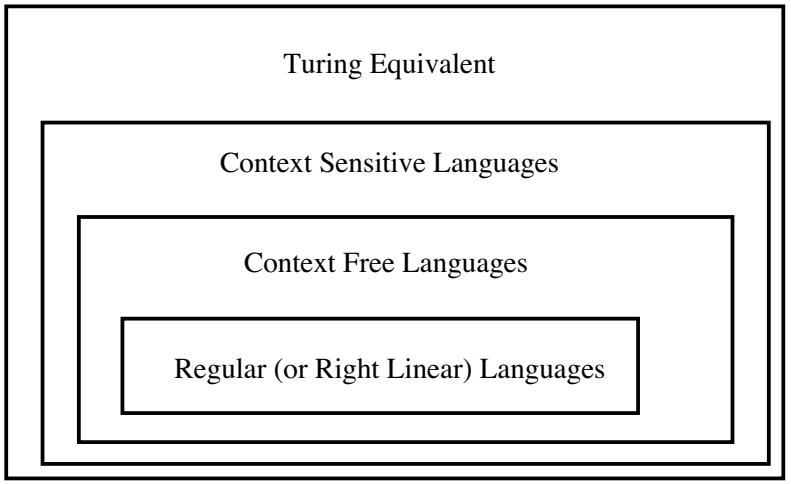
Generative Power

Chomsky defined a theory of language (syntax) in terms of **generative** linguistics.

Given a set of rules and a lexicon, what well-formed expressions can we generate and do those adequately cover the empirical data we observe?

"One grammar is of greater generative power or complexity than another if it can define a language that the other cannot define." (J&M p. 478)

The Chomsky Hierarchy



Natural Language

Is it regular? Overall no.

But, subparts of it are: phonology and morphology (can be treated via FST which are known to be regular, Kaplan and Kay 1994, Karttunen 2002).

How can we tell if a language is not regular?

The Pumping Lemma

The Pumping Lemma

Let L be an infinite regular language. Then there are strings x, y, and z, such that $y \neq \varepsilon$ and $xy^nz \in L$ for $n \geq 0$.

If a language is regular, it can be modeled by a FSA.

If you have a string which is longer than the fixed number of, the FSA must have a loop.

$a^n b^n$ is not a part of this language (see J&M p. 484)

Natural Language

Natural Language contains strings like:

The cat likes tuna fish.
The cat the dog chased likes tuna fish.
The cat the dog the rat bit chased likes tuna fish.
The cat the dog the rat the elephant admired bit chased likes tuna fish.

$a^n b^{n-1}$ so, not a regular language

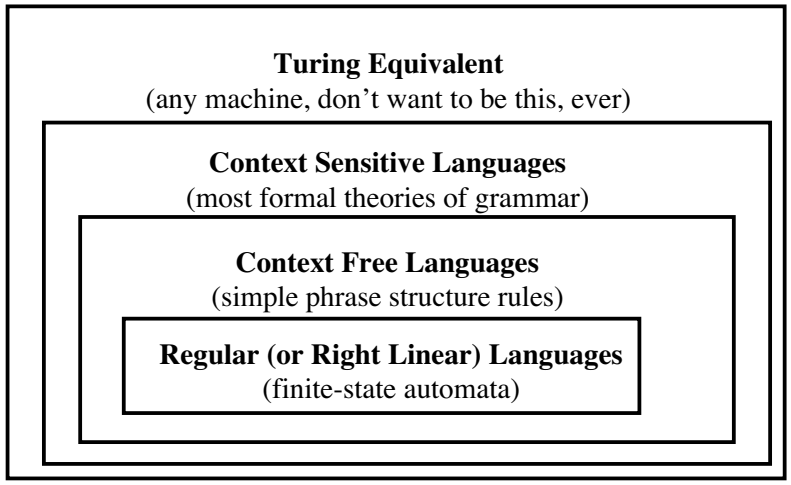
Natural Language

Is it context-free? No.

So, Natural Language turns out to be a very hard problem: an **NP-complete** problem (term from computer science).

Should we give up? No --- there are still ways to make things computable.

The Chomsky Hierarchy



Decidability

The more you know about the formal properties of an underlying syntactic theory, the better.

Montonicity: this basically means you do not overwrite information once you've got it as part of your analysis.

Mathematical Proofs: based on the properties of one's formal theory**,** one can prove whether it is *decidable* or not.

Decidability

The more you know about the formal properties of an underlying syntactic theory, the better.

GB/Minimalism: couched in a very formal way, but includes unconstrained movements, which makes it *non-monotonic* and puts it into the space of a Turing Machine.

HPSG: formal properties still under debate and an active area of research (e.g., Lexical Rules).

LFG: formal properties well understood and has been proven to be decidable.

Decidability

The more you know about the formal properties of an underlying syntactic theory, the better.

GB/Minimalism: couched in a very formal way, but includes unconstrained movements, which makes it *non-monotonic* and puts it into the space of a Turing Machine.

HPSG: formal properties still under debate and an active area of research (e.g., Lexical Rules).

LFG: formal properties well understood and has been proven to be decidable (Kaplan and Bresnan 1982, Backofen 1993).

Decidability

"First, an explanatory linguistic theory undoubtedly will impose a variety of substantive constraints on how our formal devices may be employed in grammars of human languages. ... It is quite possible that the worst case computational complexity for the subset of lexical-functional grammars that conform to such constraints will be plausibly sub-exponential." [Kaplan and Bresnan 1982]

In practice, one can (and does) also come up with smart computational techniques that avoide the worst-case scenario.