LX 321/521/621 *Syntax: Introduction to Sentential Structure*

January 19, 2024

1 Stuff we know

(1) a. Sue said that Mary borrowed my guitar.
    b. Sue said Mary borrowed my guitar.

(2) a. What did Sue say that Mary borrowed?
    b. What did Sue say Mary borrowed?

  a. Who did Sue say borrowed my guitar?
  b. * Who did Sue say that borrowed my guitar?

Poverty of the stimulus.

2 Esperanto

Martin Haspelmath (1997). *Indefinite Pronouns*, pp. 24–25.

> Then the first important generalization that we can make is that indefinite pronouns are as a rule DERIVED forms. That this is not a trivial observation can be seen from the artificial language Esperanto, designed by Ludwik Zamenhof in 1887. As a rule, Esperanto grammar follows the typological design of Standard Average European, eliminating irregularities and 'useless' features like gender and agreement. There are two indefinite series in Esperanto, which are clearly related to interrogative pronouns, much as in many natural languages:

(27)

Esperanto	interrogative	indefinite	negative
person	kiu	iu	neniu
thing	kio	io	nenio
property	kia	ia	nenia
place	kia	ie	nenie
time	kiam	iam	neniam
manner	kiel	iel	neniel

> ... However, natural languages that are structured like Esperanto in this respect are virtually unattested. I am not aware of a clear case in which an indefinite pronoun is formally unmarked with respect to a marked interrogative pronoun. With respect to its indefinite pronoun system, Esperanto is thus probably not a possible human language. ...

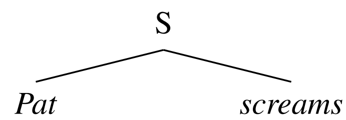
3 Prescriptive rules

(3) Prepositions are things you should know to never end a sentence with.

4 Doing science

Generative grammar, starting from S, generating all and only the acceptable sentences.

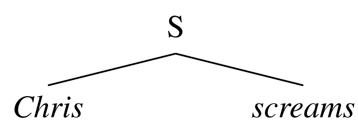
(4) a. S → *Pat screams*



- Propose a theory
- Identify the predictions
- Test the predictions against the observed data
- If they fail, revise and return to the first step

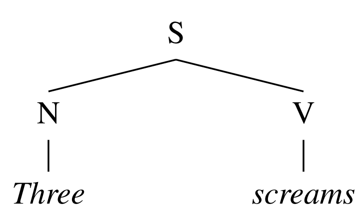
Chris screams is also English.

(5) a. S → *Pat screams*
b. S → *Chris screams*



Tracy screams is also an English sentence. As are *One screams*, *Two screams*. And *Pat sleeps*. But not *Pat Tracy* or *Screams sleeps*.

(6) a. S → N V
b. N → *Chris*
c. N → *Pat*
d. N → *Tracy*
e. N → *One*
f. N → *Two*
g. N → *Three*
...
h. N → *Two-Hundred-Forty-Eight*
i. V → *screams*
j. V → *sleeps*

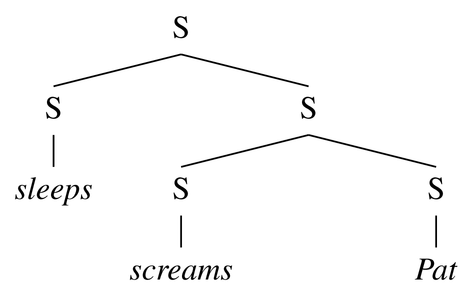


Structural rules (grammar), lexical rules (word categories).

- Observe a new sentence pattern that our existing grammar doesn't predict
- Modify the grammar so it does predict the new pattern
- See if this allows for any generalizations that can simplify the grammar
- See what predictions the new grammar makes
- Check to see if those predictions are borne out, repeat

Why is this grammar terrible?

(7) a. S $\rightarrow$ S S
b. S $\rightarrow$ *Chris*
c. S $\rightarrow$ *Pat*
...
d. S $\rightarrow$ *Two-Hundred-Forty-Eight*
e. S $\rightarrow$ *screams*
f. S $\rightarrow$ *sleeps*



5 Parts of speech

Any chance we have of creating a successful generative grammar will rely on being able to distinguish these distributional classes. These are the traditional "parts of speech" and have traditional names.

How do we know what class a word is in? They are defined distributionally, so we can use distributional tests.

5.1 Content words vs. function words

- **Content words:**
  - words that denote concepts, supplying the core notional content of the phrases they appear in.
  - Usually "open class"—new ones are frequently invented or borrowed.
  - Nouns, Verbs, Adjectives, Adverbs, Prepositions
- **Function words:**
  - words that have no/little conceptual content—instead, signal relations between elements in the sentence, conveying *grammatical* meanings.

  - usually closed class—it's very rare for new ones to be coined or borrowed
  - Determiners *the*, *that*, *this*, *a*
  - Modals *will*, *might*, *could*, *should*, ...
  - Complementizers *whether*, *if*, *that*, *for*, ...
  - Auxiliaries *have*, *be*
  - ...

5.2 Diagnosing content word categories

5.2.1 Noun [N]

- Can be the subject of the sentence (if a mass or plural noun)
- Can be the only thing following determiners like *the*, *some*, *any*, *this*, *a*, including possessives like *my*, *her*.
- Can usually be pluralized with *–s* (or some irregular plural morpheme)
- Can be modified by adjectives

5.2.2 Verb [V]

- Can be marked for past tense
- Can be suffixed with *–ing* (as long as the verb doesn't denote a state)
- Can be modified by adverbs like *again*, *often*
- Can follow modals like *can*, *will*, *may*, ... (or their negative versions), and the infinitive *to*
- Some can take the prefix *–un* (meaning 'reverse a process')

5.2.3 Adjective [Adj]

- Can directly follow *seem*
- Can appear between a determiner and a noun (be careful of compounds)
- Often end in *–y*, *–ish*, or *–ous*.
- Can be modified by *very* and *extremely* (but be alert, so can adverbs)
- Some can take the prefix *un–* (meaning 'not')

5.2.4 Adverbs [Adv]

- Can modify a verb and/or an adjective
- Often end in *–ly*
- Can be modified by *very* or *extremely* (but be alert, so can adjectives)

5.2.5 Preposition [P]

- The only non-verb category that can systematically take an object directly
- Those that denote locations or directions can be modified by *right* or *straight* (though some dialects allow *right*+Adj too)
- Occupies a strange space between open-class and closed-class. On the one hand, it's hard to imagine inventing a new preposition, like "ascreen" (facing the screen of something that has a screen), as in "Pat stood ascreen the departure monitors"—and yet we have things like "aboard" (in/on some kind of vehicle) or "chez" (at the home of).

5.2.6 Modals, auxiliaries, determiners

These are the closed-class things, not as much necessary to test them as to know them. But it is still distributional, you know what these are by virtue of what they can replace in sentences.

- Determiner [D] (including articles): *the*, *a*, demonstratives(?) *this*, possessives(?) *my*, quantifiers(?) *every*
- Auxiliary [Aux]: *have*, *be*, *do*, modals(?) *can*, *might*, *shall*
- Complementizer [C]: *that*, *for*, *whether*, *if*
- Negation [Neg]: *not* (which, admittedly, does not seem like much of a "class")

5.2.7 Examples

(8) a. John was <u>killed</u>.
  b. John was <u>skilled</u>.
  c. medic
  d. medical
  e. remove
  f. removal
  g. The bike is <u>behind the shed</u>.
  h. magically
  i. kingly
  j. The exercise is <u>boring</u> the students.
  k. The exercise is <u>boring</u> for the students.
  l. We are galloping <u>around</u> the room.

6 Phrase structure rules and trees

A phrase structure rule (a.k.a. "rewrite rule") looks like:

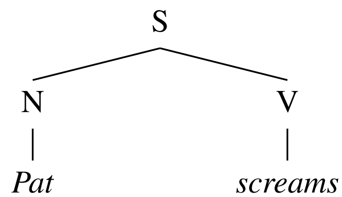
(9) $X \rightarrow Y_1\ Y_2\ Y_3 \ldots\ Y_n$

On the left is the "parent" and on the right are the "children."
There are two kinds of phrase structure rules we'll be concerned with.

- **Lexical rules**. Define category of a word. N $\rightarrow$ *Pat*.
- **Structural rules**. Define pattern of categories that form a larger expression. S $\rightarrow$ N V.

A set of phrase structure rules is said to *generate* tree diagrams.

(10) a. S $\rightarrow$ N V
b. N $\rightarrow$ *Pat*
c. V $\rightarrow$ *screams*

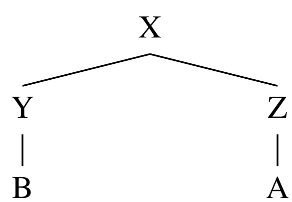


6.1 Hierarchical structures

Words can be grouped ("kicks Chris") and the grouping itself can have a distributional class ("kicks Chris" distributes as if it were a V—it is a V phrase).

6.2 Names for tree parts

There are standard terms for the parts of the tree and for their relations. They are generally derived from some combination of genealogy and actual trees. Some things are just graph-theoretic.[1]

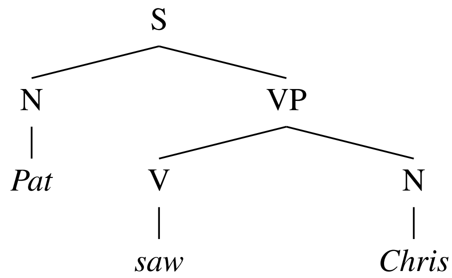


- Each point at the end of a branch is a **node** (X, Y, Z, A, B). (**Branches** connect nodes.)
- The node at the top is the **root node** (X).
- The nodes along the bottom border are the **leaf nodes** (A, B).
- The node at the other end of an upward branch is a **parent node** (Z is the parent node of A).
- The node at the other end of a downward branch is a **child node** (A is the child node of Z).
- Nodes that share the same parent node are **sibling nodes** (Y and Z are sibling nodes).

So are A and B sibling nodes? Are B and Z sibling nodes?
Note: Although you can imagine what they might refer to, nodes are not described as being "grandparents" or "cousins." Just parents, children, siblings.

[1]Often you will find some of these relations referred to in an arbitrary feminine form (mother, daughter, sister), but that's just a historical tradition and seems like an extra thing to try to retain for no real reason. If you see "sister node" in a paper or book, that's just a name people often used for "sibling node."

(11)



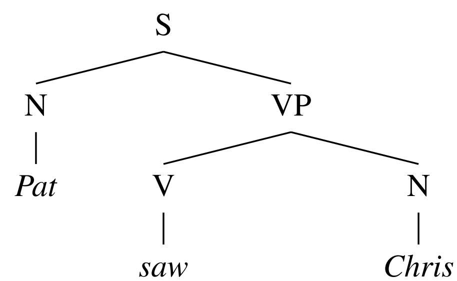
(12)
a. What is the root node?
b. What are the leaf nodes?
c. Which node is the parent of *Pat*?
d. Which node is the child of *Pat*?
e. Which node is the parent of the rightmost N?
f. Which node is the child of rightmost N?
g. Which node is the siblings of the leftmost N?
h. Which node is the siblings of *saw*?
i. Which nodes are children of S?

Do the rules in (13) generate the tree in (14)?

(13)

Grammar	
S →	N VP
VP →	V N
V →	*saw*
N →	*Pat*
N →	*Chris*

(14)

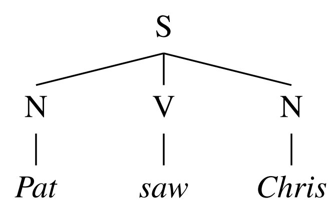


Do the rules in (15) generate the tree in (16)?

(15)

Grammar	
S →	N V N
V →	*saw*
N →	*Pat*
N →	*Chris*

(16)



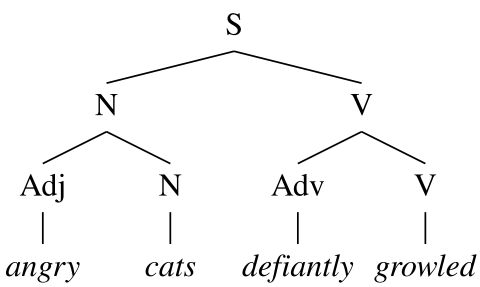
Sometimes a set of PSRs assigns more than one tree to the same string of words.
(This is a good thing, because this allows for a theory of certain kinds of ambiguity.)
How many tree diagrams does (17) generate for the sentence *Pat pats cats at Alexanderplatz*?

(17)

Grammar	
S →	N VP
VP →	VP PP
VP →	V N
N →	N PP
PP →	P N
V →	*pats*
N →	*Pat*
N →	*cats*
N →	*Alexanderplatz*
P →	*at*

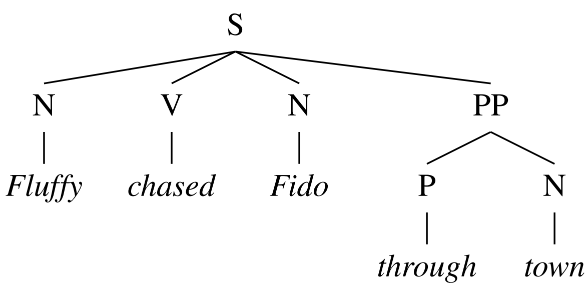
Let's walk through an example of how to go the other way, to take a tree and come up with the PSRs that can generate it. What rules generate (18)?

(18)



And what rules generate (19)?

(19)



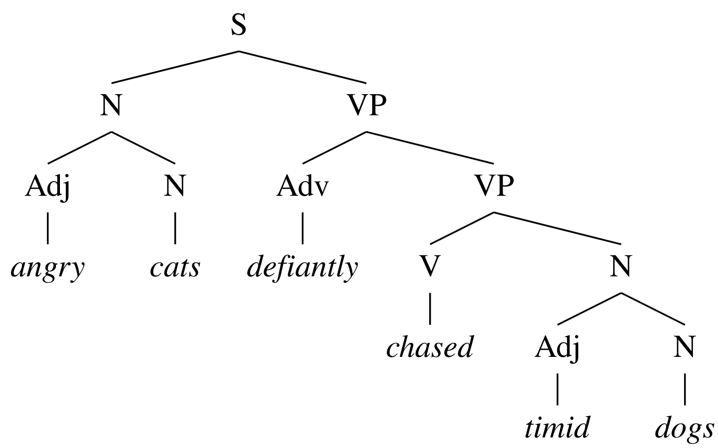
(20) Inspect the following set of PSRs and provide the tree diagram that it generates for the sentence *Fluffy chased Fido through town*.

Grammar (20)	
S →	N VP
VP →	VP PP
VP →	V N
PP →	P N
P →	*through*
V →	*chased*
N →	*Fluffy*
N →	*Fido*
N →	*town*

(21) Inspect the following, different, set of PSRs, and provide the tree diagram that it generates for the same sentence: *Fluffy chased Fido through town*.

Grammar (21)	
S →	N VP
VP →	V N P N
P →	*through*
V →	*chased*
N →	*Fluffy*
N →	*Fido*
N →	*town*

Provide a set of PSRs that will generate (22):

(22) 

(23) Look at the following set of PSRs and draw the tree diagram that it generates for the sentence *The young girl joyfully played her saxophone in the garden.*

Grammar (23)	
S →	NP VP
NP →	Det N
N →	Adj N
VP →	Adv VP PP
VP →	V NP
PP →	P NP
Det →	*her*
Det →	*the*
P →	*in*
V →	*played*
Adj →	*young*
Adv →	*joyfully*
N →	*girl*
N →	*garden*
N →	*saxophone*

Things to know:

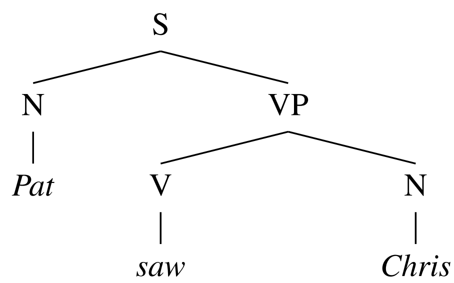
- How to identify the category of a content word using distributional evidence
- The distinction between content words and function words
- How to read and write Phrase Structure Rules
- The relationship between PSRs and tree diagrams
- How to work from PSRs to create trees and vice versa.

7 Bonus: Labeled brackets

Sometimes it is useful to represent tree structures without drawing lines. Like if you're typing an email.

A node in a tree defines a kind of sub-unit of the tree, which we will call a *constituent*.

Bracket notation represents a constituent not by the part of a tree that is under a single node, but instead by putting square brackets around it.



(24) [Pat [saw Chris]]

That's still less information than the tree has. The nodes have labels. And since paired brackets represent the nodes, we can label the pair (and provide brackets for the lexical nodes).

(25) $[_{S}$ Pat $[_{VP}$ saw Chris]]

(26) $[_{S}$ $[_{N}$ Pat] $[_{VP}$ $[_{V}$ saw] $[_{N}$ Chris]]

Depending on your method of email, this may still not be particularly helpful if you can't do subscripts. What I usually do is either just put the label up tight against the left bracket…

(27) `[S Pat [VP saw Chris ] ]`

(28) `[S [N Pat ] [VP [V saw ] [N Chris ] ]`

…or put an underscore connecting the bracket and the label.

(29) `[_S Pat [_VP saw Chris ] ]`

(30) `[_S [_N Pat ] [_VP [_V saw ] [_N Chris ] ]`