Limits of Language

Gallagher Flinn

University of Chicago

May 2015
The basics

What do we know about the structure of language?

1. Language is made up of some kind of stuff.
2. Stuff gets combined and interpreted in a predictable way.
3. Certain ways of combining stuff are unlimited. Other stuff can never be combined.

To talk about language, we need a model that has some of the same properties that language does.
What are we even doing here?

We need a way to talk about syntactic properties in a principled way. For that, we’ve got *formal grammars*. Since the 50’s we’ve been treating languages as sets of *strings*, sequences of symbols that are the outputs of *rewrite rules* that tell us how they were formed.

A *formal grammar* describes the set of strings that are possible in a given language, and can recognize whether a given string belongs to the language.
Is this the right way to think about language?

It’s helpful! It fits the intuition we have that even though we can say an infinite number of sentences, there are some things we just can’t say. Ungrammatical sentences can’t be produced by our rules, so they can’t be interpreted by our grammar.
A note on ungrammaticality

A tame squirrel makes a nice pet.
A note on ungrammaticality

A tame squirrel makes a nice pet.

Grammatical.
A note on ungrammaticality

Pet a tame squirrel nice makes a.
A note on ungrammaticality

Pet a tame squirrel nice makes a.

Ungrammatical.
A note on ungrammaticality

Me a hungy boy!
A note on ungrammaticality

Me a hungy boy!

Adorable. But ungrammatical and uninterpretable aren’t necessarily the same thing.
Terminals: Stuff

Terminals:

- A formal grammar needs a set of *terminal* symbols. These symbols have a constant form and can’t be changed with any rule.
Terminals: Stuff

Terminals:

- A formal grammar needs a set of *terminal* symbols. These symbols have a constant form and can’t be changed with any rule.
- For the purpose of syntax, we can think of these as words. Or we can think of them as morphemes. It doesn’t really matter.
Terminals: Stuff

Terminals:

- A formal grammar needs a set of *terminal* symbols. These symbols have a constant form and can’t be changed with any rule.
- For the purpose of syntax, we can think of these as words. Or we can think of them as morphemes. It doesn’t really matter.
- *armadillo, bakery, if, carnivorous, speleological, mustn’t, pastafarian.*
Terminals: Stuff

Terminals:

- A formal grammar needs a set of *terminal* symbols. These symbols have a constant form and can’t be changed with any rule.
- For the purpose of syntax, we can think of these as words. Or we can think of them as morphemes. It doesn’t really matter.
- *armadillo, bakery, if, carnivorous, speleological, mustn’t, pastafarian.*
- We’ll also use an empty terminal $\epsilon$ that signifies an empty string.
Terminals: Stuff

Terminals:

- A formal grammar needs a set of *terminal* symbols. These symbols have a constant form and can’t be changed with any rule.
- For the purpose of syntax, we can think of these as words. Or we can think of them as morphemes. It doesn’t really matter.
- *armadillo*, *bakery*, *if*, *carnivorous*, *speleological*, *mustn’t*, *pastafarian*.
- We’ll also use an empty terminal $\epsilon$ that signifies an empty string.
- Usually written as lowercase letters.
Non-terminals: The structure of stuff

- A set of *non-terminal* symbols: These are intermediate forms, variables that can be rewritten and transformed into terminal symbols using rules. Think of these as phrasal categories like DP and VP.
- Usually written as uppercase letters. I’ll mostly use X and Y.
Rules: All possibilities

- Defines syntactically acceptable structures.
Rules: All possibilities

- Defines syntactically acceptable structures.
- Composed of rewrite rules $\alpha \rightarrow \beta$ where $\alpha$ contains at least one non-terminal and $\beta$ is a string of terminal and/or non-terminal symbols.

We've already seen something a little like this in phonology. The UR is the non-terminal and the SR is a string of terminals.

/l/ → [r] /lV/ is equivalent to: /lVl/ → [lVr]/
Rules: All possibilities

- Defines syntactically acceptable structures.
- Composed of rewrite rules $\alpha \rightarrow \beta$ where $\alpha$ contains at least one non-terminal and $\beta$ is a string of terminal and/or non-terminal symbols.
- Read $\alpha \rightarrow \beta$ as "$\alpha$ can be rewritten as $\beta$"
Rules: All possibilities

- Defines syntactically acceptable structures.
- Composed of rewrite rules $\alpha \rightarrow \beta$ where $\alpha$ contains at least one non-terminal and $\beta$ is a string of terminal and/or non-terminal symbols.
- Read $\alpha \rightarrow \beta$ as "$\alpha$ can be rewritten as $\beta$"
- We’ve already seen something a little like this in phonology. The UR is the non-terminal and the SR is a string of terminals.
  
  $/l/ \rightarrow [r] / lV$
Rules: All possibilities

- Defines syntactically acceptable structures.
- Composed of rewrite rules $\alpha \rightarrow \beta$ where $\alpha$ contains at least one non-terminal and $\beta$ is a string of terminal and/or non-terminal symbols.
- Read $\alpha \rightarrow \beta$ as "$\alpha$ can be rewritten as $\beta$"
- We’ve already seen something a little like this in phonology. The UR is the non-terminal and the SR is a string of terminals.
  /l/ $\rightarrow$ [r] / lV... is equivalent to:
  /lVl/ $\rightarrow$ [lVr]
Let’s write a language

We’re going to also need a non-terminal to use for a start symbol. We’ll call it K for Kool Start Symbol.

- Non-terminals: \{olives, salad, pizza\}
Let’s write a language

We’re going to also need a non-terminal to use for a start symbol. We’ll call it K for Kool Start Symbol.

- Non-terminals: \{olives, salad, pizza\}
- Terminals: \{O, S, P, K\}
Let’s write a language

We’re going to also need a non-terminal to use for a start symbol. We’ll call it K for Kool Start Symbol.

- Non-terminals: \{olives, salad, pizza\}
- Terminals: \{O, S, P, K\}
- Rules:
  \[ K \rightarrow PS \]
  \[ P \rightarrow \text{pizza} \]
  \[ S \rightarrow \text{salad} \]
  \[ P \rightarrow \text{pizza}O \]
  \[ O \rightarrow \text{olive} \]
  \[ O \rightarrow \text{SO} \]
Are these sentences grammatical?

1. pizza salad olive salad
2. pizza olive salad
3. pizza pizza pizza pizza pizza
4. pizza pizza pizza salad
These strings can be parsed with trees: *pizza olive salad*
Yaaaay! Now we have a model for language!
Yaaaay! Now we have a model for language!

No we don’t.
Yaaaay! Now we have a model for language!

No we don’t.  
But we do have a way to prove what’s inside and outside a language’s *generative capacity* depending on the kinds of rules we choose.
Types of languages and the Chomsky Hierarchy

(Greek letters here mean "terminal or non-terminal")

<table>
<thead>
<tr>
<th>Language Type</th>
<th>Rule Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recursively enumerable</td>
<td>$\alpha \rightarrow \beta$</td>
</tr>
<tr>
<td>Context-sensitive</td>
<td>$\alpha X \beta \rightarrow \alpha \gamma \beta$</td>
</tr>
<tr>
<td>Context-free</td>
<td>$X \rightarrow \alpha$</td>
</tr>
<tr>
<td>Regular</td>
<td>$X \rightarrow \text{pizza}$</td>
</tr>
<tr>
<td></td>
<td>$X \rightarrow \text{pizzaY}$</td>
</tr>
</tbody>
</table>
"In order to understand recursion, you must understand recursion."
– some Internet person
Back to Recursion

“In order to understand recursion, you must understand recursion.” – some Internet person

*recursion*: a function that can be defined within its own definition
“In order to understand recursion, you must understand recursion.”
– some Internet person

*recursion*: a function that can be defined within its own definition
Was our earlier grammar recursive?
Let’s write a language

- Non-terminals: \{olives, salad, pizza\}
Let’s write a language

- Non-terminals: \{olives, salad, pizza\}
- Terminals: \{O, S, P, K\}

Is recursion enough?
Let’s write a language

- Non-terminals: \{olives, salad, pizza\}
- Terminals: \{O, S, P, K\}
- Rules:
  - K → PS
  - P → pizza
  - S → salad
  - P → pizzaO
  - O → olive
  - O → SO

Is recursion enough?
Regular languages: This porridge is too cold.
Regular languages: This porridge is too cold.

Probably not.
Regular languages: This porridge is too cold.

Probably not.

How would we parse something like "The merchant the plague rat bit died"?
Recursively enumerable languages: This porridge is too hot.

\[ \alpha \rightarrow \beta \text{ is way too powerful.} \]
Recurisely enumerable languages: This porridge is too hot.

\( \alpha \rightarrow \beta \) is way too powerful.
You can do whatever you want! It's anarchy!
Recursively enumerable languages: This porridge is too hot.

\[ \alpha \rightarrow \beta \] is way too powerful.
You can do whatever you want! It’s anarchy!
Mild Context Sensitivity: This porridge is okay I guess.

There’s also reason to believe that context free grammars aren’t expressive enough to capture human language. But going another step up in the Chomsky hierarchy and parsing becomes too computationally difficult.
Mild Context Sensitivity: This porridge is okay I guess.

There’s also reason to believe that context free grammars aren’t expressive enough to capture human language. But going another step up in the Chomsky hierarchy and parsing becomes too computationally difficult.

Swiss-German:
...de Karl d’Maria em Peter de Hans laat hälfte lärne schwüme

English:
...Charles lets Mary help Peter to teach John to Swim

Mildly context sensitive formalisms can capture the pattern \(a^m b^n c^m d^n\) and is parseable in a reasonable amount of computing time (polynomial).
Other restrictions: Our brains... is... not so good.

Center embeddings
Other restrictions: Our brains... is... not so good.

Center embeddings

- The boy cried.

The fact that the rumor that ghosts inhabited the mansion attracted teenagers surprised no one.

There's something going on here beyond the capacity for recursion. Why aren't these parseable after a certain point?
Other restrictions: Our brains... is... not so good.

Center embeddings

- The boy cried.
- The boy who the girl loved cried.
Other restrictions: Our brains... is... not so good.

Center embeddings

- The boy cried.
- The boy who the girl loved cried.
- The boy who the girl who the cat bit loved cried.
Other restrictions: Our brains... is... not so good.

Center embeddings

- The boy cried.
- The boy who the girl loved cried.
- The boy who the girl who the cat bit loved cried.
- The boy who the girl who the cat that the mouse feared bit loved cried.

The fact that the rumor that ghosts inhabited the mansion attracted teenagers surprised no one.

There's something going on here beyond the capacity for recursion.

Why aren't these parseable after a certain point?
Other restrictions: Our brains... is... not so good.

Center embeddings

- The boy cried.
- The boy who the girl loved cried.
- The boy who the girl who the cat bit loved cried.
- The boy who the girl who the cat that the mouse feared bit loved cried.
- The boy who the girl who the cat that the mouse that the accountant saw feared loved cried.
Other restrictions: Our brains... is... not so good.

Center embeddings

- The boy cried.
- The boy who the girl loved cried.
- The boy who the girl who the cat bit loved cried.
- The boy who the girl who the cat that the mouse feared bit loved cried.
- The boy who the girl who the cat that the mouse that the accountant saw feared loved cried.

The fact that the rumor that ghosts inhabited the mansion attracted teenagers surprised no one.
Other restrictions: Our brains... is... not so good.

Center embeddings

- The boy cried.
- The boy who the girl loved cried.
- The boy who the girl who the cat bit loved cried.
- The boy who the girl who the cat that the mouse feared bit loved cried.
- The boy who the girl who the cat that the mouse that the accountant saw feared loved cried.

The fact that the rumor that ghosts inhabited the mansion attracted teenagers surprised no one.

There’s something going on here beyond the capacity for recursion. Why aren’t these parseable after a certain point?
Is this grammatical?

This isn’t grammatical:
Is this grammatical?

This isn’t grammatical:
Most people have must have been to Paris.
Is this grammatical?

This isn’t grammatical:
Most people have must have been to Paris.
But this is:
Is this grammatical?

This isn’t grammatical:
Most people have must have been to Paris.
But this is:
I’m sure that many more people have been to Paris than I have.
Is this grammatical?

This isn’t grammatical:
Most people have must have been to Paris.
But this is:
I’m sure that many more people have been to Paris than I have.
Our parsers are so robust that you can get grammatical illusions!