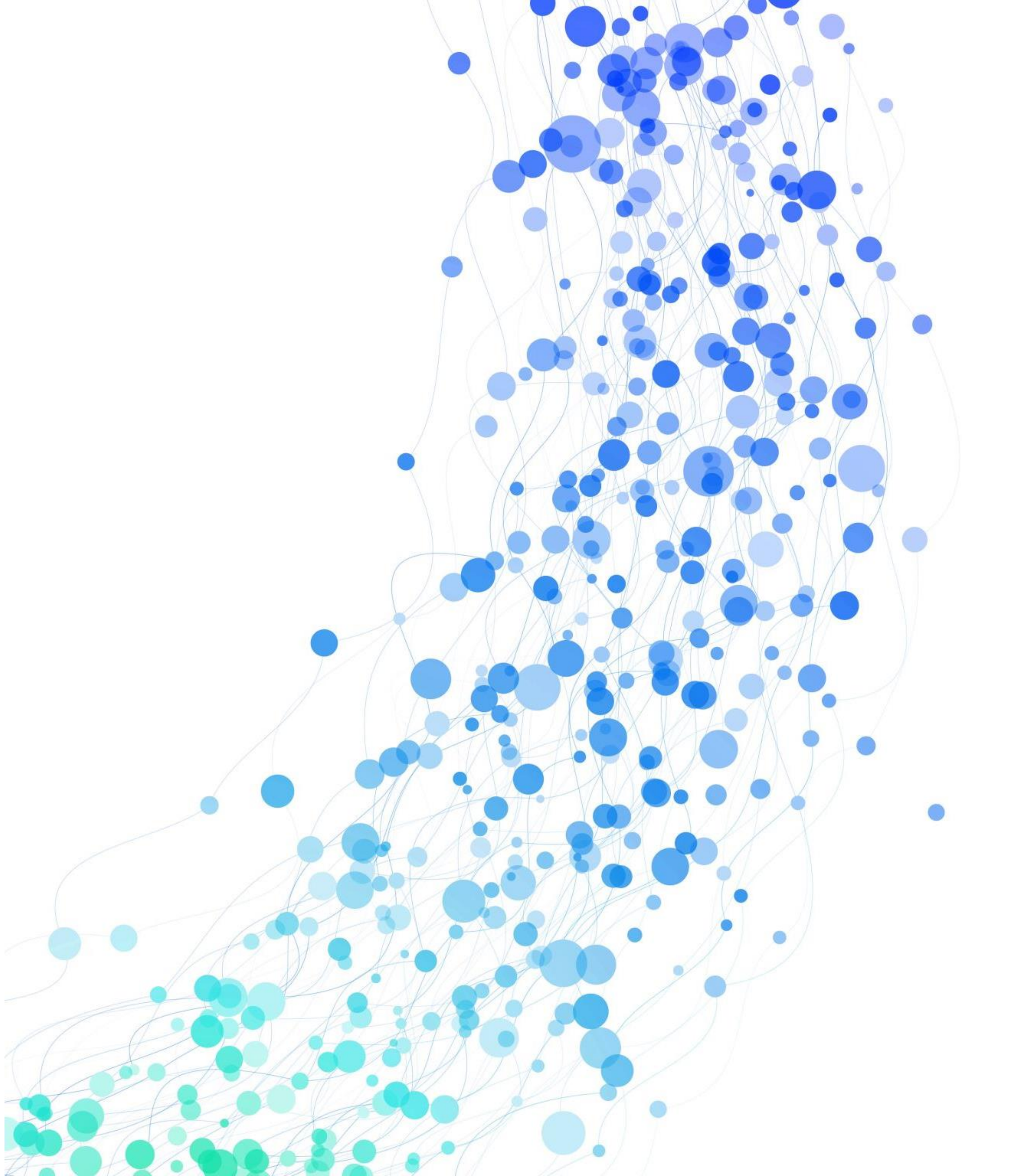
# LECTURE 17: CFG PARSING

Mehmet Can Yavuz, PhD.

Adapted from Julia Hockenmaier, NLP S2023 - course material
https://courses.grainger.illinois.edu/cs447/sp2023/



## INTRODUCTION TO SYNTAX

## PREVIOUS KEY CONCEPTS

NLP tasks dealing with **words**…
– POS-tagging, morphological analysis

… requiring **finite-state representations**,
– Finite-State Automata and Finite-State Transducers

… the corresponding **probabilistic models**,
– Probabilistic FSAs and Hidden Markov Models
– Estimation: relative frequency estimation, EM algorithm

… and **appropriate search algorithms** – Dynamic programming: Viterbi

# THE NEXT KEY CONCEPTS

NLP tasks dealing with **sentences**… – Syntactic parsing and semantic analysis

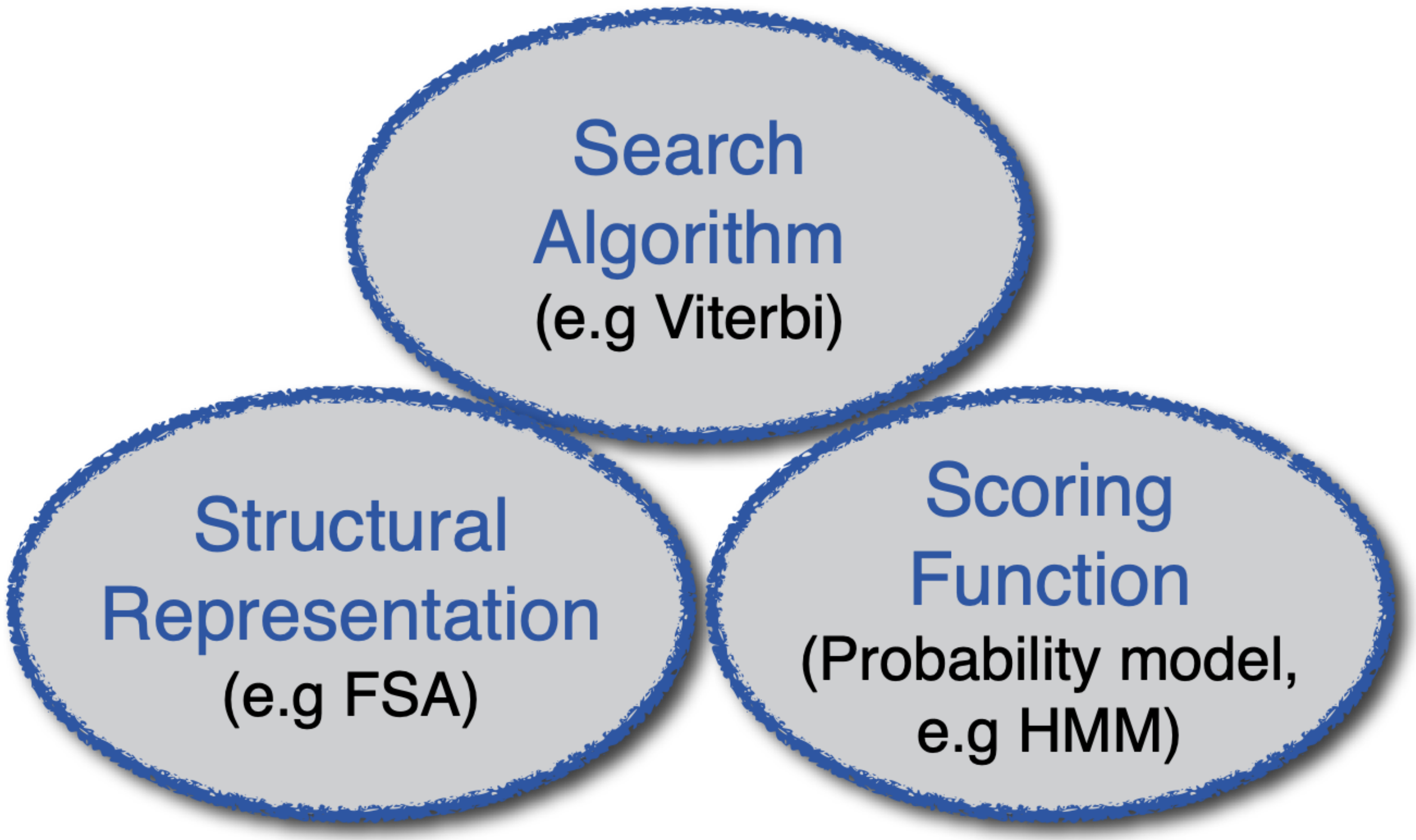
… require (at least) **context-free representations**, – Context-free grammars, dependency grammars,

unification grammars, categorial grammars … the corresponding **probabilistic models**,

– Probabilistic Context-Free Grammars … and appropriate **search algorithms**

– Dynamic programming: CKY parsing

## DEALING WITH AMBIGUITY



## TODAY'S LECTURE

Introduction to natural language syntax ('grammar'):

Part 1: Introduction to Syntax (constituency, dependencies,...)

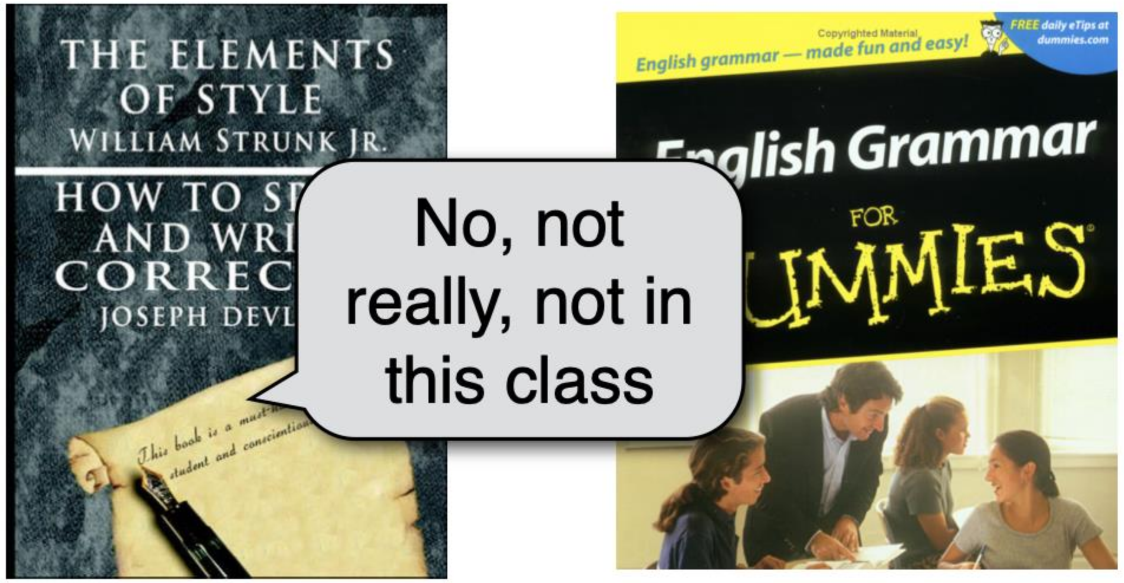
Part 2: Context-free Grammars for natural language

Part 3: A simple CFG for English

Part 4: The CKY parsing algorithm

Reading: Chapter 12 of Jurafsky & Martin

# WHAT IS GRAMMAR?


THE ELEMENTS OF STYLE
WILLIAM STRUNK JR.
No, not really, not in this class
English grammar — made fun and easy!
FREE daily eTips at dummies.com
English Grammar
FOR

Grammar formalisms:

A precise way to define and describe the structure of sentences.

There are many different formalisms out there.

## WHAT IS GRAMMAR?

**Grammar formalisms**

(= syntacticians’ programming languages)

A precise way to define and describe the structure of sentences.

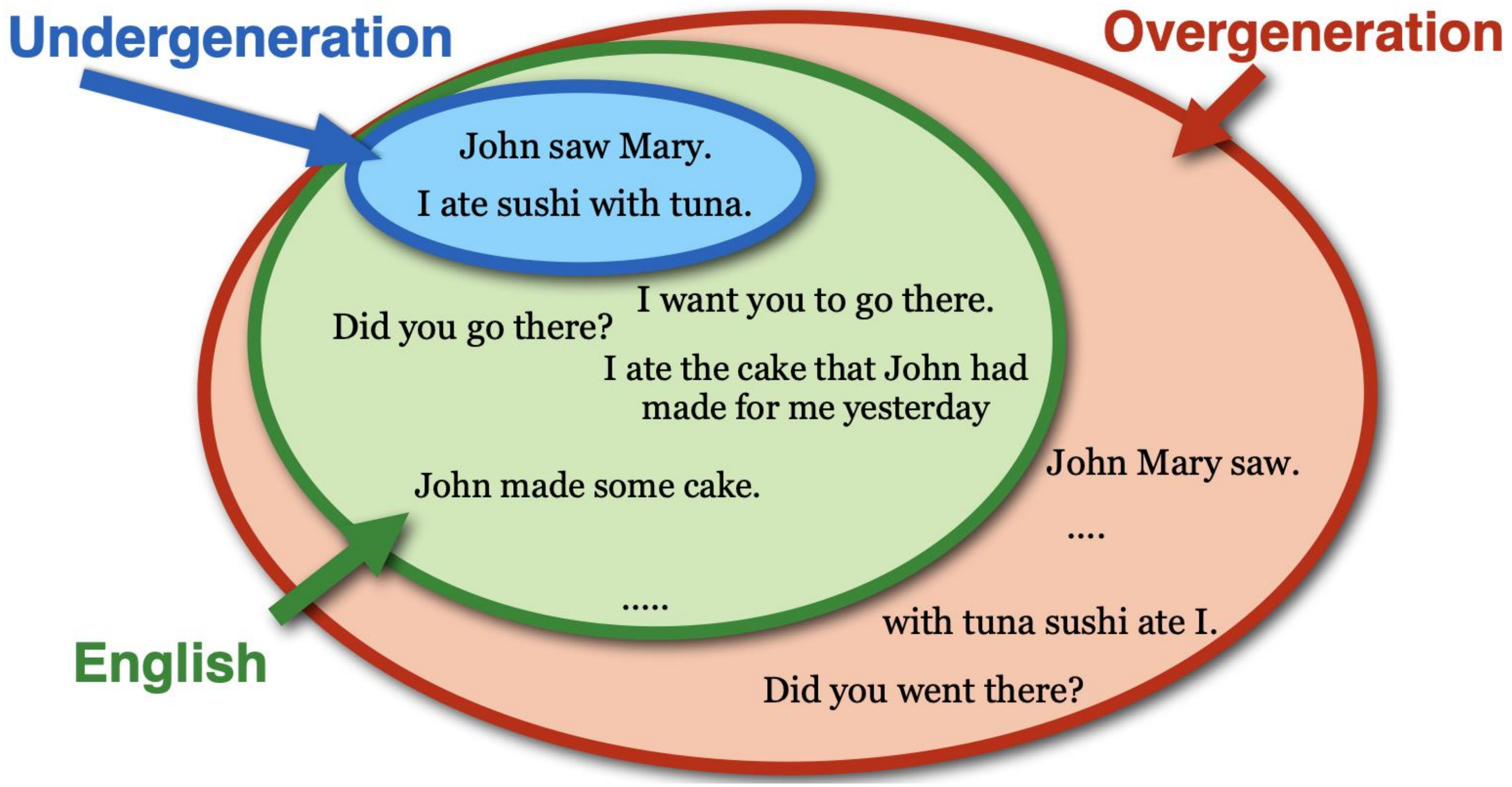
(N.B.: There are many different formalisms out there, which each define their own data structures and operations)

**Specific grammars**

(= syntacticians’ programs)

Implementations (in a particular formalism) for a particular language (English, Chinese,….)

## CAN WE DEFINE A PROGRAM THAT GENERATES ALL ENGLISH SENTENCES?



## CAN WE DEFINE A PROGRAM THAT GENERATES ALL ENGLISH SENTENCES?

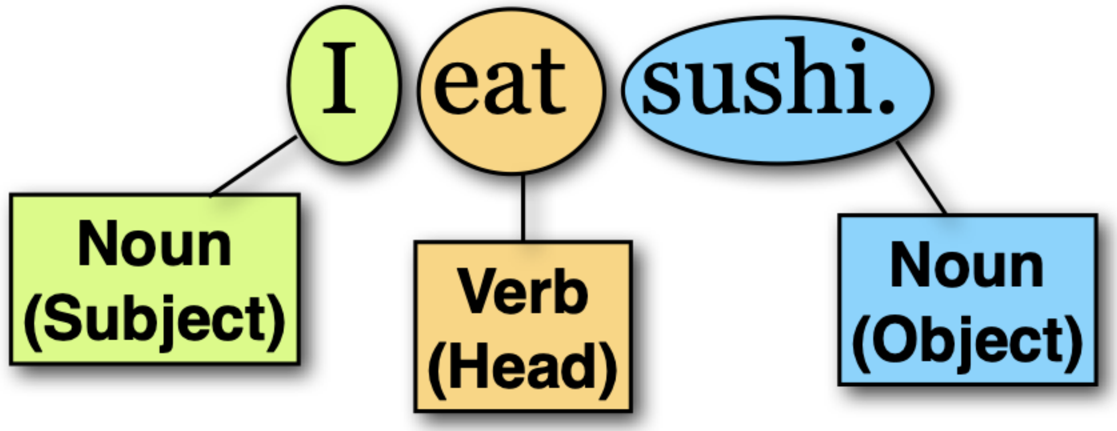
**Challenge 1: Don't *undergenerate*!**
(Your program needs to cover a lot different constructions)

**Challenge 2: Don't *overgenerate*!**
(Your program should not generate word salad)

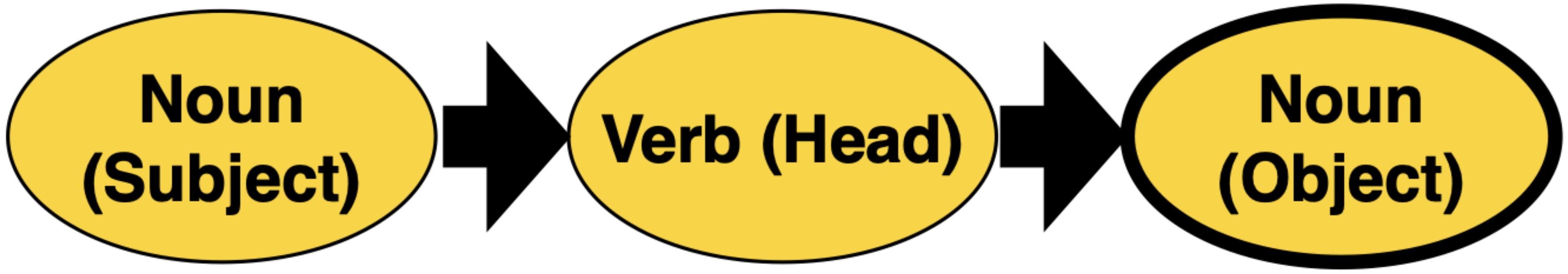
**Challenge 3: Use a finite program!**

Recursion creates an infinite number of sentences (even with a finite vocabulary),
but we need our program to be of finite size

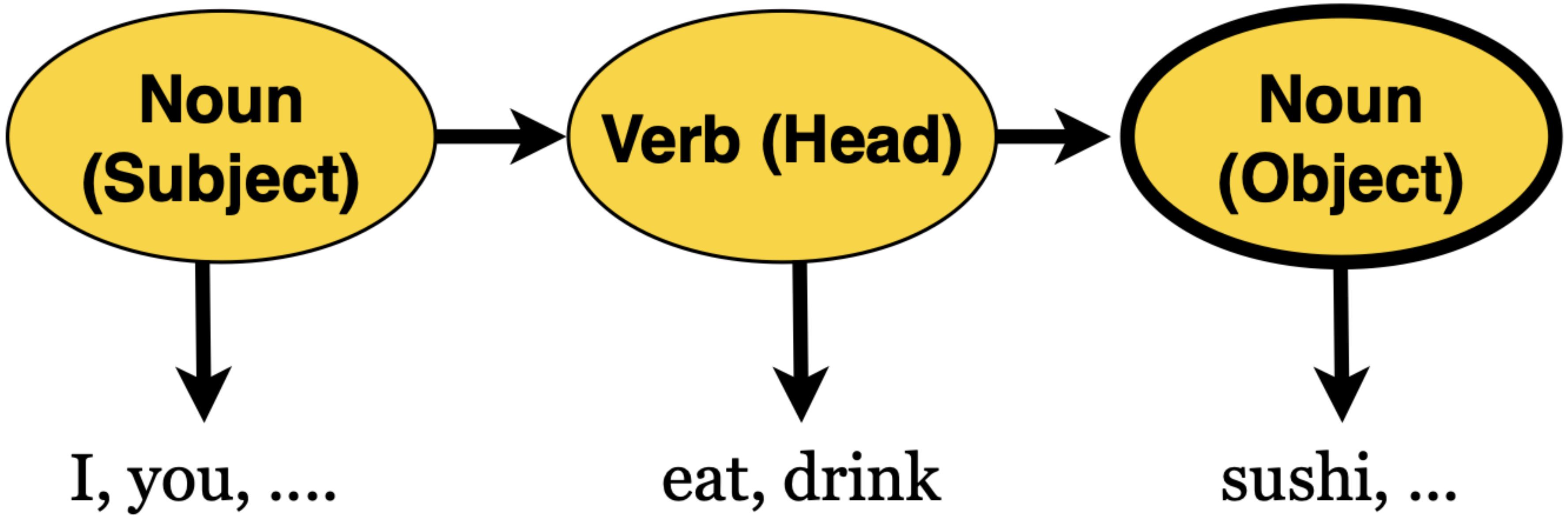
## BASIC SENTENCE STRUCTURE



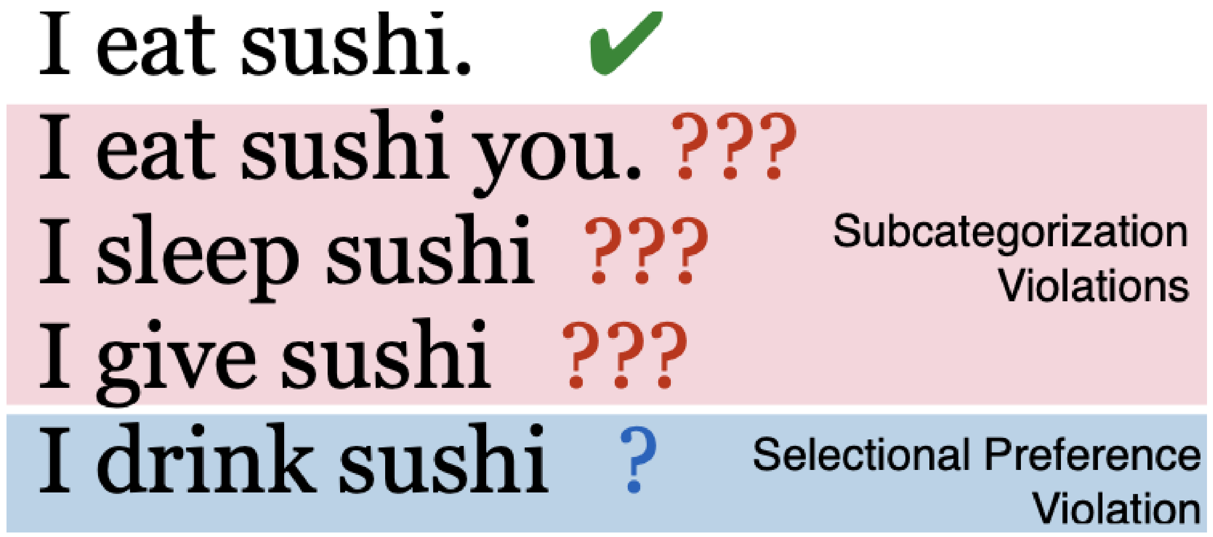
## A FINITE-STATE-AUTOMATON (FSA)



## A HIDDEN MARKOV MODEL (HMM)



## WORDS TAKE ARGUMENTS


I eat sushi. ✓

I eat sushi you. ???

I sleep sushi ???

I give sushi ???

Subcategorization Violations

I drink sushi ?

Selectional Preference Violation

- **Subcategorization**

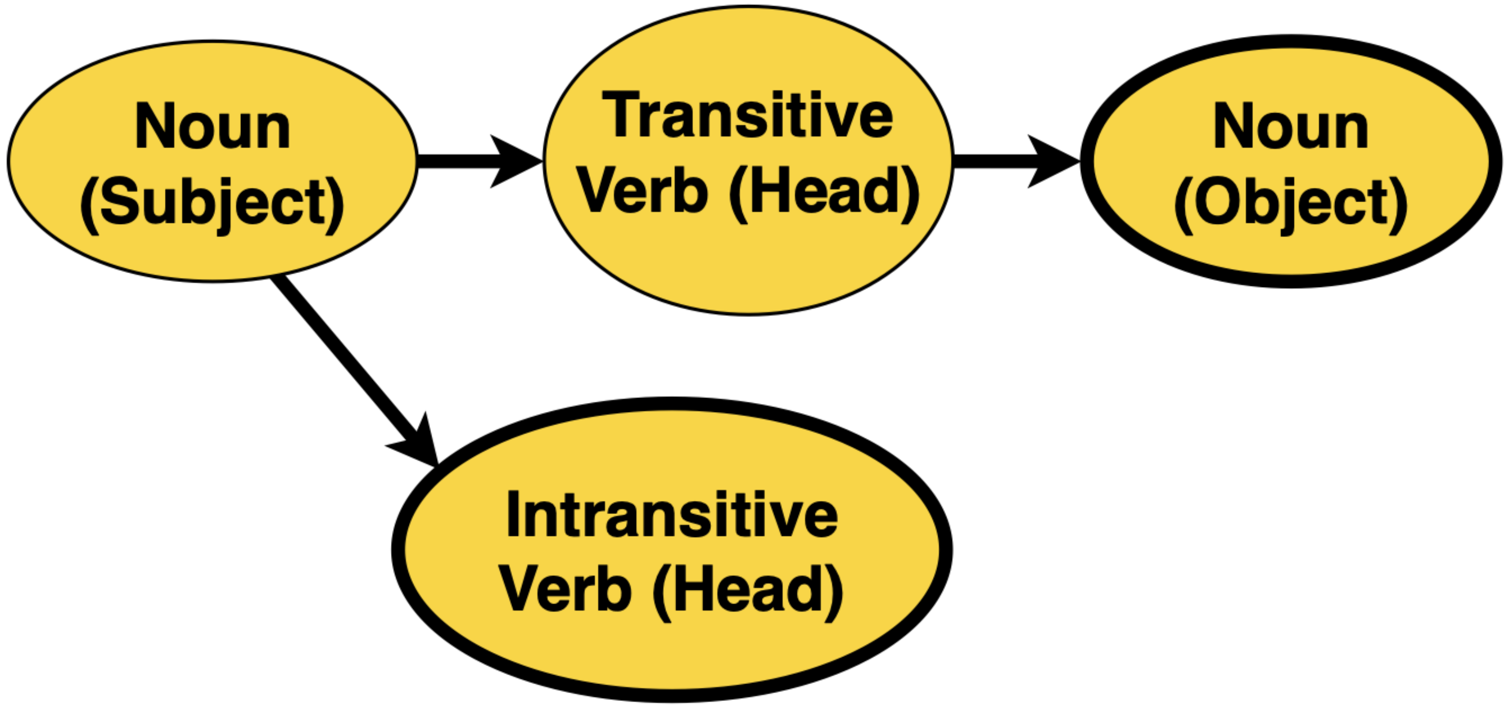
(purely syntactic: what set of arguments do words take?)
Intransitive verbs (sleep) take only a subject.
Transitive verbs (eat) take a subject and one (direct) object. Ditransitive verbs (give) take a subject, direct object and indirect object.

- **Selectional preferences**

(semantic: what types of arguments do words tend to take)

- The object of eat should be edible.

## A BETTER FSA


Noun (Subject)
Transitive Verb (Head)
Noun (Object)
Intransitive Verb (Head)

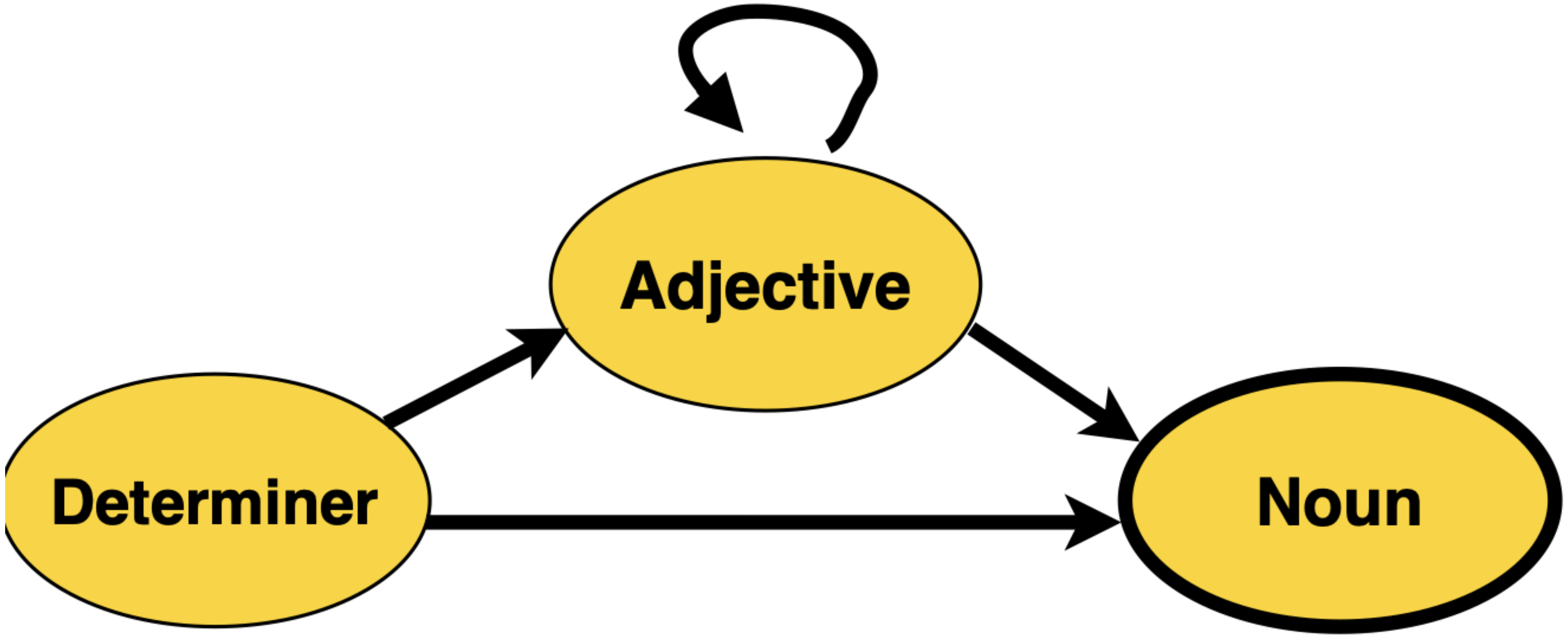
## LANGUAGE IS RECURSIVE

***the ball***
***the big ball***
***the big, red ball***
***the big, red, heavy ball***
**....**

Adjectives can **modify** nouns.

The **number of modifiers (aka adjuncts)** a word can have is (in theory) **unlimited**.

## ANOTHER FSA

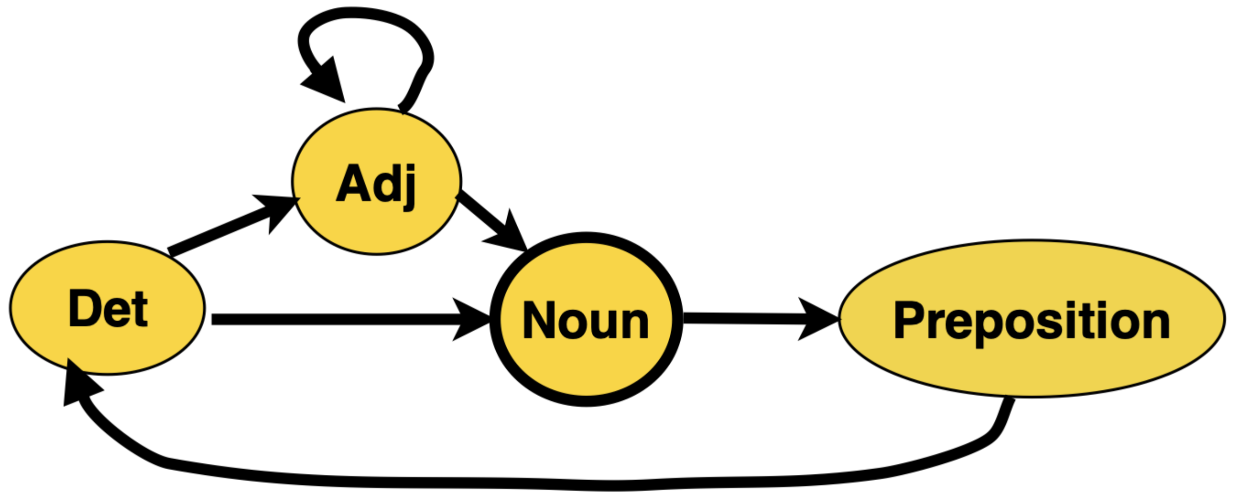


## RECURSION CAN BE MORE COMPLEX

the ball
the ball in the garden
the ball in the garden behind the house
the ball in the garden behind the house next to the school

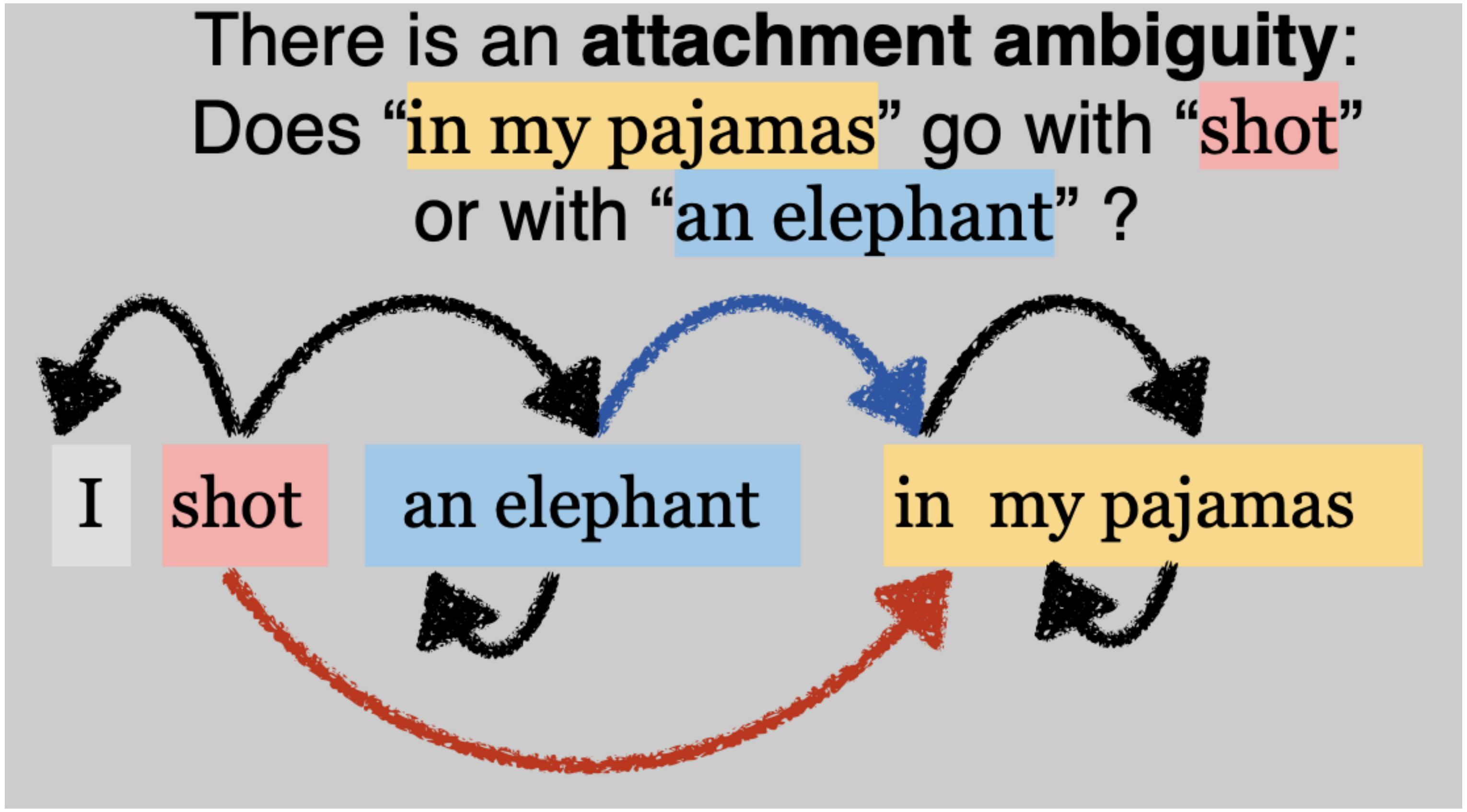
....

## YET ANOTHER FSA

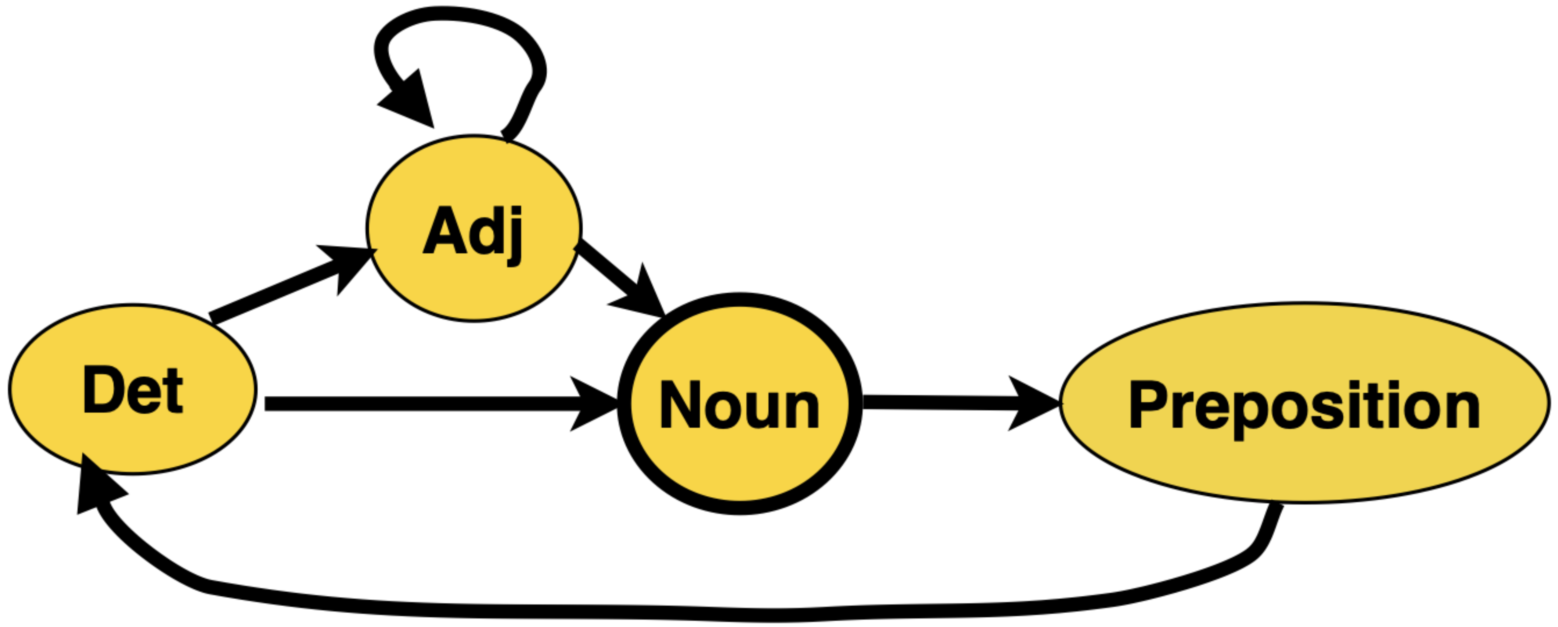


So, why do we need anything
beyond regular (finite-state) grammars?

# WHAT DOES THIS SENTENCE MEAN?


There is an **attachment ambiguity**:
Does “in my pajamas” go with “shot”
or with “an elephant” ?

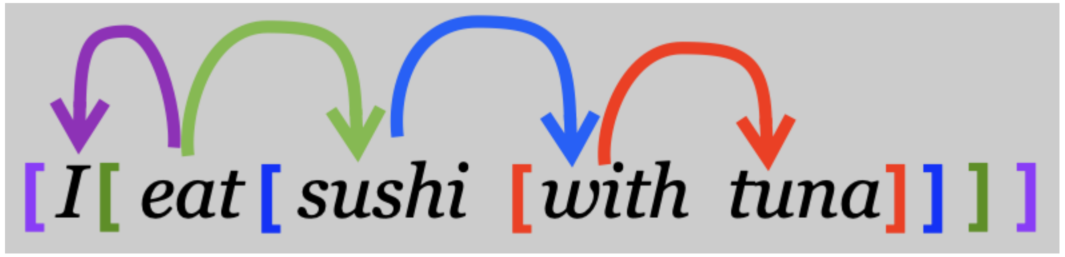
## FSAS DO NOT GENERATE HIERARCHICAL STRUCTURE



## WHAT IS THE STRUCTURE OF A SENTENCE?

- Sentence structure is **hierarchical**:
  A sentence consists of **words** (I, eat, sushi, with, tuna)
- ...which form phrases or **constituents**: “sushi with tuna”

- Sentence structure defines **dependencies** between words or phrases:



## CONTEXT-FREE GRAMMARS FOR NATURAL LANGUAGE

## FORMAL DEFINITIONS

# CONTEXT-FREE GRAMMARS

A CFG is a 4-tuple $\langle \mathbf{N}, \mathbf{\Sigma}, \mathbf{R}, S \rangle$ consisting of:

A finite set of **non-terminals N** (e.g. **N** = {S, NP, VP, PP, Noun, Verb, ....})

A finite set of **terminals Σ** (e.g. **Σ** = {I, you, he, eat, drink, sushi, ball, })

A finite set of **rules R**

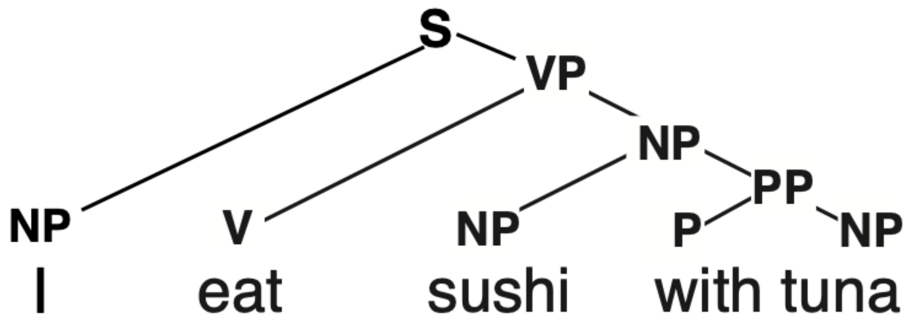
$\mathbf{R} \subseteq \{A \rightarrow \beta$ with left-hand-side (LHS) $A \in \mathbf{N}$ and right-hand-side (RHS) $\beta \in (\mathbf{N} \cup \mathbf{\Sigma})^* \}$

A unique **start symbol** $S \in \mathbf{N}$

# CONTEXT-FREE GRAMMARS (CFGS) DEFINE PHRASE STRUCTURE TREES

```
NP  ⟶  I
NP  ⟶  sushi
NP  ⟶  tuna
NP  ⟶  NP  PP
P   ⟶  with
PP  ⟶  P  NP
S   ⟶  NP  VP
V   ⟶  eat
VP  ⟶  V   NP
```

|  |  |
|---|---|
| NP: | Noun Phrase |
| P: | Preposition |
| S: | Sentence |
| PP: | Prepositional Phrase |
| V: | Verb |
| VP: | Verb Phrase |



**Leaf nodes** (I, eat, …) correspond to the words in the sentence

**Intermediate nodes** (NP, VP, PP) span substrings (= the *yield* of the node), and correspond to nonterminal *constituents*

**The root** spans the entire sentence and is labeled with the start symbol of the grammar (here, S)

## CFGS CAPTURE RECURSION

- Language has simple and complex constituents

(simple: “the garden”, complex: “the garden behind the house”)

- Complex constituents behave just like simple ones.

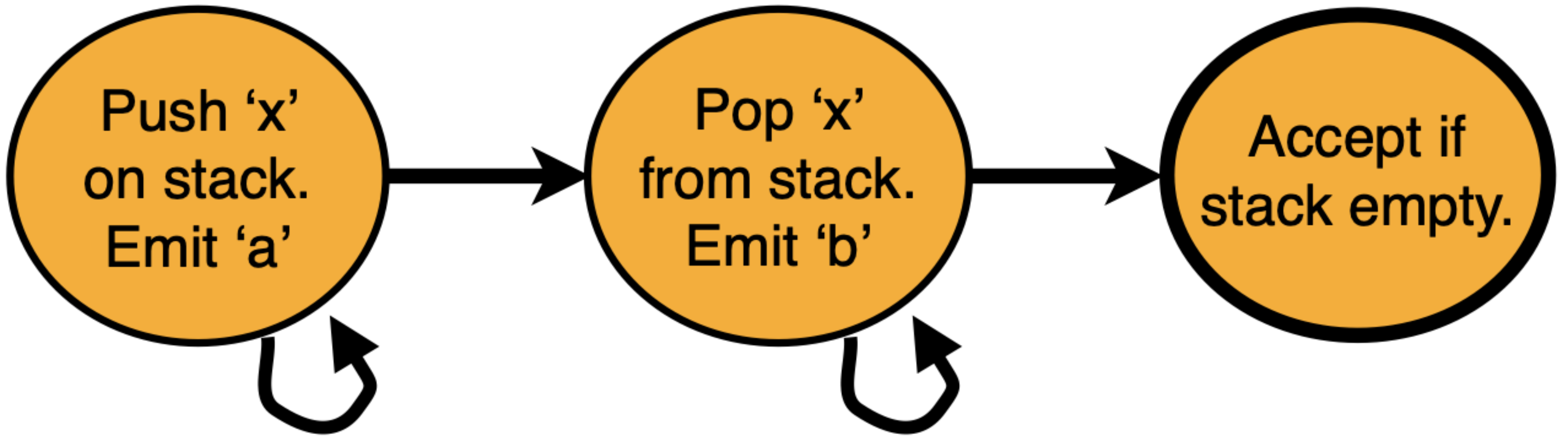
(“behind the house” can always be omitted) CFGs define **nonterminal categories** (e.g. NP) to capture **equivalence classes of constituents.**

**Recursive rules** (where the same nonterminal appears on both sides) generate recursive structures

- **`NP`** → `DT N` (**Simple**, i.e. **non-recursive** NP) ,
- **`NP`** → **`NP`** `PP` (**Complex**, i.e. **recursive**, NP)

## CFGS ARE EQUIVALENT TO PUSHDOWN AUTOMATA (PDAS)

- PDAs are FSAs with an additional stack:
- Emit a symbol and push/pop a symbol from the stack



This is equivalent to the following CFG:

```
S  → a S b
S  → a b
```

## GENERATING ANBN

| Action | Stack | String |
|---|---|---|
| 1. Push x on stack. Emit a. | x | a |
| 2. Push x on stack. Emit a. | xx | aa |
| 3. Push x on stack. Emit a. | xxx | aaa |
| 4. Push x on stack. Emit a. | xxxx | aaaa |
| 5. Pop x off stack. Emit b. | xxx | aaaab |
| 6. Pop x off stack. Emit b. | xx | aaaabb |
| 7. Pop x off stack. Emit b. | x | aaaabbb |
| 8. Pop x off stack. Emit b |  | aaaabbbb |

## ENCODING LINGUISTIC PRINCIPLES IN A CFG

## IS STRING A A CONSTITUENT?
## [SHOULD MY GRAMMAR/PARSE TREE HAVE A NONTERMINAL FOR A?]

He talks [in class].

Substitution test:
Can α be replaced by a single word? He talks [there].

Movement test:
Can α be moved around in the sentence? [In class], he talks.

Answer test:
Can α be the answer to a question? Where does he talk? - [In class].

## CONSTITUENTS: HEADS AND DEPENDENTS

There are different kinds of constituents:

**Noun phrases:** the man, a girl with glasses, Illinois

**Prepositional phrases:** with glasses, in the garden

**Verb phrases:** eat sushi, sleep, sleep soundly

Every phrase has one **head**:

NB: this is an oversimplification. Some phrases (**John, Kim and Mary**) have multiple heads, others (I like coffee and [**you tea**]) perhaps don't even have a head

**Noun phrases:** the man, a girl with glasses, Illinois

**Prepositional phrases:** with glasses, in the garden

**Verb phrases:** eat sushi, sleep, sleep soundly

The other parts are its **dependents**.

Dependents are either **arguments** or **adjuncts**

NB: some linguists think the argument-adjunct distinction isn't always clear-cut, and there are some cases that could be treated as either, or something in-between

## ARGUMENTS ARE OBLIGATORY

| | |
|---|---|
| Words **subcategorize** for specific sets of arguments: | Transitive verbs (sbj + obj): [John] likes [Mary] |
| | The set/list of arguments is called a **subcat frame** |
| All **arguments** have to be **present**: | *[John] likes. *likes [Mary]. |
| No argument slot can be **occupied multiple times**: | *[John] [Peter] likes [Ann] [Mary]. |
| Words can have **multiple subcat frames**: | Transitive eat (sbj + obj): [John] eats [sushi]. |
| | Intransitive eat (sbj): [John] eats |

## ADJUNCTS (MODIFIERS) ARE OPTIONAL

- Adverbs, PPs and adjectives can be adjuncts
- Adverbs: John runs [fast].
  a [very] heavy book.
- PPs: John runs [in the gym]. the book [on the table]
- Adjectives: a [heavy] book
- There can be an arbitrary number of adjuncts:
- John saw Mary.
  John saw Mary [yesterday].
  John saw Mary [yesterday] [in town]
  John saw Mary [yesterday] [in town] [during lunch] [Perhaps] John saw Mary [yesterday] [in town] [during lunch]

# HEADS, ARGUMENTS AND ADJUNCTS IN CFGS

How do we define CFGs that…
… identify heads and
… distinguish between arguments and adjuncts?

We have to make additional assumptions about the rules that we allow.

- Important: these are not formal/mathematical constraints, but aim to capture linguistic principles.
- A more fleshed out version of what we will describe here is known as "X-bar Theory" (Chomsky, 1970)

Phrase structure trees that conform to these assumptions can easily be translated to dependency trees

## HEADS, ARGUMENTS AND ADJUNCTS IN CFGS

- To identify **heads**:
- We assume that each RHS has one head child, e.g.

`VP → Verb NP` (Verbs are heads of VPs)

`NP → Det Noun` (Nouns are heads of NPs)

`S → NP VP` (VPs are heads of sentences)

- Exception: This does not work well for coordination:

`VP → VP conj VP`

- We need to define for each nonterminal in our grammar (S, NP, VP, ...) which nonterminals (or terminals) can be used as its head children.

## HEADS, ARGUMENTS AND ADJUNCTS IN CFGS

To distinguish between arguments and adjuncts, assume that each is introduced by different rules.

**Argument rules:** The head has a different category from the parent:

**S → NP VP** (the NP is an argument of the VP [verb])

**VP → Verb NP** (the NP is an argument of the verb) This captures that arguments are obligatory.

**Adjunct rules ("Chomsky adjunction"):** The head has the same category as the parent:

**VP→VPPP** (thePPisanadjunctoftheVP)

This captures that adjuncts are optional and that their number is unrestricted.

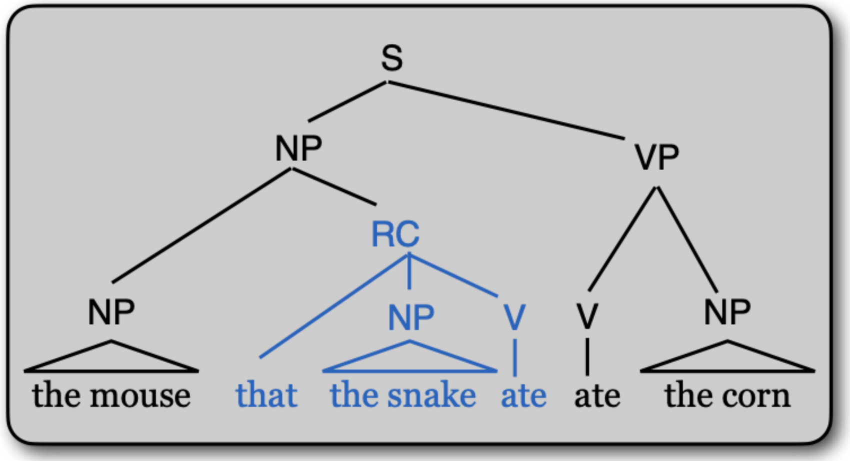
## CFGS AND UNBOUNDED RECURSION

## UNBOUNDED RECURSION: CFGS AND CENTER EMBEDDING

The mouse ate the corn.

The mouse that the snake ate ate the corn.

```
S    → NP   VP
VP   → V    NP
NP   → Det  N
NP   → NP   RC
RC   → that NP V
Det  → the
N    → mouse|corn|snake
V    → ate
```



# UNBOUNDED RECURSION: CFGS AND CENTER EMBEDDING

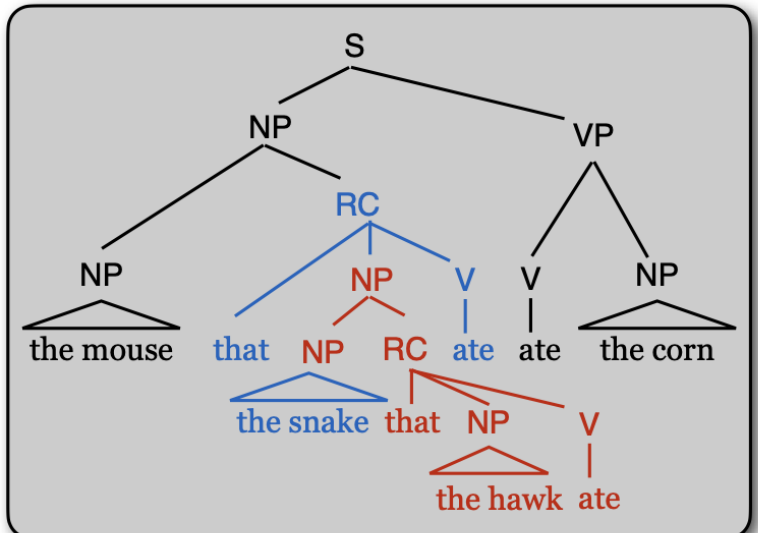
The mouse ate the corn.

The mouse that the snake ate ate the corn.

The mouse that the snake that the hawk ate ate ate the corn.

…

```
S    → NP   VP
VP   → V    NP
NP   → Det  N
NP   → NP   RC
RC   → that NP V
Det  → the
N    → mouse|corn|snake
V    → ate
```



## UNBOUNDED RECURSION: CFGS AND CENTER EMBEDDING

- These sentences are unacceptable, but formally, they are all
- grammatical, because they are generated by the recursive rules
- required for even just one relative clause:

-
  ```
  NP  ⟶ NP RC
  RC  ⟶ that NP V
  ```
- **Problem:** CFGs are not able to capture **bounded recursion.** (bounded = "only embed one or two relative clauses").
- To deal with this discrepancy between what the grammar predicts to be grammatical, and what humans consider grammatical, linguists distinguish between a speaker's **competence** (grammatical knowledge) and **performance** (processing and memory limitations)