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?

This really is a problem for the idea that you (in this instance, an infant) just *generalize* from what you hear to the grammar of the language. It's more complicated than we even knew, we weren't taught this stuff. Something about either the *acquisition procedure* or the *cognitive system* or the *way it's stored in the brain* or something like that is constraining what kinds of languages there are.

This line of argument (language knowledge is more complex than is reflected in what language acquirers get in their environment) is called the *poverty of the stimulus* argument for the concept of Universal Grammar—which is itself the idea that there is something "human" about language. That by virtue of being human (and not a cat, ape, daffodil) we have this ability and mechanism to learn language. What determines that an organism will be human is that its parents are human; same deal with birds. So there's *something* genetic involved, if language is a uniquely human trait (even if it leverages other biological/cognitive systems that are not uniquely human).

2 Esperanto

There was an interesting observation made by Martin Haspelmath, quoted below, about the structure of the indefinite pronouns (like *somewhere*) and *wh*-words (like *where*) across the many languages he surveyed. The indefinite pronouns always seem to be derived from the *wh*-words, when they are related. The complex one is the indefinite pronoun. Yet that seems not to be true of Esperanto, a human-constructed language.

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.

Prescriptive rules do not represent the actual rules of grammar, and in fact their prescription is generally counter to what the grammar dictates. Why would you need to tell people that they should not end a sentence with a preposition unless they would otherwise be inclined to do it? This suggests that stranding preposition *is* valid in the grammar of English—that if it weren't for the prescriptive rule, people would go about their days happily stranding prepositions.

Thus: prescriptive rules in general are not rules of grammar, but just a code that people who know the code can use to identify other people who know the code.

4 Doing science

There are several levels of thing we're trying to do here, but the primary one for now is that we want to characterize what English speakers know about English syntax.

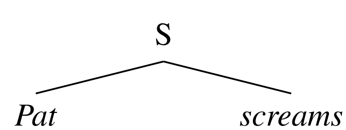
An English speaker, confronted with a string of words, can determine (through some procedure/knowledge) whether that string of words is an English sentence or not. English speakers agree on the divisions.

We've got a formalism that we're going to work with, it is a **generative grammar**—so called because it generates sentences. It is an algorithm that expands an abstract S (sentence) into a string of words. This is tied to reality by the hypothesis that **English speakers will accept as English all and only the sentences generated by the grammar**. And so now we want to work out the grammar.

Starting small.

Everybody agrees that *Pat screams* is a sentence of English.

(4) a. S $\rightarrow$ *Pat screams*



This grammar has several problems that keep us from accepting it as the thing English speakers know that makes them English speakers.

The first is that there are essentially infinitely more sentences that English speakers would accept as English other than *Pat screams*.

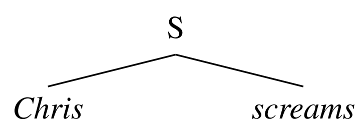
This grammar is a theory of English. It is a fairly poor theory of English. It predicts that the only sentence of English is *Pat screams*—{ *Pat screams* } is an exhaustive set of the sentences English speakers accept as English.

To demonstrate that this is a bad theory of English, we look at what it predicts (no sentence other than *Pat screams* is English), and check to see if the world bears this out. It does not. One reason, among others, is that *Chris screams* is also English, but the grammar does not generate it.

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

So, given that *Chris screams* is also English, we can revise our theory of English like so:

(5) a. S $\rightarrow$ *Pat screams*
    b. S $\rightarrow$ *Chris screams*



We have now patched the theory to account for the failure we identified in what it predicts. It now correctly predicts both that *Pat screams* and that *Chris screams* are English.

However, it incorrectly predicts that no other sentences are English.

But, turns out, *Tracy screams* is also English.

There are patterns here that are already obvious. We can always come up with a new name, and whatever name we put in there as the first word is going to wind up resulting in an English sentence. (For example, suppose that there's a science fiction story in which a group of people wake up in a spaceship having lost their memory and they adopt names based on the order they wake up. One woke up first, followed by Two. *One screams* and *Two screams* are both English. Extend this as far as you like. Whatever grammar you provide for a scenario where anyone up to *N* screams, I can invent a new scenario where there's one additional person, who wakes up last and has a correspondingly higher number, *M*, and *[M] screams* will be English too.)

Similarly, *Pat sleeps* is also English, and any sentence that ends with *screams* has a correspondng English sentence that ends with *sleeps*.

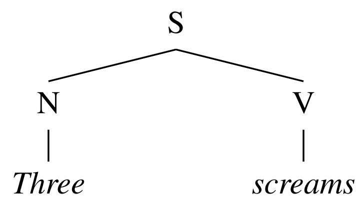
But *Pat Tracy* is not a sentence. Nor is *Screams sleeps*. So we can't just put anything anywhere, the name has to be in the first spot and the verb has to be in the second spot.

We learned something—this has told us something non-trivial. There are classes of things that go in particular places in an English sentence. In this type of sentence at least, anything that is a "name" can go in the first spot, but not the second, and anything that is in the class of words that so far includes *screams* and *sleeps* (let's call them "verbs") can go in the second spot.

To codify this in the grammar, we do this:

(6) a. S $\rightarrow$ N V

b. N $\rightarrow$ *Chris*
c. N $\rightarrow$ *Pat*
d. N $\rightarrow$ *Tracy*
e. N $\rightarrow$ *One*
f. N $\rightarrow$ *Two*
g. N $\rightarrow$ *Three*
...
h. N $\rightarrow$ *Two-Hundred-Forty-Eight*
i. V $\rightarrow$ *screams*
j. V $\rightarrow$ *sleeps*



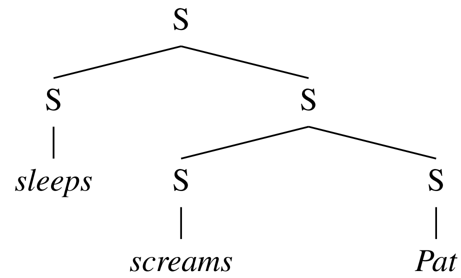
The two types of rules (the lexical and structural rules) really seem to have different conceptual status. The lexical rules are mostly just examples, we can easily extend them (particularly the ones corresponding to the open-class categories like nouns and verb) without really changing our understanding of the grammar. Extending the grammar to handle *One hated Three* is different in kind from just coming up with a new name that can go in an N spot. The real *grammar* is those structural rules, the *lexicon* is distinct (though we can use the same formalism to describe it).

The procedure we'll be essentially following is:

- 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

Given that, let me provide an alternative and terrible grammar.

(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*



Suppose that we've plugged in all of the English words into rules of the S → *word* lexical rules.

Now: try to find a sentence that this grammar does not generate. You can't.

Why is this grammar terrible? Why is this not the best grammar? It predicts that every English sentence there is is a grammatical sentence of English.

The tree above for *Sleeps screams Pat* shows one problem with it. It does successfully generate all the sentences of English. But it also generates all kinds of sentences that are not English. It is not a good model of what English speakers know about English syntax. It can't differentiate between English sentences and non-English sentences (though, perhaps it can differentiate between strings of words that are English words vs. those that are not English words).

So, success for a grammar is measured both by the degree to which it generates sentences that English speakers say are English, but also by the degree to which it fails to generate sentences that English speakers say are not English.

5 Parts of speech

In our S → N V grammar, we divided words of English into two classes, those that can substitute in for the first word, and those that can substitute in for the second word. The idea is that anything that is an "N" can go in the first spot (but not the second), and anything that is a "V" can go in the second spot (but not the first).

It is fairly evident that the distributional patterns of words differ from one another, but are generally shared between a set of words. So, there are a set of words that can grammatically fit in the "V" slot but not the "N" slot, and another set of words that has the reverse distribution. This is the kind of thing a pattern recognizer could pick up on pretty quickly. And there are more classes than "N" and "V"—there are things like *from*, which doesn't go in either the N or V spots, but seems to come with a N-type thing, and which shares a distribution with a number of other words (*under*, *toward*, etc.).

"Parts of speech" are the labels that we apply to distribution classes. Words are divided into classes, and when we look at the classes thus formed, we can name them based on what seems to characterize the words in the class.

That is to say, a "noun" is not by definition "a person, place, or thing" or whatever—a "noun" is a thing that can substitute in for other nouns, and it seems like people, places, and things are contained in that class.

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:

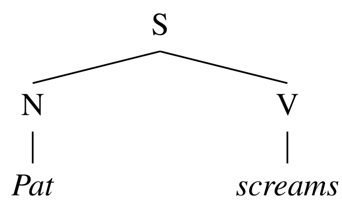
(9) $\mathrm{X} \rightarrow \mathrm{Y}_1\ \mathrm{Y}_2\ \mathrm{Y}_3 \ldots\ \mathrm{Y}_n$

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 → *Pat*.
- **Structural rules**. Define pattern of categories that form a larger expression. S → N V.

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

(10) a. S → N V
b. N → *Pat*
c. V → *screams*

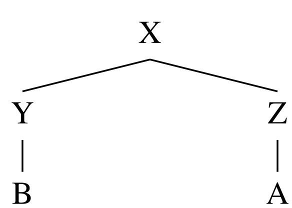


6.1 Hierarchical structures

In addition to sentences like "Pat screams," we also have sentences like "Pat kicks Chris." We have a sense from our past that "screams" and "kicks" are the same sort of thing, at some level. They both inflect for tense, for one thing. They're both verbs. They aren't exactly interchangeable though. You can't put "kicks" in place of "screams" (very easily at least), so "Pat kicks" seems weird, incomplete. But, looking at it another way, you can put "kicks Chris" in place of "screams." There is a sense in which we can consider "kicks Chris" (a complex group of words) to be a "V" as far as distributional classes are concerned. It is for this reason that we will call this grouping ("kicks Chris") a "VP" (verb phrase)—it goes where verbs go, and it is a phrase (potentially containing multiple words). Things will get more complicated/formalized, but this provides a basis for the idea that words can be grouped and the grouping itself can have a distributional class.

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]



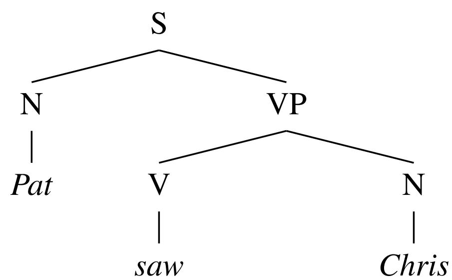
[1]Often you will find some of these relations referred to in an arbitrary feminine form (mother, daughter, sister), but that's just a historical tradition and seems like an extra thing to try to retain for no real reason. If you see "sister node" in a paper or book, that's just a name people often used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

- 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.

(11)



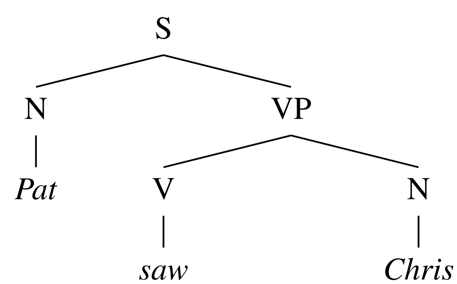
(12)
a. What is the root node?
b. What are the leaf nodes?
c. Which node is the parent of *Pat*?
d. Which node is the child of *Pat*?
e. Which node is the parent of the rightmost N?
f. Which node is the child of rightmost N?
g. Which node is the siblings of the leftmost N?
h. Which node is the siblings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

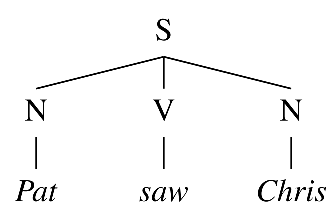


Do the rules in (15) generate the tree in (16)?

(15)

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

(16)



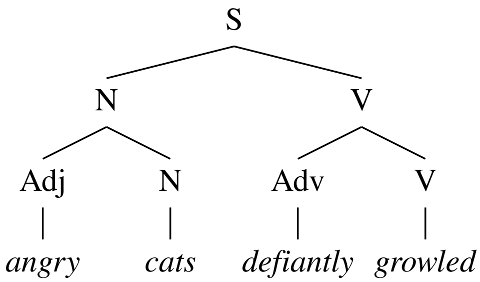
Sometimes a set of PSRs assigns more than one tree to the same string of words.
(This is a good thing, because this allows for a theory of certain kinds of ambiguity.)
How many tree diagrams does (17) generate for the sentence *Pat pats cats at Alexanderplatz*?

(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*

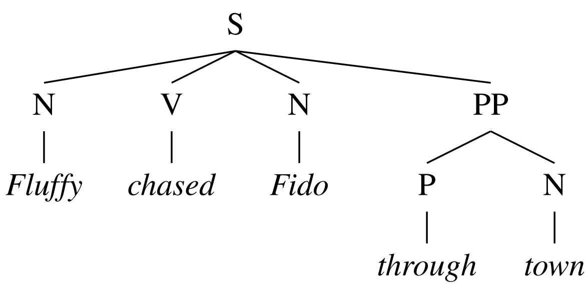
Let's walk through an example of how to go the other way, to take a tree and come up with the PSRs that can generate it. What rules generate (18)?

(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*.

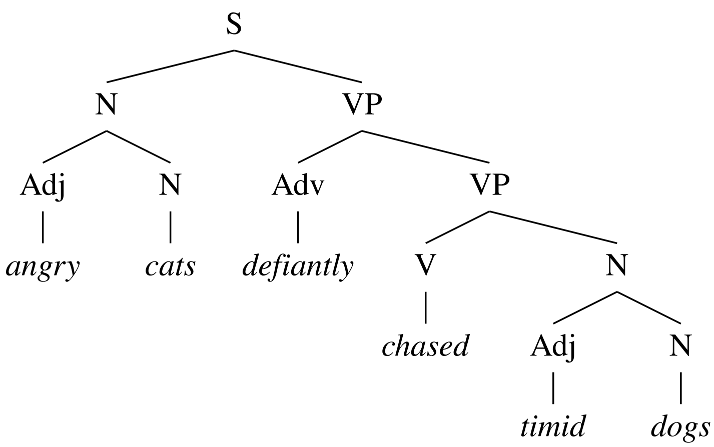
Grammar (20)	
S →	N VP
VP →	VP PP
VP →	V N
PP →	P N
P →	*through*
V →	*chased*
N →	*Fluffy*
N →	*Fido*
N →	*town*

(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

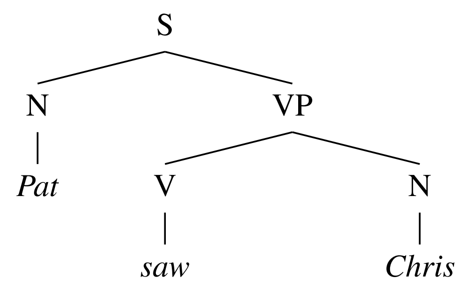
- How to read and write Phrase Structure Rules
- The relationship between PSRs and tree diagrams
- How to work from PSRs to create trees and vice versa.

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 ] ]`