Topics in Natural Language Processing

Shay Cohen

Institute for Language, Cognition and Computation

University of Edinburgh

Lecture 4

Grammars

Grammar: a formal system of rules that govern the production of a language Language can be formal language or natural language

Grammars

Grammar: a formal system of rules that govern the production of a language Language can be formal language or natural language

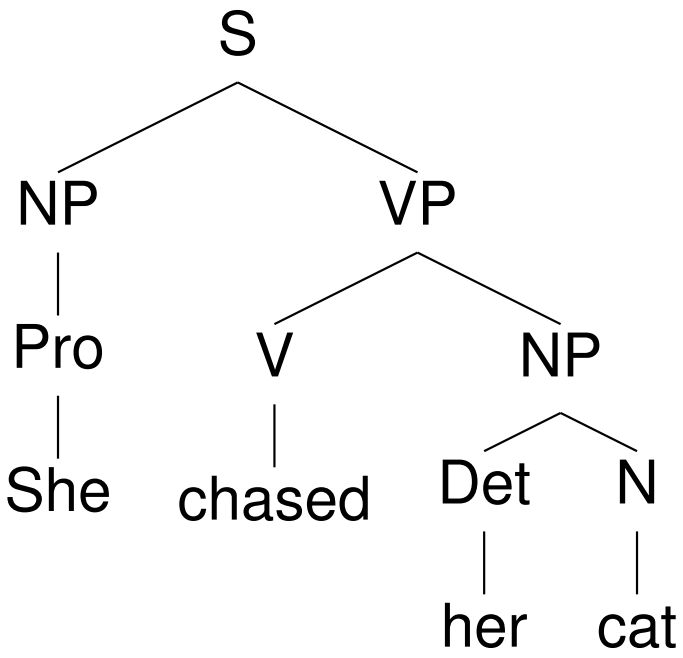
Probabilistic Grammar: augment a set of rules with probabilities to get distributions over derivations and strings

Context-Free Grammars

Context-free grammars, quick reminder:
A grammar $G = \langle N, S, V, R \rangle$ consists of

- Nonterminals N with a special start nonterminal $S \in N$
- Terminals V
- A set R of production rules of the form $A \rightarrow \alpha$ where $A \in N$ and $\alpha \in (V \cup N)^*$

They describe a top-down process for creating phrase-structure trees:



Why Do We Need Grammars?

- They implement the idea of compositionality very elegantly
- They are often interpretable – you can understand why rules are there and what they represent in language or otherwise
- They often have relatively efficient *inference* and *parsing* algorithms to find most likely derivations and structures

Why Do We Need Grammars?

- They implement the idea of compositionality very elegantly
- They are often interpretable – you can understand why rules are there and what they represent in language or otherwise
- They often have relatively efficient *inference* and *parsing* algorithms to find most likely derivations and structures

Originally formal grammars came as an attempt to formalise the rules behind natural language, now they are ubiquitous in computer science and there is an area of research called “formal language theory” that studies them.



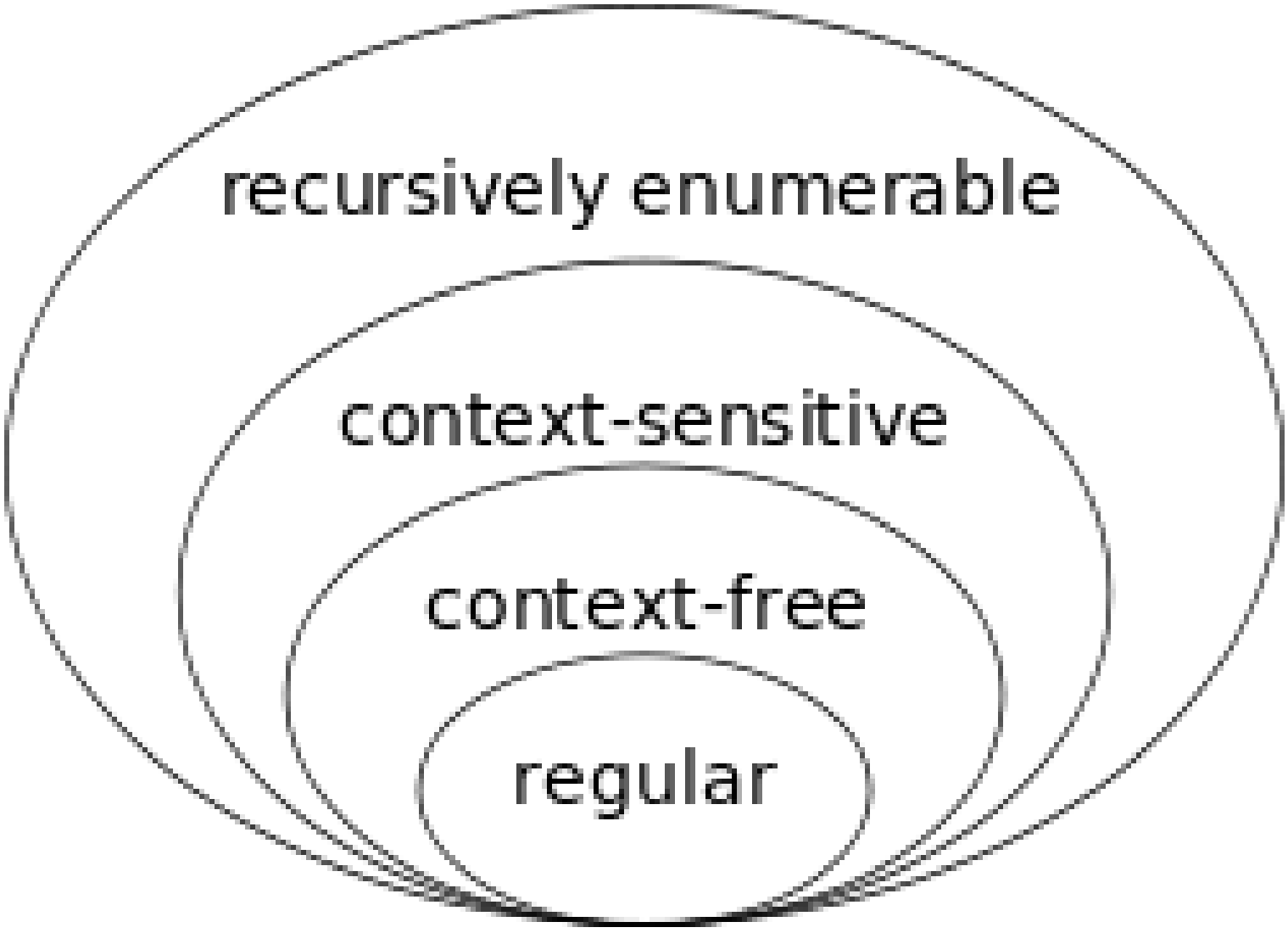
Also in computer vision...

Important Notions

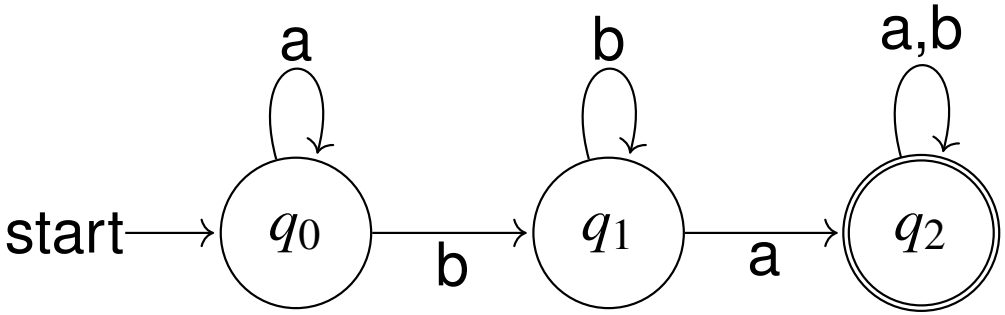
Let G be a grammar:

- String language $L(G)$
  The string represented by a given structure in the grammar (such as the yield of a phrase structure tree)
- Derivation, derivation language $D(G)$
  Describing the steps it takes to derive a structure
- Structure language, $S(G)$
  Such as phrase structure trees

The Chomsky Hierarchy



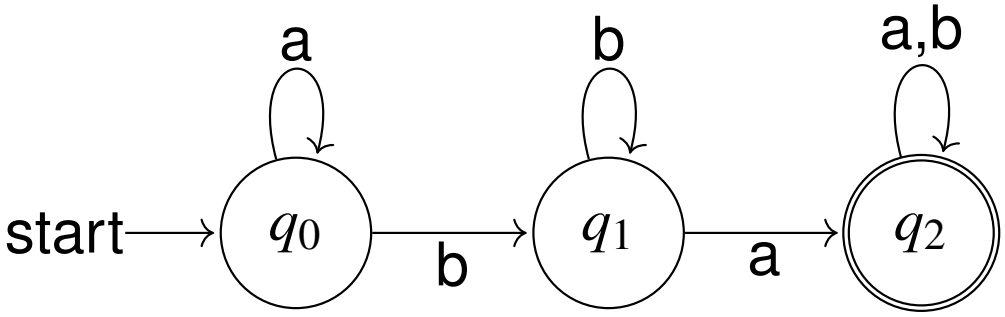
Regular Languages as Context-Free Languages



$$q_0 \to a\ q_0,\ q_0 \to b\ q_1,\ q_1 \to b\ q_1$$

$$q_1 \to a\ q_2,\ q_2 \to a\ q_2,\ q_2 \to a\ q_2,\ q_2 \to \varepsilon$$

Regular Languages as Context-Free Languages



$$q_0 \to a\ q_0,\ q_0 \to b\ q_1,\ q_1 \to b\ q_1$$

$$q_1 \to a\ q_2,\ q_2 \to a\ q_2,\ q_2 \to a\ q_2,\ q_2 \to \varepsilon$$

The states become nonterminals. We have rules of the form $\text{InState} \to x\,\text{OutState}$ for every edge in the FSA that transitions from InState to OutState emitting x.

This means we get a *linear* grammar – a grammar with a single nonterminal on the righthand side

Is Language Regular?

Center embedding implies language is not regular:

- This is the rat that ate the malt.

Is Language Regular?

Center embedding implies language is not regular:

- This is the rat that ate the malt.
- This is the malt that the rat ate.

- This is the cat that bit the rat that ate the malt.
- This is the malt that the rat that the cat bit ate.

- This is the dog that chased the cat that bit rat that ate the malt.
- This is the malt that the rat that the cat that the dog chased bit ate.

Main Idea

We want to show that the language $L = \{A^n B^n\} \cup \{C^n D^n\}$ is not regular.

We know the language $L' = \{A^n B^n\}$ is not regular.

Can we claim that because $L' \subseteq L$, L is not regular either?

Main Idea

We want to show that the language $L = \{A^n B^n\} \cup \{C^n D^n\}$ is not regular.

We know the language $L' = \{A^n B^n\}$ is not regular.

Can we claim that because $L' \subseteq L$, L is not regular either?
No! What if $L = \{\textit{any string}\}$. It is regular, and $L' \subseteq L$.

Our proof technique: *intersect* L with L'' where L'' is regular. If we do not get a regular language, then L is not regular, because the intersection of two regular languages is always regular

In this case, we will choose $L'' = A^*B^*$. We get that $L \cap L'' = L'$. L' is not regular, therefore L'' is not regular.

Is Language Regular?

Let $A = \{a_1, \ldots, a_m\}$ be the set of nouns and $B = \{b_1, \ldots, b_m\}$ the set of matching verbs.

Assume English was regular. Intersect it with the regular language **This is the malt (that the** A**)*** B^*.

Then we get **This is the malt (that the** A**)**^{n} B^n. Clearly not regular.

But the intersection of any regular language with another is also regular. Hence English cannot be regular.

Is Language Regular?

Let $A = \{a_1, \ldots, a_m\}$ be the set of nouns and $B = \{b_1, \ldots, b_m\}$ the set of matching verbs.

Assume English was regular. Intersect it with the regular language **This is the malt (that the** A**)*** B^*.

Then we get **This is the malt (that the** A**)**^{n} B^n. Clearly not regular.

But the intersection of any regular language with another is also regular. Hence English cannot be regular.

Point for a philosophical debate: is unbounded center embedding really part of English? There is strong evidence that we cannot grasp center embedding of depth larger than 3. See Levinson (2013) for a discussion.

Is Language Context-Free?

Let G be a grammar.

$T(G) =$

$L(G) =$

- Dutch - there are structures in Dutch which do not appear in any $T(G)$ for any G context-free grammar

Is Language Context-Free?

Let G be a grammar.

$T(G) =$

$L(G) =$

- Dutch - there are structures in Dutch which do not appear in any $T(G)$ for any G context-free grammar
- Swiss-German - there are strings in Swiss-German which do not appear in any $L(G)$ (and hence in any $T(G)$) for any G CFG

The constructions are similar to demonstrate that. Swiss-German uses case markers.

Non-projectivity

mer	em	Hans	s	huus	hälfed	aastriiche
we		Hans	the	house	helped	paint

... and “we have wanted to let the children help Hans paint the house.”

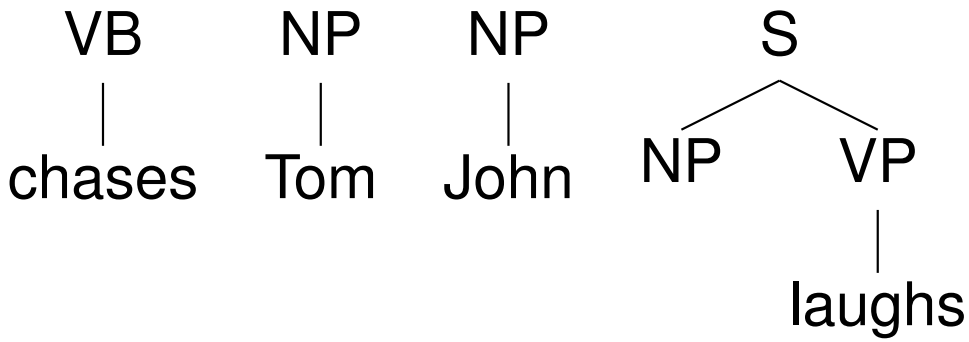
Intersect Swiss-German with a regular language and you get a non-context-free language.

But intersection of context-free languages with regular languages is context-free.

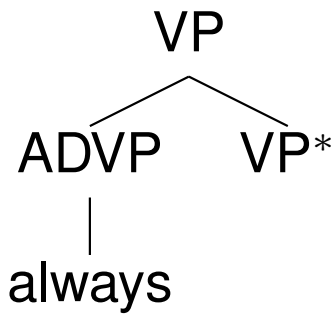
More about this in the assignment!

Tree Adjoining Grammars

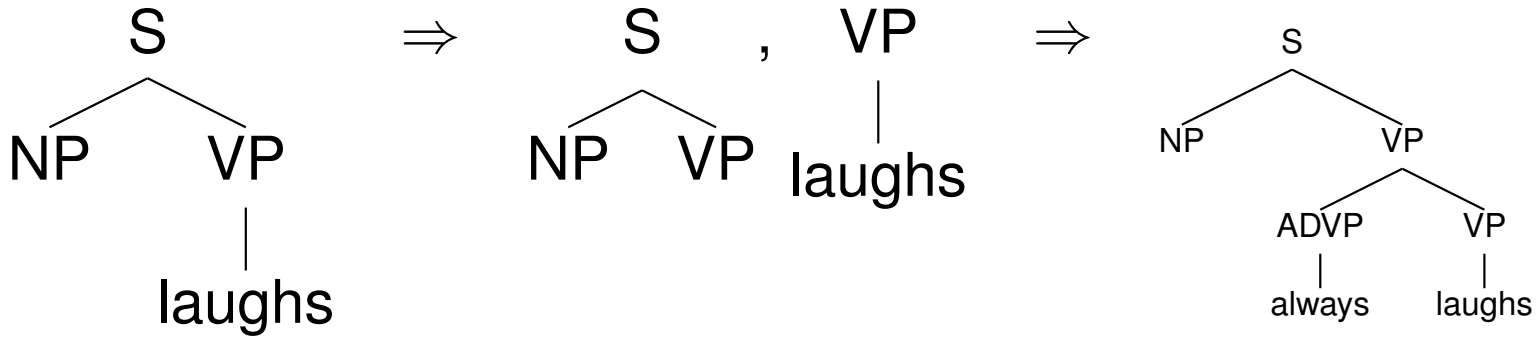
Initial trees:



Auxiliary trees:



Derivational process (when doing adjunction):



Tree Adjoining Grammars

- The initial trees and auxiliary trees together are also called “elementary trees.”
- Can have a constraint on the nodes where adjunction is allowed and where it is not allowed.

Tree Adjoining Grammars

Quick question: is $\{ww^R \mid w \in \Sigma^*\}$ a context-free language where w^R is the reverse string of w?
Yes, the following grammar accepts that language:

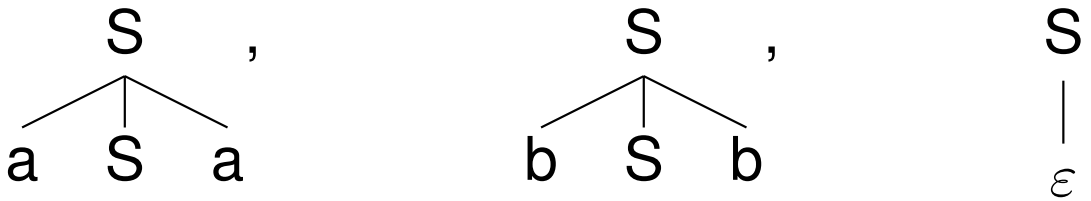
$$S \to aSa \mid bSb \mid \varepsilon$$

Tree Adjoining Grammars

Quick question: is $\{ww^R \mid w \in \Sigma^*\}$ a context-free language where w^R is the reverse string of w?
Yes, the following grammar accepts that language:

$$S \to aSa \mid bSb \mid \varepsilon$$

Is it a tree adjoining language?
Yes. Any context-free grammar is also a tree adjoining grammar without adjunction.

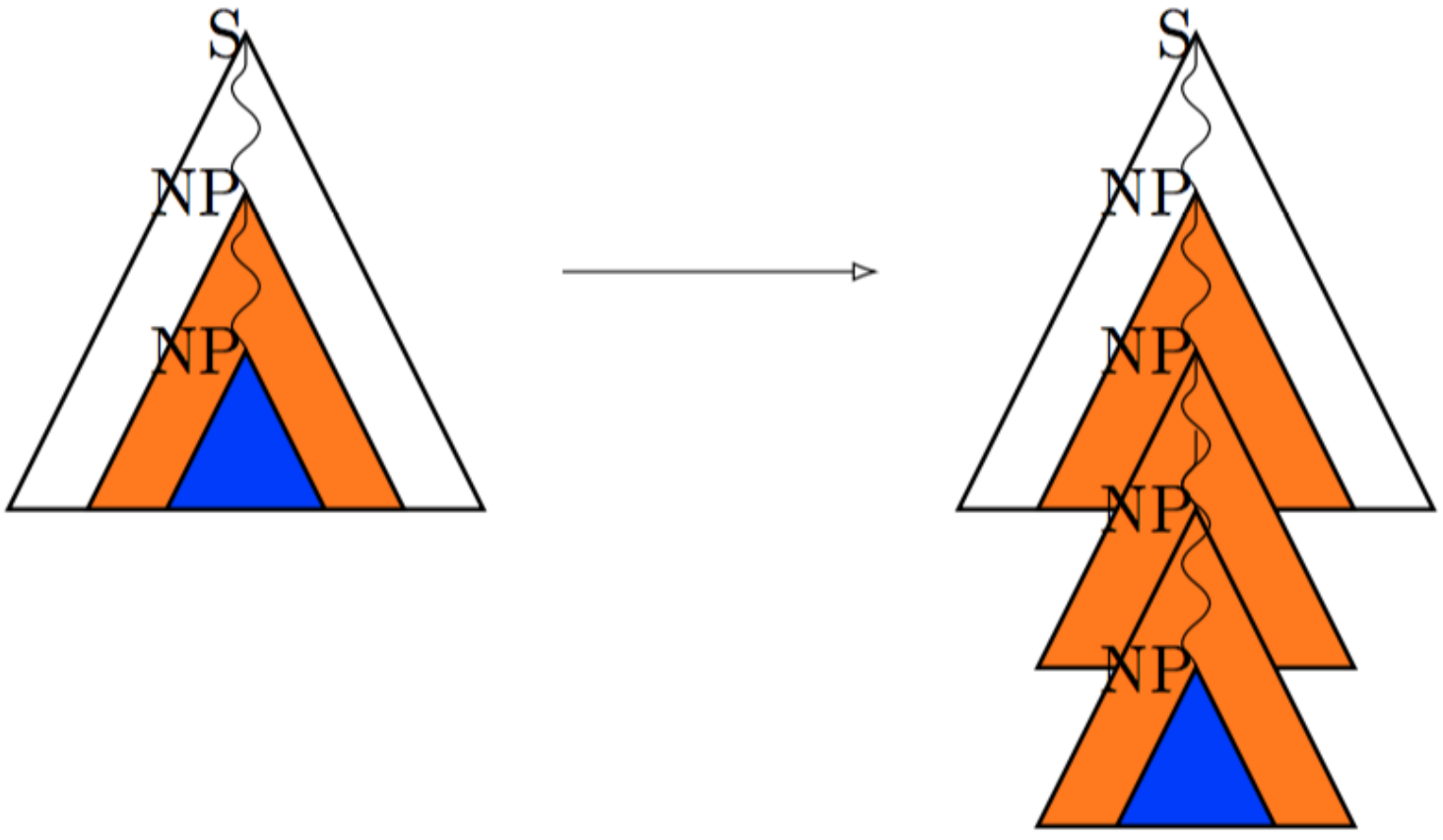


Tree Adjoining Grammars

Quick question: is $\{ww \mid w \in \Sigma^*\}$ a context-free language?
No. Show it by using the pumping lemma.

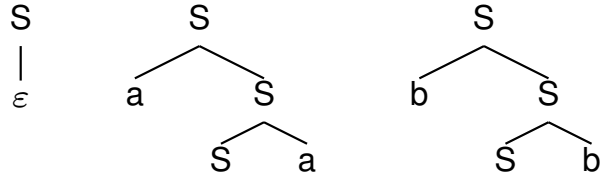
Tree Adjoining Grammars

Quick question: is $\{ww \mid w \in \Sigma^*\}$ a context-free language?
No. Show it by using the pumping lemma.

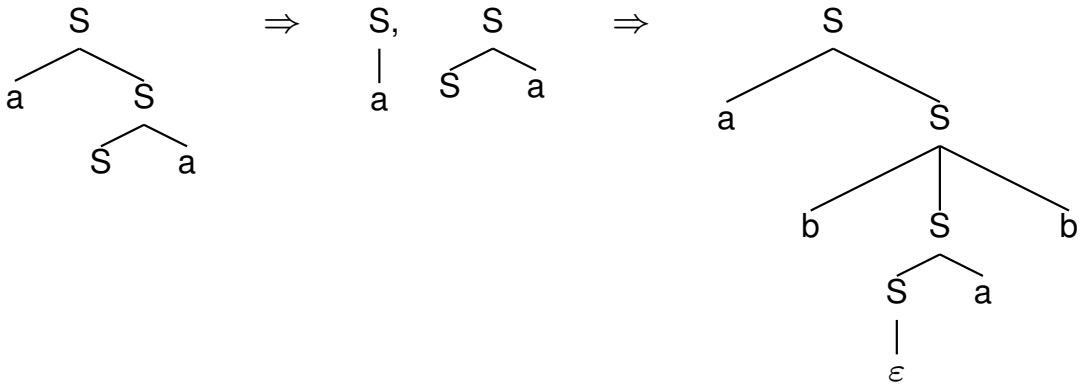


Tree Adjoining Grammars

Quick question: is $\{ww \mid w \in \Sigma^*\}$ a context-free language?
Is it a tree adjoining language?
Yes. Grammar:



Derivation:



Tree Adjoining Grammars

Another quick question: is $\{a^n b^n c^n d^n \mid n \geq 1\}$ a context-free language?

Tree Adjoining Grammars

Another quick question: is $\{a^n b^n c^n d^n \mid n \geq 1\}$ a context-free language?
Not context-free. Again show by the pumping lemma. It is a tree adjoining language.

Tree Adjoining Grammars

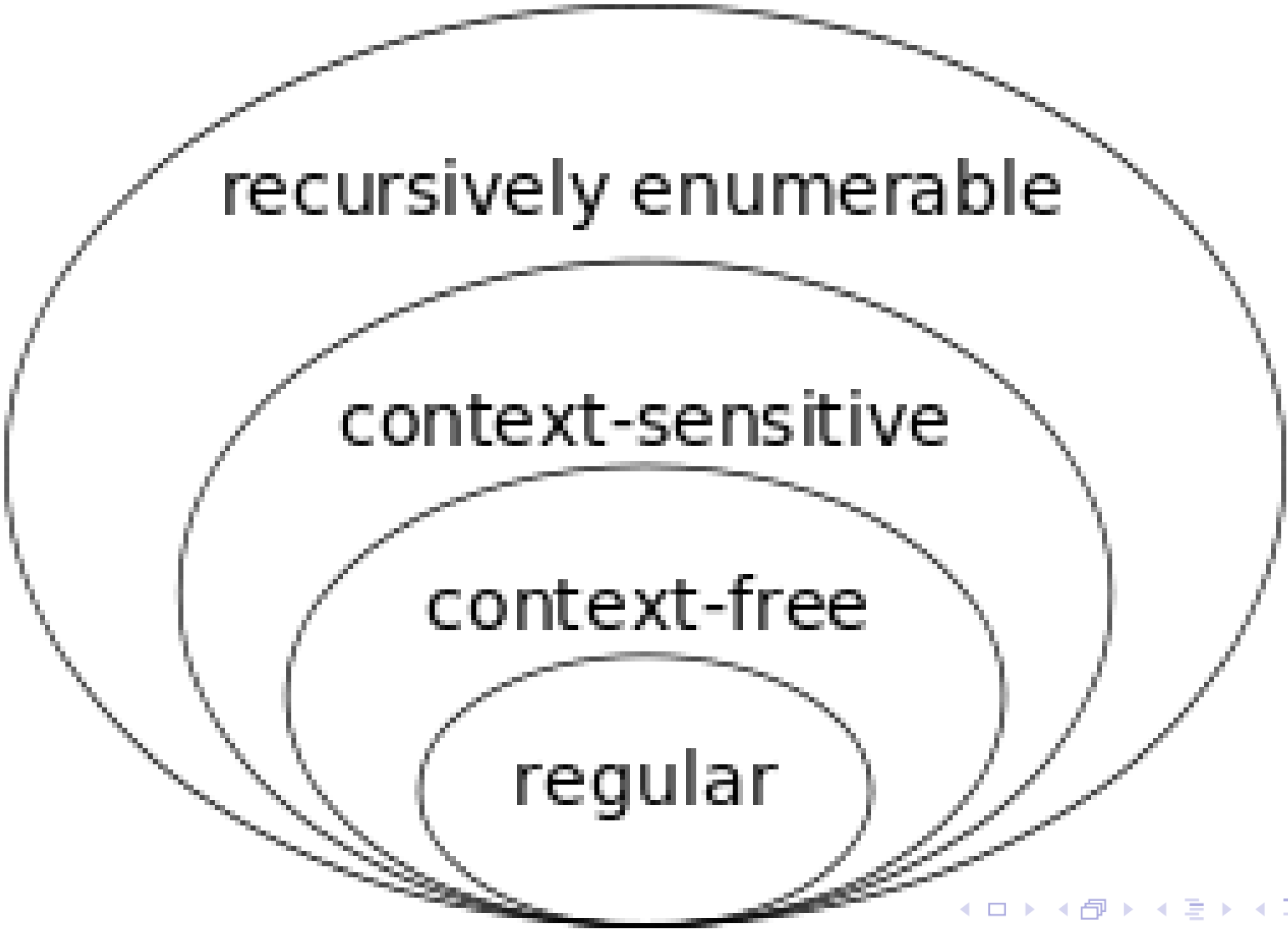
They add the “minimum needed” in order to capture phenomena such as cross-serial dependencies

They are part of a family of grammar formalisms called “mildly context sensitive”

Other examples which are weakly equivalent: combinatory categorial grammars, head grammars, linear indexed grammars

The Chomsky Hierarchy Plus

Where do we add tree adjoining grammars (part of “mildly context-sensitive languages”)?

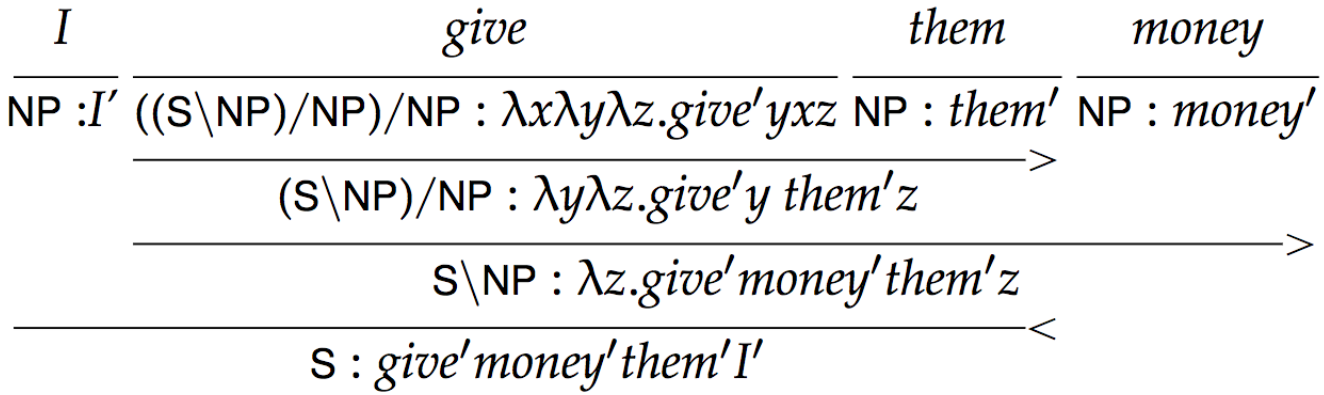


Another Mildly Context-Sensitive Formalism: CCG

Combinatory Categorial Grammars (due to Mark Steedman):

- Give easy access to logical form *semantics*
- Categories are "functions". There are some atomic categories (**NP** for noun phrase, **S** for sentence) and composed ones such as verb: **S** $\backslash$ **NP**: a category that takes NP on the right and gives back an S
- Main operations:
  - Application: $X/Y, Y \rightarrow X$
  - Application: $Y, X\backslash Y \rightarrow X$
  - Composition (forward): $X/Y, Y/Z \rightarrow X/Z$
  - Composition (backward): $X\backslash Y, Y\backslash Z \rightarrow X\backslash Z$
- Steedman (2000) also uses crossed composition, generalised composition, generalised crossed composition and type-raising.

CCG Derivation



(From Hockenmaier and Steedman (2007))

- A CCG consists of a lexicon that attaches each word a category and a semantic attachment (in the form of a λ expression)
- A certain version of CCG is *weakly* equivalent to tree adjoining grammars (Vijay-Shanker and Weir, 1994)

A More Powerful Formalism: LCFRS

Linear Context-Free Rewriting Systems are a more powerful formalism than CCG and TAG, which is still considered mildly context sensitive.

A rewrite rule in an LCFRS can generate several discontinuous strings that can move around in the derivation tree to different places.

A version of LCFRS has been used by Stabler to formalise the minimalist programme of Chomsky, where “movement” of constituents is a central part of the theory

Recipe for Mildly Context-Sensitive Formalisms

A set $\mathcal{L}$ of languages is mildly context-sensitive iff:

- $\mathcal{L}$ contains all context-free languages
- $\mathcal{L}$ can describe cross-serial dependencies: There is an $n \geq 2$ such that $\{w^k \mid w \in \mathcal{T}^*\} \in \mathcal{L}$ for all $k \leq n$
- The languages in $\mathcal{L}$ are polynomially parsable
- The languages in $\mathcal{L}$ have the constant growth property (if we order the words by their length, the length grows in constant steps)

A formalism is mildly context-sensitive iff the set of languages it defines is mildly context-sensitive

Theory of Syntax

Mainstream claim in CL is that mild context-sensitivity in some form is sufficient to capture any natural language, most likely in the form of TAG and CCG.

Just like any other scientific theory, if you want to prove otherwise, you need to falseify this theory by giving an example that shows language is not mildly context-sensitive.

There have been some attempts to construct such counterexamples, but most of them turned out to be either ill-constructed or use wrong linguistic data.

Probabilistic Grammars

We augment the rules with probabilities

The probability of a derivation is then the product of all rule probabilities:

Often can be thought of as a generative process: we start with the initial symbol and probabilistically choose rules until we reach terminal nodes

There are also weighted versions: