Grammars

Language view:
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.

CFG production rules

\[ S \rightarrow NP \ VP \]

left hand side \hspace{20pt} right hand side

[single symbol] \hspace{20pt} [one or more symbols]
CFG example

S → A B C
A → I
B → really
B → really, B
C → 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 → α, A ∈ NT and α ∈ (T ∪ NT)*
S ∈ NT: start symbol

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.
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.

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.
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.

A really like cs

I really like cs

No more rules apply, so we’re done!
Grammars “generate” or “derive” strings:

\[
\begin{align*}
S & \rightarrow A \ B \ C \\
A & \rightarrow I \\
B & \rightarrow \text{really} \\
\underline{B} & \rightarrow \text{really, } B \\
C & \rightarrow \text{like } \text{cs}
\end{align*}
\]

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

Example:

\[
\begin{align*}
A & \rightarrow I \\
B & \rightarrow \text{really, } B \\
C & \rightarrow \text{like } \text{cs}
\end{align*}
\]

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

Example:

\[
\begin{align*}
A & \rightarrow I \\
\underline{B} & \rightarrow \text{really, } B \\
C & \rightarrow \text{like } \text{cs}
\end{align*}
\]

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

Example:

\[
\begin{align*}
S & \rightarrow A \ B \ C \\
A & \rightarrow I \\
B & \rightarrow \text{really, } B \\
C & \rightarrow \text{like } \text{cs}
\end{align*}
\]

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

Example:

\[
\begin{align*}
S & \rightarrow A \ B \ C \\
A & \rightarrow I \\
\underline{B} & \rightarrow \text{really, } B \\
C & \rightarrow \text{like } \text{cs}
\end{align*}
\]

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

Example:

\[
\begin{align*}
S & \rightarrow A \ B \ C \\
A & \rightarrow I \\
B & \rightarrow \text{really, } B \\
C & \rightarrow \text{like } \text{cs}
\end{align*}
\]

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

Example:

\[
\begin{align*}
S & \rightarrow A \ B \ C \\
A & \rightarrow I \\
B & \rightarrow \text{really, } B \\
C & \rightarrow \text{like } \text{cs}
\end{align*}
\]

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

Example:

\[
\begin{align*}
S & \rightarrow A \ B \ C \\
A & \rightarrow I \\
B & \rightarrow \text{really, } B \\
C & \rightarrow \text{like } \text{cs}
\end{align*}
\]

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

Example:

\[
\begin{align*}
S & \rightarrow A \ B \ C \\
A & \rightarrow I \\
B & \rightarrow \text{really, } B \\
C & \rightarrow \text{like } \text{cs}
\end{align*}
\]

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

Example:

\[
\begin{align*}
S & \rightarrow A \ B \ C \\
A & \rightarrow I \\
\underline{B} & \rightarrow \text{really, } B \\
C & \rightarrow \text{like } \text{cs}
\end{align*}
\]

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

Example:

\[
\begin{align*}
S & \rightarrow A \ B \ C \\
A & \rightarrow I \\
\underline{B} & \rightarrow \text{really, } B \\
C & \rightarrow \text{like } \text{cs}
\end{align*}
\]

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

Example:

\[
\begin{align*}
S & \rightarrow A \ B \ C \\
A & \rightarrow I \\
\underline{B} & \rightarrow \text{really, } B \\
C & \rightarrow \text{like } \text{cs}
\end{align*}
\]

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

Example:

\[
\begin{align*}
S & \rightarrow A \ B \ C \\
A & \rightarrow I \\
\underline{B} & \rightarrow \text{really, } B \\
C & \rightarrow \text{like } \text{cs}
\end{align*}
\]

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

Example:

\[
\begin{align*}
S & \rightarrow A \ B \ C \\
A & \rightarrow I \\
\underline{B} & \rightarrow \text{really, } B \\
C & \rightarrow \text{like } \text{cs}
\end{align*}
\]

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

Example:

\[
\begin{align*}
S & \rightarrow A \ B \ C \\
A & \rightarrow I \\
\underline{B} & \rightarrow \text{really, } B \\
C & \rightarrow \text{like } \text{cs}
\end{align*}
\]

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

Example:

\[
\begin{align*}
S & \rightarrow A \ B \ C \\
A & \rightarrow I \\
\underline{B} & \rightarrow \text{really, } B \\
C & \rightarrow \text{like } \text{cs}
\end{align*}
\]

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

Example:

\[
\begin{align*}
S & \rightarrow A \ B \ C \\
A & \rightarrow I \\
\underline{B} & \rightarrow \text{really, } B \\
C & \rightarrow \text{like } \text{cs}
\end{align*}
\]

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

Example:

\[
\begin{align*}
S & \rightarrow A \ B \ C \\
A & \rightarrow I \\
\underline{B} & \rightarrow \text{really, } B \\
C & \rightarrow \text{like } \text{cs}
\end{align*}
\]

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

Example:

\[
\begin{align*}
S & \rightarrow A \ B \ C \\
A & \rightarrow I \\
\underline{B} & \rightarrow \text{really, } B \\
C & \rightarrow \text{like } \text{cs}
\end{align*}
\]

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

Example:

\[
\begin{align*}
S & \rightarrow A \ B \ C \\
A & \rightarrow I \\
\underline{B} & \rightarrow \text{really, } B \\
C & \rightarrow \text{like } \text{cs}
\end{align*}
\]

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

Example:

\[
\begin{align*}
S & \rightarrow A \ B \ C \\
A & \rightarrow I \\
\underline{B} & \rightarrow \text{really, } B \\
C & \rightarrow \text{like } \text{cs}
\end{align*}
\]

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

Example:

\[
\begin{align*}
S & \rightarrow A \ B \ C \\
A & \rightarrow I \\
\underline{B} & \rightarrow \text{really, } B \\
C & \rightarrow \text{like } \text{cs}
\end{align*}
\]

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

Example:

\[
\begin{align*}
S & \rightarrow A \ B \ C \\
A & \rightarrow I \\
\underline{B} & \rightarrow \text{really, } B \\
C & \rightarrow \text{like } \text{cs}
\end{align*}
\]

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

What language does this represent?

S → aS
S → E
E → bE
E → b

S

aS

S

aS

aS
What language does this represent?

\[
\begin{align*}
S & \rightarrow aS \\
S & \rightarrow E \\
E & \rightarrow bE \\
E & \rightarrow b
\end{align*}
\]
\[
\begin{align*}
\text{aaS} & \quad \downarrow \\
\text{aaaS}
\end{align*}
\]
- 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?

\[
\begin{align*}
S & \rightarrow aS \\
S & \rightarrow E \\
E & \rightarrow bE \\
E & \rightarrow b
\end{align*}
\]
\[
\begin{align*}
\text{aaaS} & \quad \downarrow \\
\text{aaaE}
\end{align*}
\]

Two options

What language does this represent?

\[
\begin{align*}
S & \rightarrow aS \\
S & \rightarrow E \\
E & \rightarrow bE \\
E & \rightarrow b
\end{align*}
\]
\[
\begin{align*}
\text{aaE} & \quad \downarrow \\
\text{aaabE}
\end{align*}
\]
### What language does this represent?

<table>
<thead>
<tr>
<th>Production</th>
<th>Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S \rightarrow aS$</td>
<td>$aaabE$</td>
</tr>
<tr>
<td>$S \rightarrow E$</td>
<td>$aaabE$</td>
</tr>
<tr>
<td>$E \rightarrow bE$</td>
<td></td>
</tr>
<tr>
<td>$E \rightarrow b$</td>
<td></td>
</tr>
</tbody>
</table>

- 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>Production</th>
<th>Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S \rightarrow aS$</td>
<td>$aaabbE$</td>
</tr>
<tr>
<td>$S \rightarrow E$</td>
<td>$aaabbE$</td>
</tr>
<tr>
<td>$E \rightarrow bE$</td>
<td></td>
</tr>
<tr>
<td>$E \rightarrow b$</td>
<td></td>
</tr>
</tbody>
</table>

Eventually, apply second rule
### What language does this represent?

<table>
<thead>
<tr>
<th>Grammar</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S \rightarrow aS )</td>
<td>( a ) followed by any number of ( b )s</td>
</tr>
<tr>
<td>( S \rightarrow aS )</td>
<td>( a ) followed by any number of ( b )s</td>
</tr>
<tr>
<td>( E \rightarrow bE )</td>
<td>( a ) followed by any number of ( b )s</td>
</tr>
</tbody>
</table>

Grammar represents all strings with zero or more \( a \)'s followed by one or more \( b \)'s.

### Notational convenience

<table>
<thead>
<tr>
<th>Grammar</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S \rightarrow aS )</td>
<td>( a ) followed by any number of ( b )s</td>
</tr>
<tr>
<td>( S \rightarrow E )</td>
<td>( a ) followed by any number of ( b )s</td>
</tr>
<tr>
<td>( E \rightarrow bE )</td>
<td>( a ) followed by any number of ( b )s</td>
</tr>
</tbody>
</table>

Often many ways to write the same language:

<table>
<thead>
<tr>
<th>Grammar</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S \rightarrow aS )</td>
<td>( a ) followed by any number of ( b )s</td>
</tr>
<tr>
<td>( E \rightarrow bE )</td>
<td>( a ) followed by any number of ( b )s</td>
</tr>
</tbody>
</table>

### What languages do these represent?

<table>
<thead>
<tr>
<th>Grammar</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S \rightarrow aEa )</td>
<td>( a ) followed by any number of ( b )s</td>
</tr>
<tr>
<td>( E \rightarrow Ea )</td>
<td>( a ) followed by any number of ( b )s</td>
</tr>
<tr>
<td>( E \rightarrow aB )</td>
<td>( a ) followed by any number of ( b )s</td>
</tr>
<tr>
<td>( E \rightarrow bE )</td>
<td>( a ) followed by any number of ( b )s</td>
</tr>
<tr>
<td>( S \rightarrow aSb )</td>
<td>( a ) followed by any number of ( b )s</td>
</tr>
<tr>
<td>( S \rightarrow ab )</td>
<td>( a ) followed by any number of ( b )s</td>
</tr>
</tbody>
</table>

Nothing
What languages do these represent?

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S \rightarrow aEa \mid bEb$</td>
<td>strings of $a$'s followed by an equal number of $b$'s</td>
</tr>
<tr>
<td>$E \rightarrow Ea \mid Eb \mid a \mid b$</td>
<td>strings of $a$'s that start and end with the same letter</td>
</tr>
<tr>
<td>$S \rightarrow aSb \mid ab$</td>
<td>all strings of $a$'s and $b$'s with even length</td>
</tr>
<tr>
<td>$S \rightarrow aaS \mid abS \mid baS \mid bbS \mid \varepsilon$</td>
<td>all strings of $a$'s and $b$'s that start and end with the same letter</td>
</tr>
</tbody>
</table>

Writing CFGs

Write a CFG to represent the language containing all strings that start with $a$.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S \rightarrow aT$</td>
<td>all strings that start with $a$</td>
</tr>
<tr>
<td>$T \rightarrow Ta \mid Tb \mid \varepsilon$</td>
<td>strings of $a$'s and $b$'s with even length</td>
</tr>
</tbody>
</table>

Write a CFG to represent the language containing all strings with exactly two $b$s.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S \rightarrow TbTbT$</td>
<td>all strings with exactly two $b$s</td>
</tr>
<tr>
<td>$T \rightarrow Ta \mid Tb \mid \varepsilon$</td>
<td>strings of $a$'s and $b$'s with even length</td>
</tr>
</tbody>
</table>

CFG: Another example

Many possible CFGs for English, here is an example (fragment):

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S \rightarrow NP \ VP$</td>
<td>all strings that start with $a$</td>
</tr>
<tr>
<td>$VP \rightarrow V \ NP$</td>
<td>strings of $a$'s and $b$'s with even length</td>
</tr>
<tr>
<td>$NP \rightarrow DetP \ N \mid DetP \ AdjP \ N$</td>
<td>all strings of $a$'s and $b$'s that start and end with the same letter</td>
</tr>
<tr>
<td>$AdjP \rightarrow Adj \mid Adv \ AdjP$</td>
<td>strings of $a$'s followed by an equal number of $b$'s</td>
</tr>
<tr>
<td>$N \rightarrow \text{boy} \mid \text{girl}$</td>
<td>strings of $a$'s and $b$'s with even length</td>
</tr>
<tr>
<td>$V \rightarrow \text{sees} \mid \text{likes}$</td>
<td>strings of $a$'s and $b$'s with even length</td>
</tr>
<tr>
<td>$Adj \rightarrow \text{big} \mid \text{small}$</td>
<td>strings of $a$'s and $b$'s with even length</td>
</tr>
<tr>
<td>$Adv \rightarrow \text{very}$</td>
<td>strings of $a$'s and $b$'s with even length</td>
</tr>
<tr>
<td>$DetP \rightarrow \text{a} \mid \text{the}$</td>
<td>strings of $a$'s and $b$'s with even length</td>
</tr>
</tbody>
</table>
Derivations in a CFG

S → NP VP
VP → V NP
NP → DetP N | DetP AdjP N
AdjP → Adj | Adv AdjP
N → boy | girl
V → sees | likes
Adj → big | small
Adv → very
DetP → a | the

What can we do?
Derivations in a CFG

$S \rightarrow NP \ VP$

$NP \rightarrow DetP \ N \mid DetP \ AdjP \ N$

$VP \rightarrow V \ NP$

$DetP \rightarrow a \mid the$

$NP \rightarrow DetP \ N \mid DetP \ AdjP \ N$

$AdjP \rightarrow Adj \mid Adv \ AdjP$

$N \rightarrow boy \mid girl$

$V \rightarrow sees \mid likes$

$Adv \rightarrow very$

Derivations in a CFG

$S \rightarrow NP \ VP$

$NP \rightarrow DetP \ N \mid DetP \ AdjP \ N$

$VP \rightarrow V \ NP$

$DetP \rightarrow a \mid the$

$NP \rightarrow DetP \ N \mid DetP \ AdjP \ N$

$AdjP \rightarrow Adj \mid Adv \ AdjP$

$N \rightarrow boy \mid girl$

$V \rightarrow sees \mid likes$

$Adv \rightarrow very$

Derivations in a CFG

$S \rightarrow NP \ VP$

$VP \rightarrow V \ NP$

$NP \rightarrow DetP \ N \mid DetP \ AdjP \ N$

$AdjP \rightarrow Adj \mid Adv \ AdjP$

$N \rightarrow boy \mid girl$

$V \rightarrow sees \mid likes$

$Adv \rightarrow very$

$DetP \rightarrow a \mid the$

Derivations in a CFG

$S \rightarrow NP \ VP$

$VP \rightarrow V \ NP$

$NP \rightarrow DetP \ N \mid DetP \ AdjP \ N$

$AdjP \rightarrow Adj \mid Adv \ AdjP$

$N \rightarrow boy \mid girl$

$V \rightarrow sees \mid likes$

$Adv \rightarrow very$

$DetP \rightarrow a \mid the$
Derivations in a CFG

**S → NP VP**
**VP → V NP**
**NP → DetP N | DetP AdjP N**
**AdjP → Adj | Adv AdjP**
**N → boy | girl**
**V → sees | likes**
**Adv → very**
**DetP → a | the**

- the boy likes a girl
- the boy likes
- a girl

Derivations in a CFG:
Order of Derivation Irrelevant

Another CFG example

**S → NP VP**
**VP → V | V ADV**
**NP → ART PreNP**
**PreNP → N | ADJ PreNP**
**ADV → furiously | soothingly | intentionally**
**ADJ → colorless | green | smelly**
**ART → the | a**
**V → sleeps | eats | swims | sprints**
**N → idea | bagel | milk | cow**

What can we generate?
One last example

S → N
S → ( S )
S → S + S | S - S
S → S * S | S / S
N → 0 | 1 | 2 | ... | 9
N → N N

What language does this CFG represent?

S → N
S → ( S )
S → S + S | S - S
S → S * S | S / S
N → 0 | 1 | 2 | ... | 9
N → N N

All arithmetic expressions!

One last example

Parsing

Given a CFG and a sentence, determine the possible parse tree(s)

I eat sushi with tuna

What parse trees are possible for this sentence?

How did you do it?

What if the grammar is much larger?

Parsing

I eat sushi with tuna

I eat sushi with tuna

What is the difference between these parses?
CFGs implemented