# Context Free Grammar

# Introduction

- Why do we want to learn about Context Free Grammars?
  - Used in many parsers in compilers
    - Yet another compiler-compiler, **`yacc`**
    - GNU project parser generator, **`bison`**
  - Web application
    - In describing the formats of XML (eXtensible Markup Language) documents

# Context Free Grammar

- $G = (V, T, P, S)$
  - V – a set of variables, e.g. {S, A, B, C, D, E}
  - T – a set of terminals, e.g. {a, b, c}
  - P – a set of productions rules
    - In the form of A  α, where $A \in V$, $\alpha \in (V \cup T)^*$ e.g. S  aB
  - S is a special variable called the **start symbol**

# CFG – Example 1

- Construct a CFG for the following language : $L_1=\{0^n1^n \mid n>0\}$
- Thought: If $\omega$ is a string in $L_1$, so is $0\omega1$, i.e. $0^{k+1}1^{k+1} = 0(0^k1^k)1$
- S □ 01 | 0S1

# CFG – Example 2

- Construct a CFG for the following language : $L_2=\{0^i1^j \mid i\neq j \text{ and } i, j>0\}$
- Thought: Similar to L1, but at least one more ‘1’ or at least one more ‘0’
- S □ AC | CB
- A □ 0A | 0
- B □ B1 | 1
- C □ 0C1 | ε

# CFG – Example 3

- Determine the language generated by the CFG:

  S □ AS | ε

  A □ A1 | 0A1 | 01
- S generates consecutive 'A's
- A generates $0^i1^j$ where $i \leq j$
- Languages: each block of '0's is followed by at least as many '1's

# Derivation

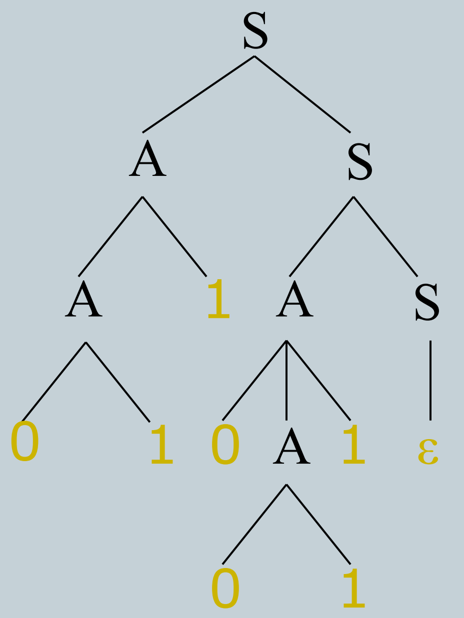
- How a string ω is generated by a grammar G
- Consider the grammar G

  S □ AS | ε

  A □ A1 | 0A1 | 01
- $S \Rightarrow 0110011$ ?

# Derivation

**Parse Tree or Derivation Tree**




**Leftmost Derivation**

$$
\begin{aligned}
S &\Rightarrow AS\\
&\Rightarrow A1S\\
&\Rightarrow 011S\\
&\Rightarrow 011AS\\
&\Rightarrow 0110A1S\\
&\Rightarrow 0110011S\\
&\Rightarrow 0110011\ \varepsilon
\end{aligned}
$$

**Rightmost Derivation**

$$
\begin{aligned}
S &\Rightarrow AS\\
&\Rightarrow AAS\\
&\Rightarrow AA\ \varepsilon\\
&\Rightarrow A0A1\ \varepsilon\\
&\Rightarrow A0011\ \varepsilon\\
&\Rightarrow A10011\ \varepsilon\\
&\Rightarrow 0110011\ \varepsilon
\end{aligned}
$$

# Ambiguity

- Each parse tree has one unique leftmost derivation and one unique rightmost derivation.

- A grammar is **ambiguous** if some strings in it have more than one parse trees, i.e., it has more than one leftmost derivations (or more than one rightmost derivations).

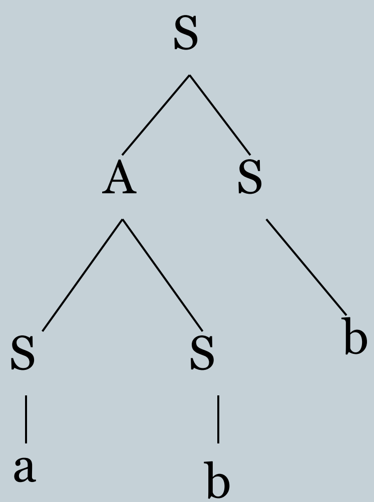
# Ambiguity

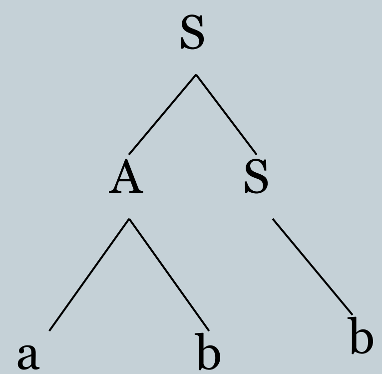
Consider the following grammar G:

$$S \rightarrow AS \mid a \mid b$$

$$A \rightarrow SS \mid ab$$

A string generated by this grammar can have more than one parse trees. Consider the string abb:



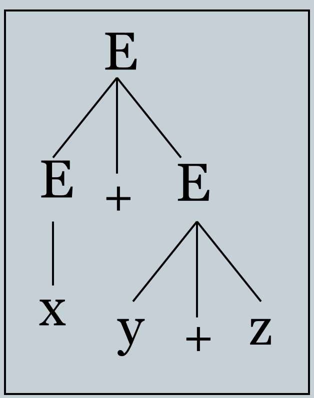



# Ambiguity

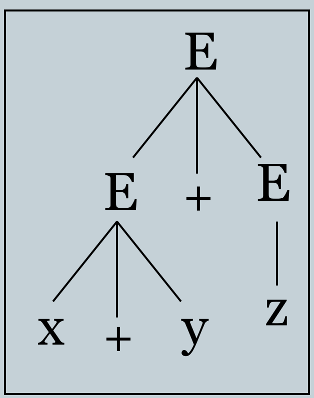
As another example, consider the following grammar:

$$E \rightarrow E + E \mid E * E \mid (E) \mid x \mid y \mid z$$

There are 2 leftmost derivations for x + y + z:




$$\begin{aligned} E &\Rightarrow E + E \\ &\Rightarrow x + E \\ &\Rightarrow x + E + E \\ &\Rightarrow x + y + E \\ &\Rightarrow x + y + z \end{aligned}$$




$$\begin{aligned} E &\Rightarrow E + E \\ &\Rightarrow E + E + E \\ &\Rightarrow x + E + E \\ &\Rightarrow x + y + E \\ &\Rightarrow x + y + z \end{aligned}$$

# Simplification of CFG

- Remove useless variables
  - Generating Variables
  - Reachable Variables
- Remove ε-productions, e.g. A□ ε
- Remove unit-productions, e.g. A□B

# Useless Variables

A variable X is useless if:

- X does not generate any string of terminals, or
- The start symbol, S, cannot generate X.

$$S \Rightarrow \alpha X \beta \Rightarrow \omega$$

where $\alpha, \beta \in (V \cup T)^*$, $X \in V$ and $\omega \in T^*$

# Removal of Non-generating Variables

1. Mark each production of the form:

   $X \to \omega$ where $\omega \in T^*$

- Repeat

  Mark $X \to \alpha$ where $\alpha$ consists of terminals or variables which are on the left hand side of some marked productions.

  Until no new productions is marked.

3. Remove all unmarked productions.

# Example

CFG:

$S \rightarrow AB|CA$

$B \rightarrow BC|AB$

$C \rightarrow aB|b$

$A \rightarrow a$

$\rightarrow$

$S \rightarrow CA$

$A \rightarrow a$

$C \rightarrow b$

# Example

CFG:

$S \to aAa|aB$

$A \to aS|bD$

$B \to aBa|b$

$C \to abb|DD$

$D \to aDa$

$\to$

$S \to aAa|aB$

$A \to aS$

$B \to aBa|b$

$C \to abb$

# Removal of Non-Reachable Variables

1 Mark each production of the form:

$$S \rightarrow \alpha \text{ where } \alpha \in (V \cup T)^*$$

- Repeat

  Mark $X \rightarrow \beta$ where X appears on the right hand side of some marked productions.

  Until no new productions is marked.

3 Remove all unmarked productions.

# Example

CFG:

$S \to aAa|aB$

$A \to aS$

$B \to aBa|b$

$C \to abb$

$$\to \begin{array}{l} S \to aAa|aB \\ A \to aS \\ B \to aBa|b \end{array}$$

# Example

## CFG:

$S \rightarrow Aab|AB$

$A \rightarrow a$

$B \rightarrow bD \mid bB$

$C \rightarrow A \mid B$

$D \rightarrow a$

$S \rightarrow Aab|AB$

$\rightarrow A \rightarrow a$

$B \rightarrow bD \mid bB$

$D \rightarrow a$

# Remove Useless Variables

$A \rightarrow Bac|bTC|ba$

$T \rightarrow aB|TB$

$B \rightarrow aTB|bBC$

$C \rightarrow TBc|aBC|ac$

$$\rightarrow \begin{array}{l} A \rightarrow ba \\ C \rightarrow ac \end{array}$$

$$\rightarrow A \rightarrow ba$$

# Removal of ε-productions

- Step 1: Find nullable variables
- Step 2: Remove all ε-productions by replacing some productions

# How to find nullable variables ?

1. Mark all variables A for which there exists a production of the form $A \to \varepsilon$.
2. Repeat

   Mark X for which there exists $X \to \beta$ where $\beta \in V^*$ and all symbols in $\beta$ have been marked.

   Until no new variables is marked.

# Removal of ε-Productions

We can remove all ε-productions (except $S \to \varepsilon$ if S is nullable) by rewriting some other productions.

e.g., If $X_1$ and $X_3$ are nullable, we should replace

$A \to X_1X_2X_3X_4$ by:

$A \to X_1X_2X_3X_4 \mid X_2X_3X_4 \mid X_1X_2X_4 \mid X_2X_4$

# Removal of ε-productions

Example:

$S \rightarrow A \mid B \mid C$

$A \rightarrow aAa \mid B$

$B \rightarrow bB \mid bb$

$C \rightarrow aCaa \mid D$

$D \rightarrow baD \mid abD \mid \varepsilon$

Step1: nullable variables are $D$, $C$ and $S$

# Removal of ε-productions(3)

Step 2:

- eliminate D □ ε by replacing:
  - *D □ baD | abD* into *D □ baD | abD | ba | ab*
- eliminate C □ ε by replacing:
  - *C □ aCaa | D* into *C □ aCaa | D | aaa*
  - $S \rightarrow A \mid B \mid C$ *into* $S \rightarrow A \mid B \mid C \mid \varepsilon$
- The new grammar:
  - $S \rightarrow A \mid B \mid C \mid \varepsilon$
  - $A \rightarrow aAa \mid B$
  - $B \rightarrow bB \mid bb$
  - $C \rightarrow aCaa \mid D \mid aaa$
  - $D \rightarrow baD \mid abD \mid ba \mid ab$

Example:
$S \rightarrow A \mid B \mid C$
$A \rightarrow aAa \mid B$
$B \rightarrow bB \mid bb$
$C \rightarrow aCaa \mid D$
$D \rightarrow baD \mid abD \mid \varepsilon$

# Example

$S \rightarrow ABCD$

$A \rightarrow a$

$B \rightarrow \varepsilon$

$C \rightarrow ED \mid \varepsilon$

$D \rightarrow BC$

$E \rightarrow b$

$S \rightarrow ABCD|ABC|ABD|ACD|AB|AC|AD|A$

$A \rightarrow a$

$C \rightarrow ED|E$

$D \rightarrow BC|B|C$

$E \rightarrow b$

# Removal of Unit Productions

Unit production: $A \rightarrow B$, where A and B $\in V$

1. Detect cycles of unit productions:

$A_1 \Rightarrow A_2 \Rightarrow A_3 \Rightarrow \ldots \Rightarrow A_1$

Replaced $A_1, A_2, A_3, \ldots, A_k$ by any one of them

2. Detect paths of unit productions:

$A_1 \Rightarrow A_2 \Rightarrow A_3 \Rightarrow \ldots, A_k \Rightarrow \alpha$, where $\alpha \in (V \cup T)^*/V$

Add productions $A_1 \rightarrow \alpha, A_2 \rightarrow \alpha, \ldots, A_k \rightarrow \alpha$ and remove all the unit productions

# Removal of Unit Productions

Example:

$S \rightarrow A \mid B \mid C$

$A \rightarrow aa \mid B$

$B \rightarrow bb \mid C$

$C \rightarrow cc \mid A$

Cycle: $A \rightarrow B \rightarrow C \rightarrow A$
Remove by replace B,C by A

$S \rightarrow A \mid A \mid A$
$A \rightarrow aa \mid A$
$A \rightarrow bb \mid A$
$A \rightarrow cc \mid A$

Becomes:

$S \rightarrow A$

$A \rightarrow aa \mid bb \mid cc$

Path: $S \rightarrow A$
Remove by adding productions

$S \rightarrow aa \mid bb \mid cc$

# Example

CFG:

$S \rightarrow A \mid Bb$

$A \rightarrow C \mid a$

$B \rightarrow aBa \mid b$

$C \rightarrow aSa$

$$\rightarrow \begin{array}{l} S \rightarrow Bb \mid aSa \mid a \\ A \rightarrow a \mid aSa \\ B \rightarrow aBa \mid b \\ C \rightarrow aSa \end{array}$$

# Example

$$S \to aA$$

$$A \to a$$

$$\cancel{A \to B}$$

$$B \to A$$

$$B \to bb$$

Substitute

$$A \to B$$

$$S \to aA \mid aB$$

$$A \to a$$

$$B \to A \mid B$$

$$B \to bb$$

# Example

Unit productions of form $X \rightarrow X$ can be removed immediately

$$S \rightarrow aA \mid aB$$

$$A \rightarrow a$$

$$B \rightarrow A \mid \cancel{B}$$

$$B \rightarrow bb$$

Remove

$$B \rightarrow B$$

$$S \rightarrow aA \mid aB$$

$$A \rightarrow a$$

$$B \rightarrow A$$

$$B \rightarrow bb$$

# Example

$$S \rightarrow aA \mid aB$$

$$A \rightarrow a$$

$$\cancel{B \rightarrow A}$$

$$B \rightarrow bb$$

Substitute

$$B \rightarrow A$$

$$S \rightarrow aA \mid aB \mid aA$$

$$A \rightarrow a$$

$$B \rightarrow bb$$

# Example

Remove repeated productions

Final grammar

$$S \to aA \mid aB \mid \cancel{aA}$$

$$A \to a$$

$$B \to bb$$

$$S \to aA \mid aB$$

$$A \to a$$

$$B \to bb$$

# Thank You