What is a grammar?

A grammar is a set of structural rules that govern the composition of sentences, phrases and words.

Computational view:
A grammar (often called a “formal grammar”) is a set of rules that describe what strings are valid in a formal language.

Lots of different kinds of grammars:
- regular
- context-free
- context-sensitive
- recursively enumerable
- transformation grammars
Context Free Grammars (CFG)

How many people have heard of them?

What do you know about them?

Where are they used?

CFG production rules

\[ S \rightarrow NP \ VP \]

<table>
<thead>
<tr>
<th>left hand side</th>
<th>right hand side</th>
</tr>
</thead>
<tbody>
<tr>
<td>(single symbol)</td>
<td>(one or more symbols)</td>
</tr>
</tbody>
</table>

CFG example

Grammars “generate” or “derive” strings:

- \( S \rightarrow A \ B \ C \)
- \( A \rightarrow I \)
- \( B \rightarrow \text{really} \)
- \( B \rightarrow \text{really, } B \)
- \( C \rightarrow \text{like } cs \)

 CFG example

Grammars “generate” or “derive” strings:

- \( S \rightarrow A \ B \ C \)
- \( A \rightarrow I \)
- \( B \rightarrow \text{really} \)
- \( B \rightarrow \text{really, } B \)
- \( C \rightarrow \text{like } cs \)

We can apply a rule by substituting the symbol on the left hand side with the symbols on the right.
Grammars “generate” or “derive” strings:

**CFG example**

\[ S \rightarrow A \ B \ C \]
\[ A \rightarrow I \]
\[ B \rightarrow \text{really} \]
\[ B \rightarrow \text{really, B} \]
\[ C \rightarrow \text{like cs} \]

We can apply a rule by substituting the symbol on the left hand side with the symbols on the right.
Grammars "generate" or "derive" strings:

- S → A B C
- A → I
- B → really
- B → really, B
- C → like cs

We can apply a rule by substituting the symbol on the left hand side with the symbols on the right.

No more rules apply, so we're done!
Grammars "generate" or "derive" strings:

\[ S \rightarrow A \ B \ C \]
\[ A \rightarrow I \]
\[ B \rightarrow \text{really} \]
\[ B \rightarrow \text{really, B} \]
\[ C \rightarrow \text{like cs} \]

We can apply a rule by substituting the symbol on the left hand side with the symbols on the right.

Is this the only string that can be derived?

Grammars describe a language, i.e. the strings (aka sentences) that are part of that language.

\[ S \rightarrow A \ B \ C \]
\[ A \rightarrow I \]
\[ B \rightarrow \text{really} \]
\[ B \rightarrow \text{really, B} \]
\[ C \rightarrow \text{like cs} \]

We can apply a rule by substituting the symbol on the left hand side with the symbols on the right.

\[ A \rightarrow \text{really, B} \ C \]
\[ B \rightarrow \text{really, B} \]
\[ C \rightarrow \text{like cs} \]

I really, really, ... like cs
### CFGs formally

\[ G = (NT, T, P, S) \]

- **NT**: finite set of nonterminal symbols
- **T**: finite set of terminal symbols, NT and T are disjoint
- **P**: finite set of productions of the form \( A \rightarrow \alpha \), \( A \in NT \) and \( \alpha \in (T \cup NT)^* \)
- **S \in NT**: start symbol

### What language does this represent?

- \( S \rightarrow aS \)
- \( S \rightarrow E \)
- \( E \rightarrow bE \)
- \( E \rightarrow b \)

### What language does this represent?

<table>
<thead>
<tr>
<th>Production 1</th>
<th>Production 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S \rightarrow aS )</td>
<td>( S \rightarrow aS )</td>
</tr>
<tr>
<td>( S \rightarrow E )</td>
<td>( S \rightarrow E )</td>
</tr>
<tr>
<td>( E \rightarrow bE )</td>
<td>( E \rightarrow bE )</td>
</tr>
<tr>
<td>( E \rightarrow b )</td>
<td>( E \rightarrow b )</td>
</tr>
</tbody>
</table>

Two options
What language does this represent?

S → aS
S → E
E → bE
E → b

What language does this represent?

S → aS
S → E
E → bE
E → b

- Can do this as many times as we want
- Keeps adding more a’s to the front

Eventually, apply second rule

What language does this represent?

S → aS
S → E
E → bE
E → b

What language does this represent?

S → aS
S → E
E → bE
E → b

Two options
What language does this represent?

- S → aS
- S → E
- E → bE
- E → b

What language does this represent?

- S → aS
- S → E
- E → bE
- E → b

What language does this represent?

- S → aS
- S → E
- E → bE
- E → b

- Can do this as many times as we want
- Keeps adding more b’s to the end
What language does this represent?

<table>
<thead>
<tr>
<th>Grammar</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S \rightarrow aS$</td>
<td>$aaabb...bE$</td>
</tr>
<tr>
<td>$S \rightarrow E$</td>
<td>$aaabb...bb$</td>
</tr>
<tr>
<td>$E \rightarrow bE$</td>
<td>$aaabb...bb$</td>
</tr>
<tr>
<td>$E \rightarrow b$</td>
<td>$aaabb...bb$</td>
</tr>
<tr>
<td>Eventually, apply second rule</td>
<td></td>
</tr>
</tbody>
</table>

Grammar represents all strings with zero or more a's followed by one or more b's