LECTURE 19: LINGUISTICALLY EXPRESSIVE GRAMMARS

Mehmet Can Yavuz, PhD.

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

PART 1: GRAMMAR IS IN NLP: WHAT AND WHY

WHAT IS GRAMMAR?

Grammar formalisms

(= linguists' 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

(= linguists' programs)

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

(NB: any practical parser will need to also have a model/scoring function to identify which grammatical analysis should be assigned to a given sentence)

WHY STUDY GRAMMAR?

Linguistic questions:

- What kind of constructions occur in natural language(s)?

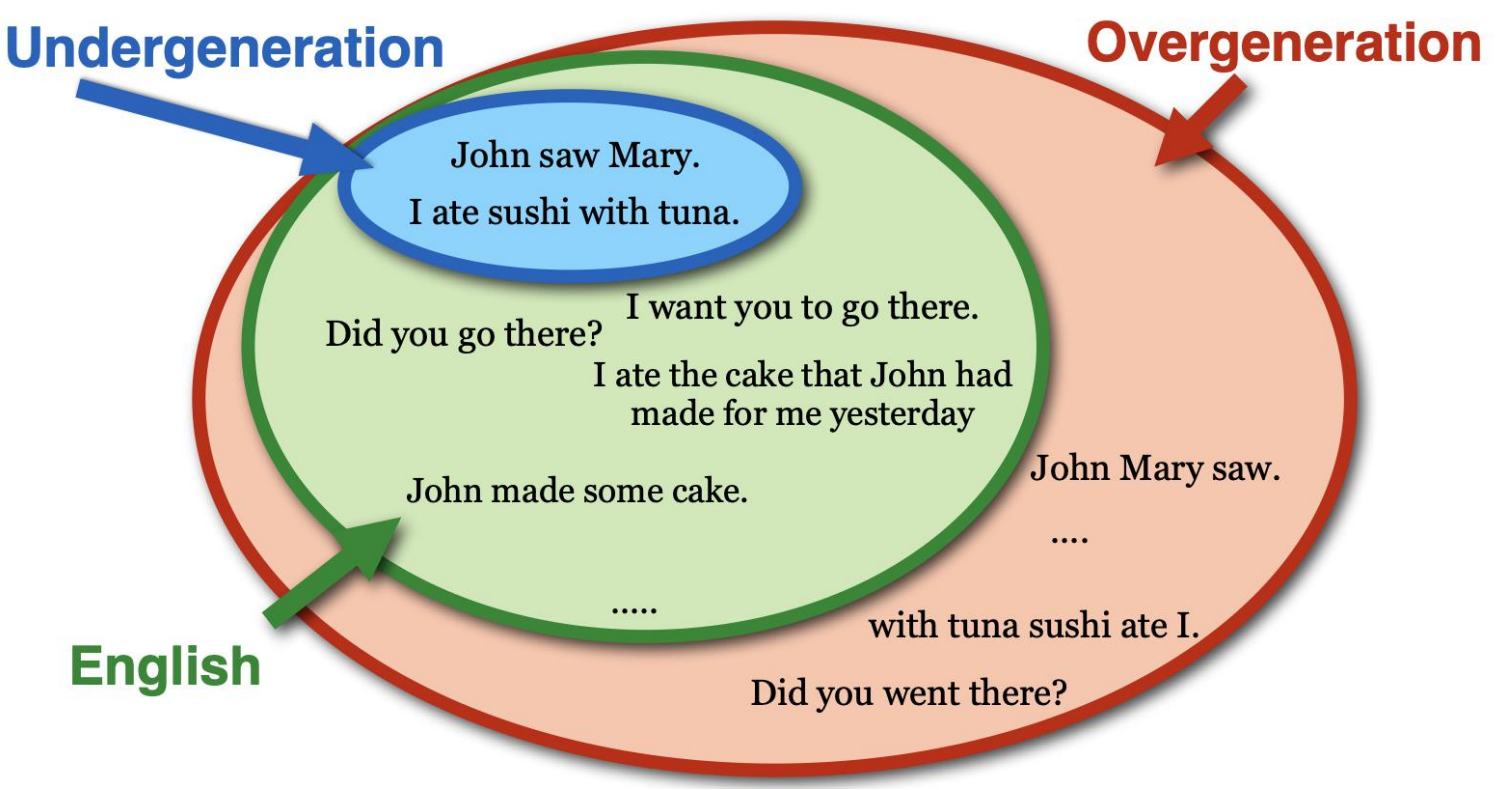
Formal questions:

- Can we define formalisms that allow us to characterize which strings belong to a language?
- Those formalisms have appropriate weak generative capacity
- Can we define formalisms that allow us to map sentences to their appropriate structures?
- Those formalisms have appropriate strong generative capacity

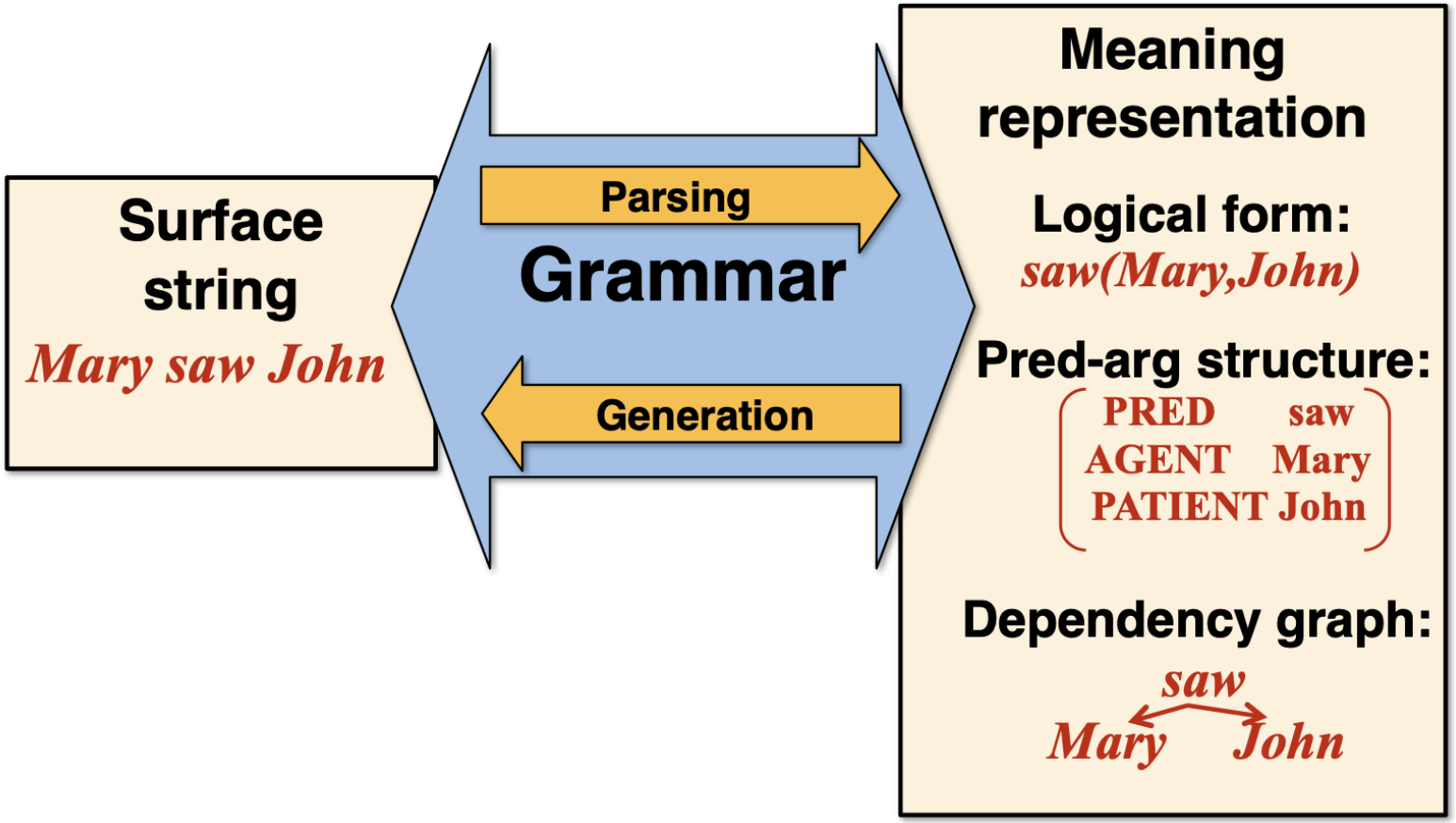
Practical applications (Syntactic/Semantic Parsing):

- Can we identify the grammatical structure of sentences?
- Can we translate sentences to appropriate meaning representations?

CAN WE DEFINE A PROGRAM THAT GENERATES ALL ENGLISH SENTENCES?



SYNTAX AS AN INTERFACE TO SEMANTICS



GRAMMAR FORMALISMS

Formalisms provide a formal **language** in which linguistic theories can be expressed and implemented

Formalisms define **elementary objects** (trees, strings, feature structures)
and **recursive operations** which generate complex objects from simple objects.

Different formalisms may impose different **constraints** (e.g. on the kinds of dependencies they can capture)

WHAT MAKES A FORMALISM "EXPRESSIVE"?

"Expressive" formalisms are richer than context-free grammars.

Different formalisms use different mechanisms, data structures and operations to **go beyond CFGs**

EXAMPLES OF EXPRESSIVE GRAMMAR FORMALISMS

Tree-adjoining Grammar (TAG):	Fragments of phrase-structure trees
Combinatory Categorial Grammar (CCG):	Syntactic categories paired with meaning representations
Lexical-functional Grammar (LFG):	Annotated phrase-structure trees (c-structure) linked to feature structures (f-structure)
Head-Driven Phrase Structure Grammar(HPSG):	Complex feature structures (Attribute-value matrices)

PART 2: WHY GO BEYOND CFGS?

THE DEPENDENCIES SO FAR:

Arguments:
Verbs take arguments: subject, object, complements, … **Heads subcategorize for their arguments**

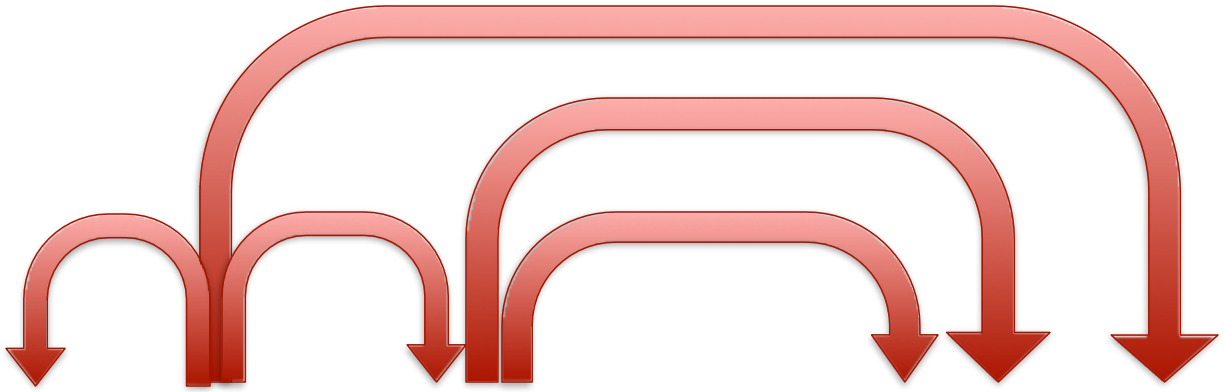
Adjuncts/Modifiers:
Adjectives modify nouns, adverbs modify VPs or adjectives, PPs modify NPs or VPs
Modifiers subcategorize for the head

Typically, these are *local* dependencies: they can be expressed *within individual CFG rules*

VP → Adv Verb NP

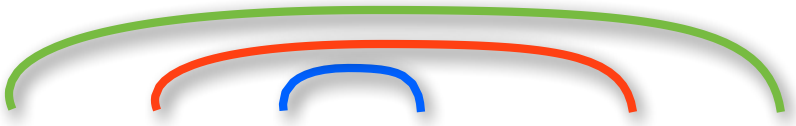
CONTEXT-FREE GRAMMARS

- CFGs capture only **nested** dependencies
  - The dependency graph is a **tree**
  - The dependencies **do not cross**



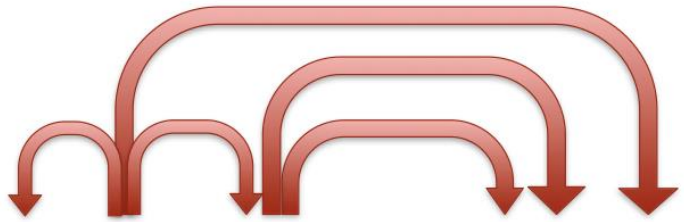
GERMAN: CENTER EMBEDDING

- ...daß ich [Hans schwimmen] sah
- ...that I Hans swim saw
- *...that I saw [Hans swim]*

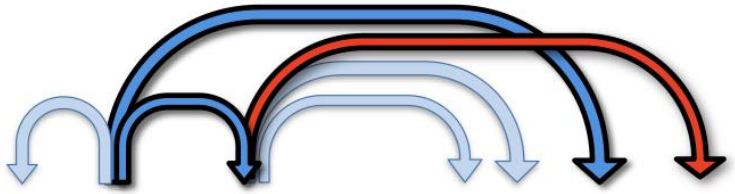
- ...daß ich [Maria [Hans schwimmen] helfen] sah
- ...that I Maria Hans swim help saw
- *...that I saw [Mary help [Hans swim]]*
- 
- ...daß ich [Anna [Maria [Hans schwimmen] helfen] lassen] sah
- ...that I Anna Maria Hans swim help let saw
- *...that I saw [Anna let [Mary help [Hans swim]]]*

DEPENDENCY STRUCTURES IN GENERAL

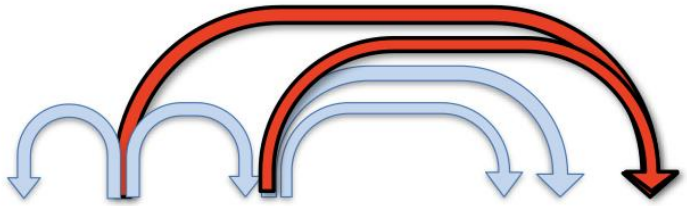
Nested (projective) dependency trees (CFGs)



Non-projective dependency trees

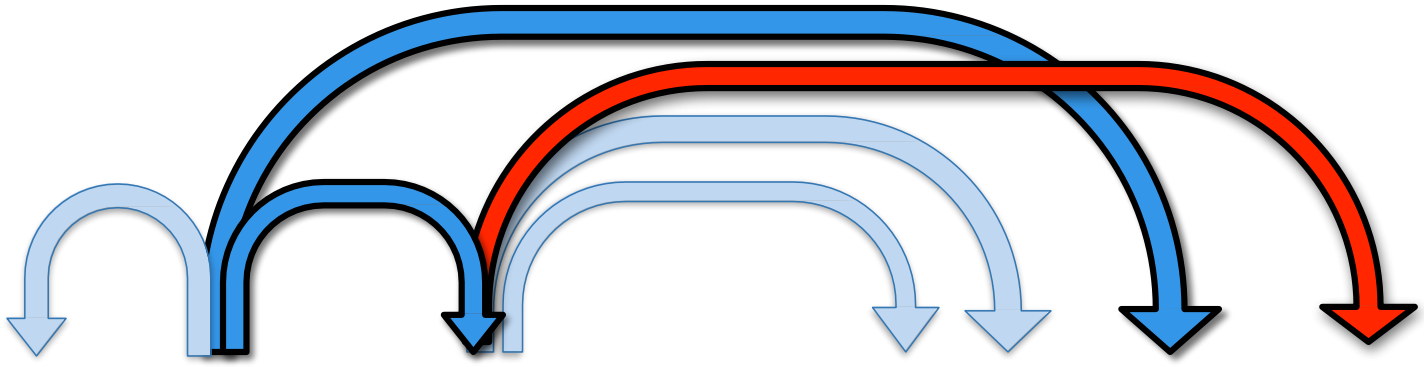


Non-local dependency graphs



BEYOND CFGS: NONPROJECTIVE DEPENDENCIES

- Dependencies form a **tree with crossing branches**



DUTCH: CROSS-SERIAL DEPENDENCIES

- ...dat ik Hans zag zwemmen
- ...that I Hans saw swim
- ...that I saw [Hans swim]
- ...dat ik Maria Hans zag helpen zwemmen
- ...that I Maria Hans saw help swim
- *...that I saw [Mary help [Hans swim]]*

- ...dat ik Anna Maria Hans zag laten helpen zwemmen
- ...that I Anna Maria Hans saw let help swim
- *...that I saw [Anna let [Mary help [Hans swim]]]*

Such cross-serial dependencies require *mildly context-sensitive grammars*

OTHER CROSSING (NON-PROJECTIVE) DEPENDENCIES

(Non-local) scrambling: In a sentence with multiple verbs, the argument of a verb appears in a different clause from that which contains the verb (arises in languages with freer word order than English)

- ***Die Pizza*** *hat Klaus versprochen zu* ***bringen***
- The pizza has Klaus promised to bring
- *Klaus has promised to bring the pizza*

Extraposition: Here, a modifier of the subject NP is moved to the end of the sentence

- *The* ***guy*** *is coming* ***who is wearing a hat***
- Compare with the non-extraposed variant
- *The* ***[guy [who is wearing a hat]]*** *is coming*

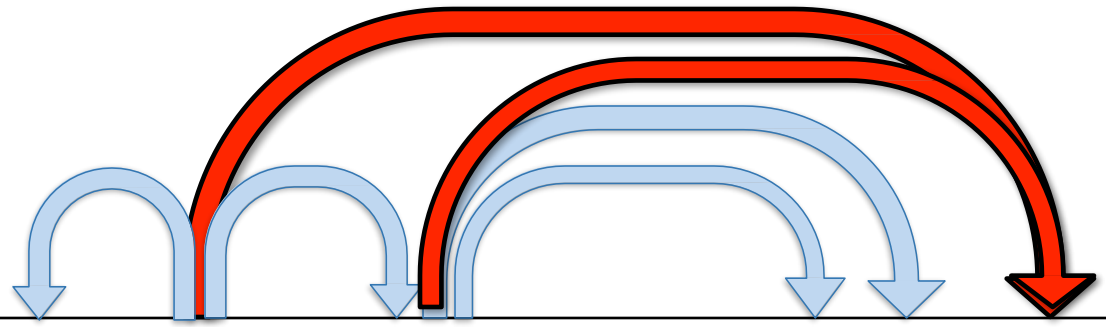
Topicalization: Here, the argument of the embedded verb is moved to the front of the sentence.

- ***Cheeseburgers****, I [thought [he* ***likes****]]*

BEYOND CFGS: NONLOCAL DEPENDENCIES

- Dependencies form a **DAG**
  (a node may have **multiple incoming edges**)
- Arise in the following constructions:
  - **Control** *(**He** has **promised** me to **go**)*, **raising** (**He seems** to **go**)
  - **Wh-movement** *(the **man** who you **saw** yesterday **is** here again)*,
  - **Non-constituent** coordination

    (right-node raising, gapping, argument-cluster coordination)



WH-EXTRACTION (E.G. IN ENGLISH)

Relative clauses:

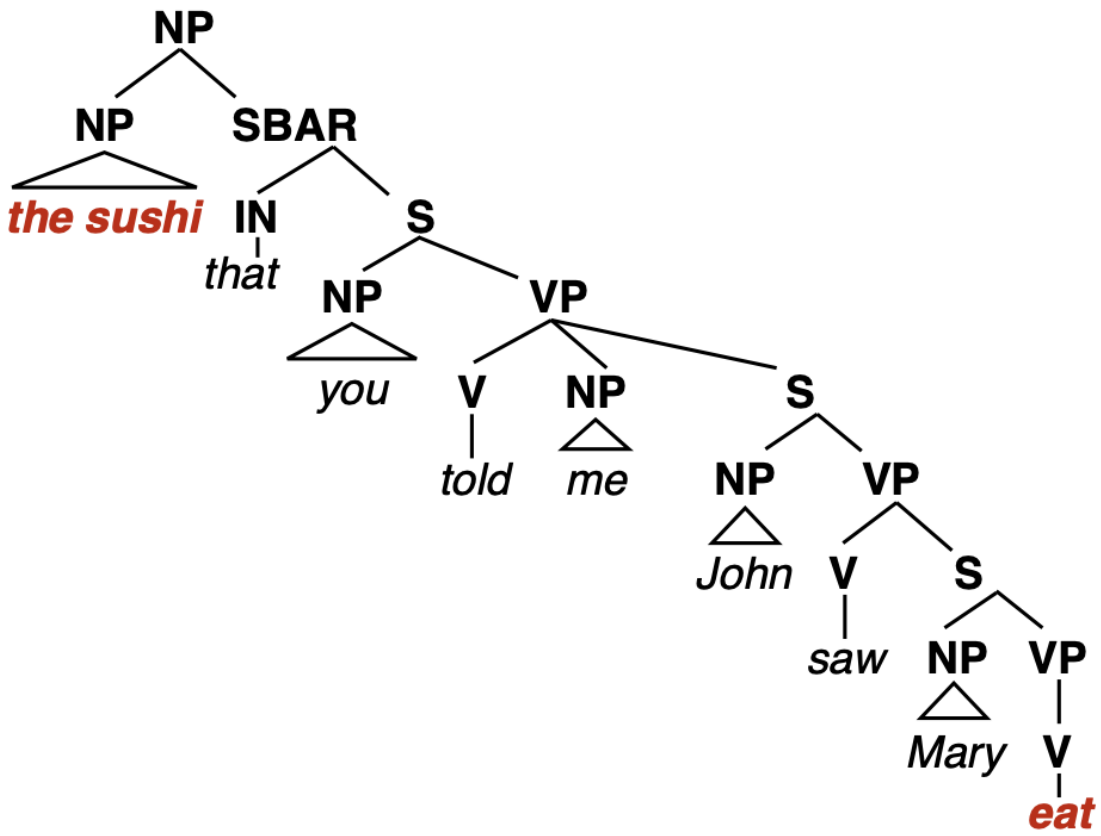
the sushi that [*you told me* [*John saw* [*Mary eat*]]]’

Wh-Questions:
‘what [*did you tell me* [*John saw* [*Mary eat*]]]?’

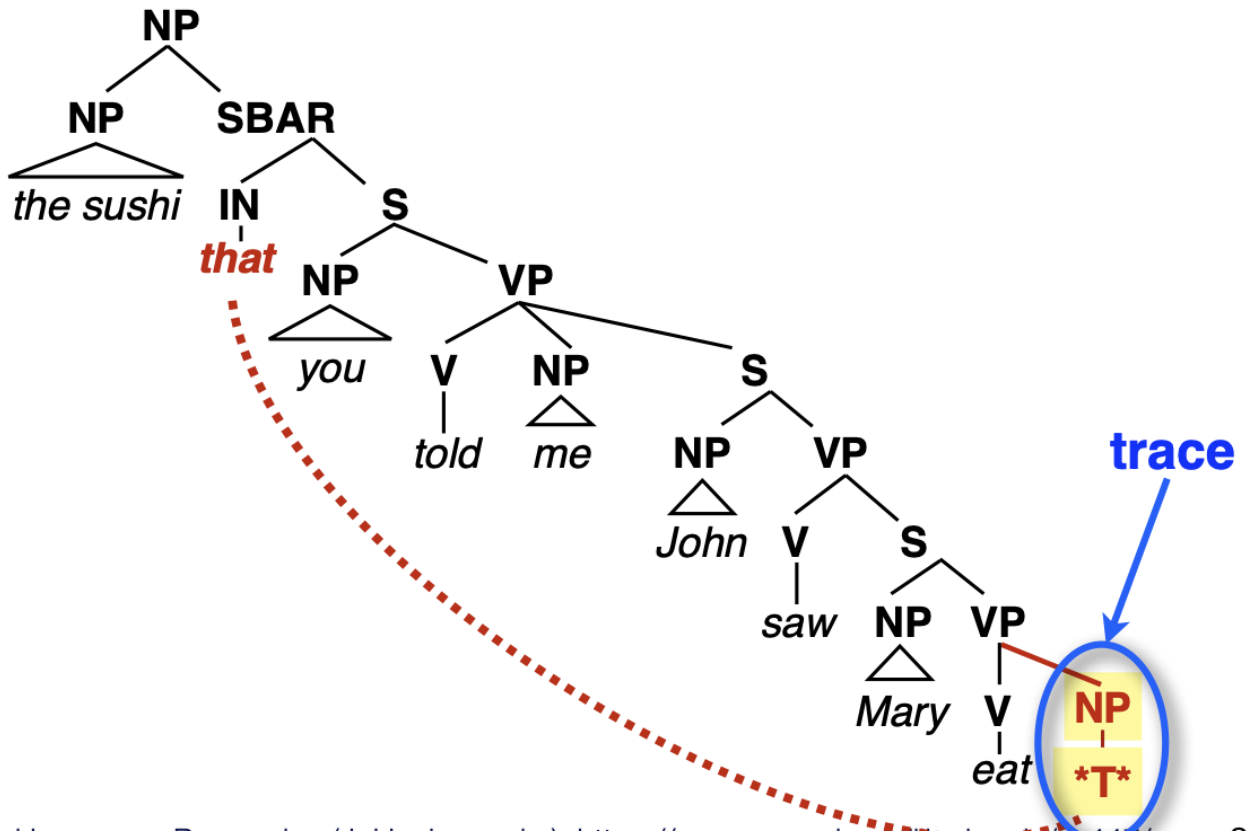
Wh-questions (what, who, …) and relative clauses contain so-called *unbounded* nonlocal dependencies because the verb that subcategorizes for the moved NP may be arbitrarily deeply embedded in the tree

Linguists call this phenomenon **wh-extraction** (wh-movement).

AS A PHRASE STRUCTURE TREE:



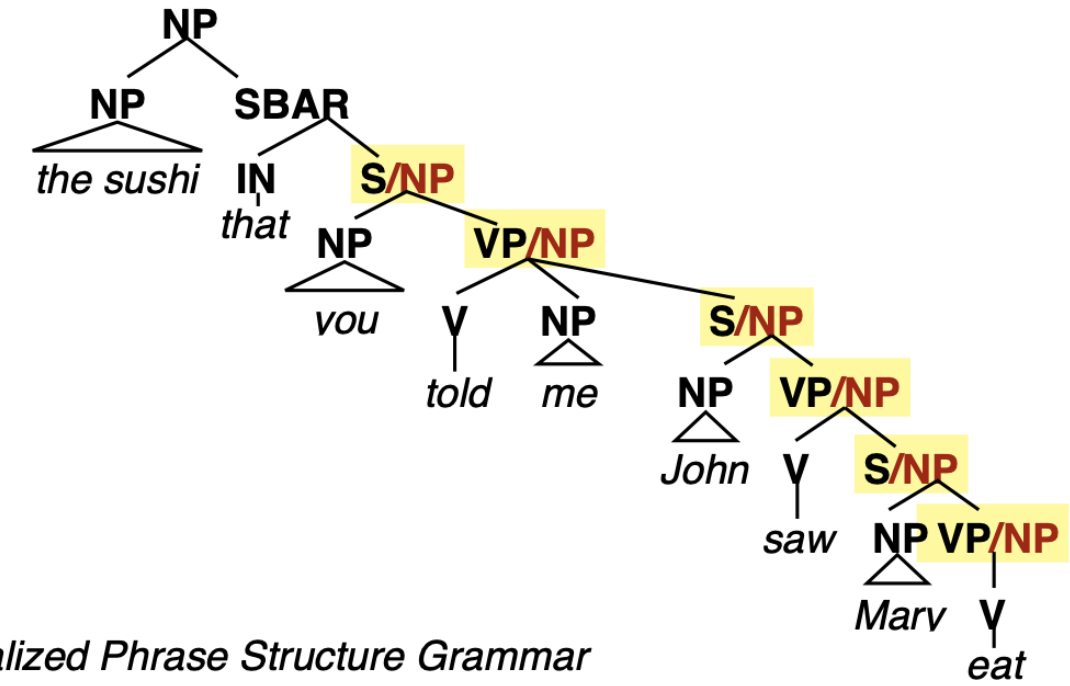
THE TRACE ANALYSIS OF WH-EXTRACTION



SLASH CATEGORIES FOR WH-EXTRACTION

Because only one element can be extracted,
we can use **slash categories.**

This is still a CFG: the set of nonterminals is finite.



Generalized Phrase Structure Grammar (GPSG), Gazdar et al. (1985)