Context Free Grammars

In a regular grammar, production rules have the form: $A \to bC$ or $A \to \Lambda$.
In a context free grammar, we relax that restriction.
Rules are of the form: $A \to \alpha$, where $A \in V$ and $\alpha \in (V \cup \Sigma)^*$

Context Free Grammars

In a regular grammar, production rules have the form: $A \to bC$ or $A \to \Lambda$.
In a context free grammar, we relax that restriction.
Rules are of the form: $A \to \alpha$, where $A \in V$ and $\alpha \in (V \cup \Sigma)^*$

Example:
$G = (\{S\}, \{a, b\}, S, \{S \to aSb \mid \Lambda\})$

Context Free Grammars

In a regular grammar, production rules have the form: $A \to bC$ or $A \to \Lambda$.
In a context free grammar, we relax that restriction.
Rules are of the form: $A \to \alpha$, where $A \in V$ and $\alpha \in (V \cup \Sigma)^*$

Example:
$G = (\{S\}, \{a, b\}, S, \{S \to aSb \mid \Lambda\})$
$L(G) = \{a^n b^n \mid n \geq 0\}$

Context Free Grammars

In a regular grammar, production rules have the form: $A \to bC$ or $A \to \Lambda$.
In a context free grammar, we relax that restriction.
Rules are of the form: $A \to \alpha$, where $A \in V$ and $\alpha \in (V \cup \Sigma)^*$

Example:
$G = (\{S\}, \{a, b\}, S, \{S \to aSb \mid \Lambda\})$
$L(G) = \{a^n b^n \mid n \geq 0\}$

Example:
$G = (\{S\}, \{`(', `)'\}, S, \{S \to (S) \mid SS \mid \Lambda\})$

Context Free Grammars

In a regular grammar, production rules have the form: $A \rightarrow bC$ or $A \rightarrow \Lambda$.
In a context free grammar, we relax that restriction.
Rules are of the form: $A \rightarrow \alpha$, where $A \in V$ and $\alpha \in (V \cup \Sigma)^*$

Example:
$G = (\{S\}, \{a, b\}, S, \{S \rightarrow aSb \mid \Lambda\})$
$L(G) = \{a^n b^n \mid n \geq 0\}$

Example:
$G = (\{S\}, \{`(', `)'\}, S, \{S \rightarrow (S) \mid SS \mid \Lambda\})$
Try to derive the strings: $(()())$ $(()(()))$

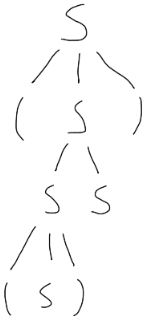
Derivation Trees

S

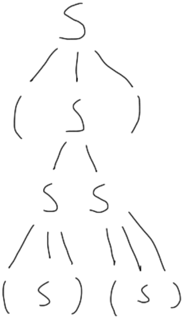
Derivation Trees

Derivation Trees

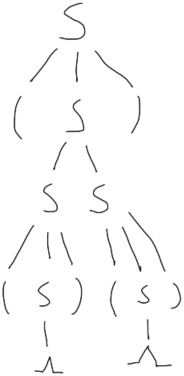
Derivation Trees



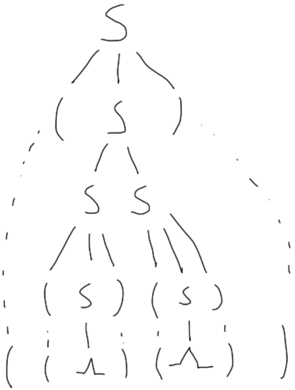
Derivation Trees



Derivation Trees



Derivation Trees

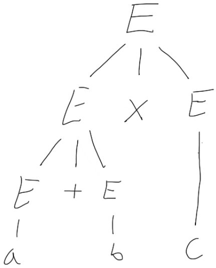


Ambiguity

Consider the following CFG:
$E \to E + E \mid E \times E \mid a \mid b \mid c$

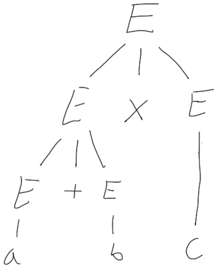
Ambiguity

Consider the following CFG:
$E \to E + E \mid E \times E \mid a \mid b \mid c$



Ambiguity

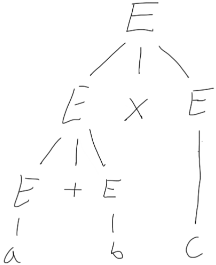
Consider the following CFG:
$E \to E + E \mid E \times E \mid a \mid b \mid c$



Two derivations that are consistent with this structure:
$E \Rightarrow E \times E \Rightarrow E + E \times E \Rightarrow a + E \times E \Rightarrow a + b \times E \Rightarrow a + b \times c$

Ambiguity

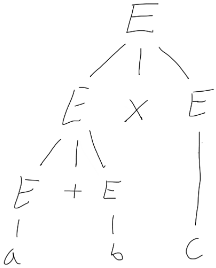
Consider the following CFG:
$E \rightarrow E + E \mid E \times E \mid a \mid b \mid c$



Two derivations that are consistent with this structure:
$E \Rightarrow E \times E \Rightarrow E + E \times E \Rightarrow a + E \times E \Rightarrow a + b \times E \Rightarrow a + b \times c$
$E \Rightarrow E \times E \Rightarrow E \times c \Rightarrow E + E \times c \Rightarrow E + b \times c \Rightarrow a + b \times c$

Ambiguity

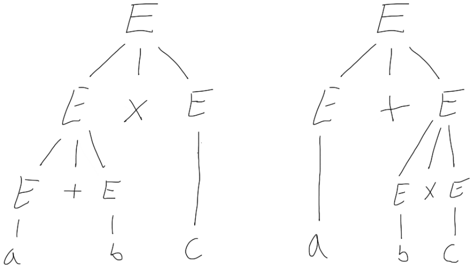
Consider the following CFG:
$E \to E + E \mid E \times E \mid a \mid b \mid c$



Two derivations that are consistent with this structure:
$E \Rightarrow E \times E \Rightarrow E + E \times E \Rightarrow a + E \times E \Rightarrow a + b \times E \Rightarrow a + b \times c$
$E \Rightarrow E \times E \Rightarrow E \times c \Rightarrow E + E \times c \Rightarrow E + b \times c \Rightarrow a + b \times c$
The first is called a "leftmost derivation": we always replace the leftmost variable from V first.

Ambiguity

Consider the following CFG:
$E \to E + E \mid E \times E \mid a \mid b \mid c$

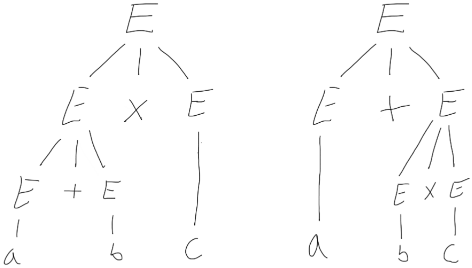


Leftmost derivation consistent with the left derivation tree:
$E \Rightarrow E \times E \Rightarrow E + E \times E \Rightarrow a + E \times E \Rightarrow a + b \times E \Rightarrow a + b \times c$

Leftmost derivation consistent with the right derivation tree:

Ambiguity

Consider the following CFG:
$E \to E + E \mid E \times E \mid a \mid b \mid c$

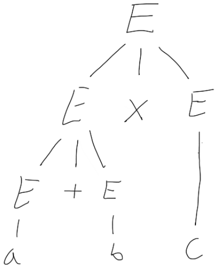


Leftmost derivation consistent with the left derivation tree:
$E \Rightarrow E \times E \Rightarrow E + E \times E \Rightarrow a + E \times E \Rightarrow a + b \times E \Rightarrow a + b \times c$

Leftmost derivation consistent with the right derivation tree:
$E \Rightarrow E + E \Rightarrow a + E \Rightarrow a + E \times E \Rightarrow a + b \times E \Rightarrow a + b \times c$

Ambiguity

Consider the following CFG:
$E \to E + E \mid E \times E \mid a \mid b \mid c$



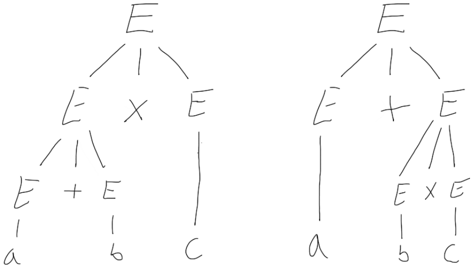
Leftmost derivation consistent with the left derivation tree:
$E \Rightarrow E \times E \Rightarrow E + E \times E \Rightarrow a + E \times E \Rightarrow a + b \times E \Rightarrow a + b \times c$

Leftmost derivation consistent with the right derivation tree:
$E \Rightarrow E + E \Rightarrow a + E \Rightarrow a + E \times E \Rightarrow a + b \times E \Rightarrow a + b \times c$

“The girl touches the boy with the flower”

Ambiguity

Consider the following CFG:
$E \to E + E \mid E \times E \mid a \mid b \mid c$



Leftmost derivation consistent with the left derivation tree:
$E \Rightarrow E \times E \Rightarrow E + E \times E \Rightarrow a + E \times E \Rightarrow a + b \times E \Rightarrow a + b \times c$

Leftmost derivation consistent with the right derivation tree:
$E \Rightarrow E + E \Rightarrow a + E \Rightarrow a + E \times E \Rightarrow a + b \times E \Rightarrow a + b \times c$

"The girl touches the boy with the flower"

Ambiguous Grammars

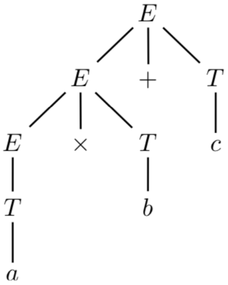
A string is *ambiguous* with respect to some CFG if the grammar can generate the string with at least 2 different derivation trees. An ambiguous grammar is a grammar that generates at least 1 ambiguous string.

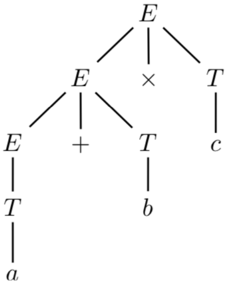
Disambiguating the Grammar

$$E \to E + T \mid E \times T \mid T$$
$$T \to a \mid b \mid c$$

Disambiguating the Grammar

$E \to E + T \mid E \times T \mid T$
$T \to a \mid b \mid c$





Chomsky Normal Form

We can restrict the grammar without changing the language class (by much):
$A \to BC$ or $A \to a$.

Chomsky Normal Form

We can restrict the grammar without changing the language class (by much):
$A \to BC$ or $A \to a$.
Note that no $A \to \Lambda$ is allowed!
Any language that does not contain Λ is called a Λ-free language.

Chomsky Normal Form

We can restrict the grammar without changing the language class (by much):
$A \to BC$ or $A \to a$.
Note that no $A \to \Lambda$ is allowed!
Any language that does not contain Λ is called a Λ-free language.

Theorem

For any context free language L that is Λ-free, there exists a CFG in Chomsky Normal Form such that $L = \mathcal{L}(G)$.

Limits on CFGs

$L = \{a^n b^n c^n \mid n > 0\}$ is not context free.

Limits on CFGs

$L = \{a^n b^n c^n \mid n > 0\}$ is not context free.

Claim 1

For any CFL L of infinite size, for any G in Chomsky normal form that generates L, there is no bound on the height of the derivation trees required of G by strings in L.

Limits on CFGs

$L = \{a^n b^n c^n \mid n > 0\}$ is not context free.

Claim 1

For any CFL L of infinite size, for any G in Chomsky normal form that generates L, there is no bound on the height of the derivation trees required of G by strings in L.

Proof: The trees are binary trees, and a tree of height h can only generate strings of length at most 2^{h-1}.

Limits on CFGs

$L = \{a^n b^n c^n \mid n > 0\}$ is not context free.

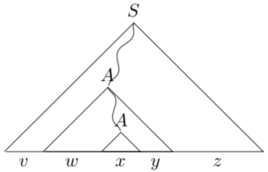
Claim 1

For any CFL L of infinite size, for any G in Chomsky normal form that generates L, there is no bound on the height of the derivation trees required of G by strings in L.

Proof: The trees are binary trees, and a tree of height h can only generate strings of length at most 2^{h-1}.

Claim 2

For any CFL L of infinite size, for any G in Chomsky normal form that generates L, G has some derivation tree with a path from the root to the leaf containing two occurrences of the same nonterminal symbol.



Limits on CFGs

$L = \{a^n b^n c^n \mid n > 0\}$ is not context free.

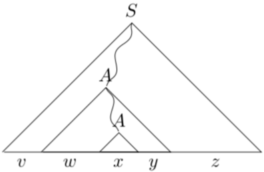
Claim 1

For any CFL L of infinite size, for any G in Chomsky normal form that generates L, there is no bound on the height of the derivation trees required of G by strings in L.

Proof: The trees are binary trees, and a tree of height h can only generate strings of length at most 2^{h-1}.

Claim 2

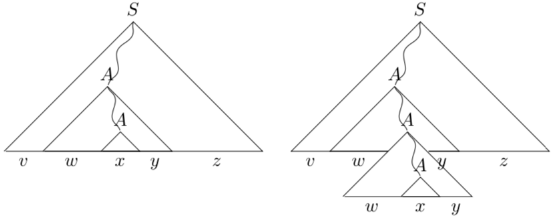
For any CFL L of infinite size, for any G in Chomsky normal form that generates L, G has some derivation tree with a path from the root to the leaf containing two occurrences of the same nonterminal symbol.



Proof: By Claim 1, we know that there are some derivation trees of height greater than $n = |V|$. Since there are no terminal symbols at internal nodes of the tree, by the pigeon-hole principal, there must be a repetition of one of the variables in V.

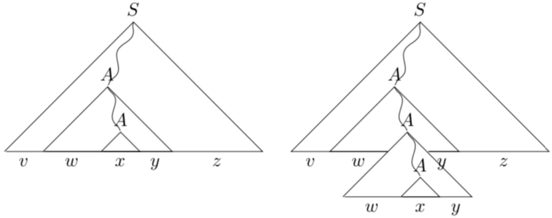
Limits on CFGs

$L = \{a^n b^n c^n \mid n > 0\}$ is not context free.



Limits on CFGs

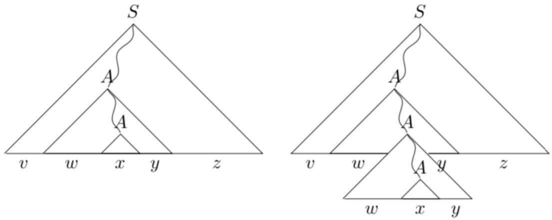
$L = \{a^n b^n c^n \mid n > 0\}$ is not context free.



Let x be the string generated by the lower A.
Let wxy be the string generated by the upper A.

Limits on CFGs

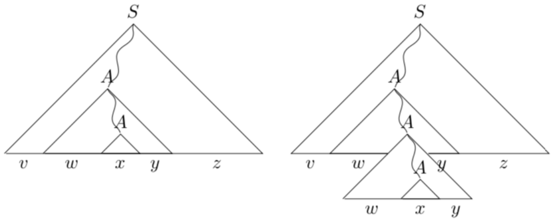
$L = \{a^n b^n c^n \mid n > 0\}$ is not context free.



Let x be the string generated by the lower A.
Let wxy be the string generated by the upper A.
Note that either w or y is non-empty.

Limits on CFGs

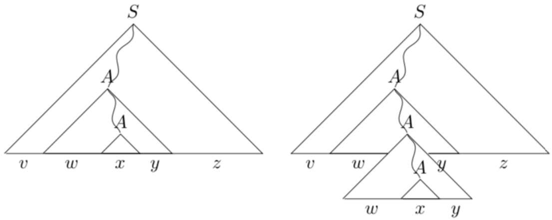
$L = \{a^n b^n c^n \mid n > 0\}$ is not context free.



Let x be the string generated by the lower A.
Let wxy be the string generated by the upper A.
Note that either w or y is non-empty.
Let $u = vwxyz$ be the entire string generated by the root of the tree.
Since $u \in L$, $u = a^m b^m c^m$ for some integer $m > 0$.

Limits on CFGs

$L = \{a^n b^n c^n \mid n > 0\}$ is not context free.

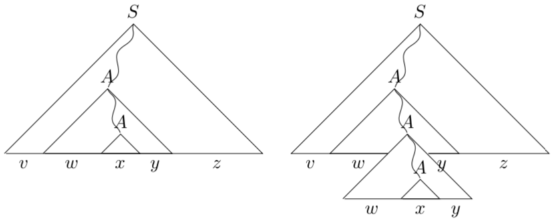


Let x be the string generated by the lower A.
Let wxy be the string generated by the upper A.
Note that either w or y is non-empty.
Let $u = vwxyz$ be the entire string generated by the root of the tree.
Since $u \in L$, $u = a^m b^m c^m$ for some integer $m > 0$.

Use the subtree rooted at the upper A to replace the subtree rooted at the lower A.
Let $u' = vwwxyyz$ be the resulting string.
$u' \in L$, so $u' = a^n b^n c^n$ for some $n > m$.

Limits on CFGs

$L = \{a^n b^n c^n \mid n > 0\}$ is not context free.



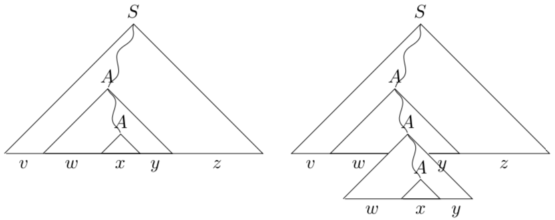
Let x be the string generated by the lower A.
Let wxy be the string generated by the upper A.
Note that either w or y is non-empty.
Let $u = vwxyz$ be the entire string generated by the root of the tree.
Since $u \in L$, $u = a^m b^m c^m$ for some integer $m > 0$.

Use the subtree rooted at the upper A to replace the subtree rooted at the lower A.
Let $u' = vwwxyyz$ be the resulting string.
$u' \in L$, so $u' = a^n b^n c^n$ for some $n > m$.

A string is homogenous if it contains only one symbol.
w is homogenous: suppose it contains ab. Then ww has a b before an a, and $u' \notin L$.

Limits on CFGs

$L = \{a^n b^n c^n \mid n > 0\}$ is not context free.



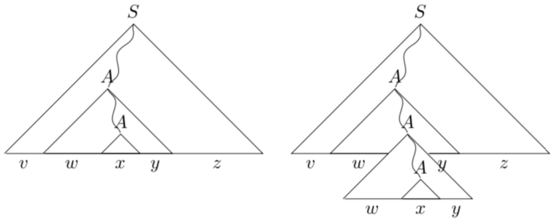
Let x be the string generated by the lower A.
Let wxy be the string generated by the upper A.
Note that either w or y is non-empty.
Let $u = vwxyz$ be the entire string generated by the root of the tree.
Since $u \in L$, $u = a^m b^m c^m$ for some integer $m > 0$.

Use the subtree rooted at the upper A to replace the subtree rooted at the lower A.
Let $u' = vwwxyyz$ be the resulting string.
$u' \in L$, so $u' = a^n b^n c^n$ for some $n > m$.

A string is homogenous if it contains only one symbol.
w is homogenous: suppose it contains ab. Then ww has a b before an a, and $u' \notin L$.
y is homogenous: suppose it contains ab. Then yy has a b before an a, and $u' \notin L$.

Limits on CFGs

$L = \{a^n b^n c^n \mid n > 0\}$ is not context free.



Let x be the string generated by the lower A.
Let wxy be the string generated by the upper A.
Note that either w or y is non-empty.
Let $u = vwxyz$ be the entire string generated by the root of the tree.
Since $u \in L$, $u = a^m b^m c^m$ for some integer $m > 0$.

Use the subtree rooted at the upper A to replace the subtree rooted at the lower A.
Let $u' = vwwxyyz$ be the resulting string.
$u' \in L$, so $u' = a^n b^n c^n$ for some $n > m$.

A string is homogenous if it contains only one symbol.
w is homogenous: suppose it contains ab. Then ww has a b before an a, and $u' \notin L$.
y is homogenous: suppose it contains ab. Then yy has a b before an a, and $u' \notin L$.
u' is bigger than u, but the only added characters come from w and y which are homogenous, so there exists some terminal character that appears only m times in u'.