

# Konsep Tata Bahasa Bebas konteks, Pohon Penurunan, ambigiutas

Mahasiswa mampu menerapkan Tata Bahasa  
Bebas Konteks, Pohon Penurunan Dan Ambigiutas

# Pengertian Tata Bahasa Bebas Konteks

Sebuah grammar dikatakan context free **jika** semua sisi kirinya hanya terdiri dari satu simbol non-terminal (Variabel), dan sisi kanan setidaknya terdiri dari satu simbol terminal.

**Pada Aturan Produksi :**

$$\alpha \rightarrow \beta$$

**Contoh  $B \rightarrow CDeFg$**

**$D \rightarrow BcDe$**

Tata Bahasa Bebas Konteks ( CFG ) digunakan sebagai cara untuk menunjukkan bagaimana menghasilkan utai- utai dalam sebuah Bahasa.

Pada saat menurunkan utai, symbol – symbol variable akan mewakili bagian – bagian yang diturunkan dari utai tersebut.

Bahasa bebas konteks menjadi dasar dalam membentuk parser/ proses analisis sintaksis.

## PENERAPAN TATA BAHASA BEBAS KONTEKS DENGAN POHON PENURUNAN

### Pohon Penurunan (Derivation Tree)

Pohon penurunan (derivation tree / parse tree) berguna untuk menggambarkan bagaimana memperoleh suatu string (untai) dengan cara menurunkan simbol- simbol variabel menjadi simbol-simbol terminal. Setiap simbol variabel akan diturunkan menjadi terminal sampai tidak ada yang belum tergantikan.

### Contoh

Misalkan terdapat tata bahasa bebas konteks dengan aturan produksi :

$$S \rightarrow AB$$

$$A \rightarrow aA \mid a$$

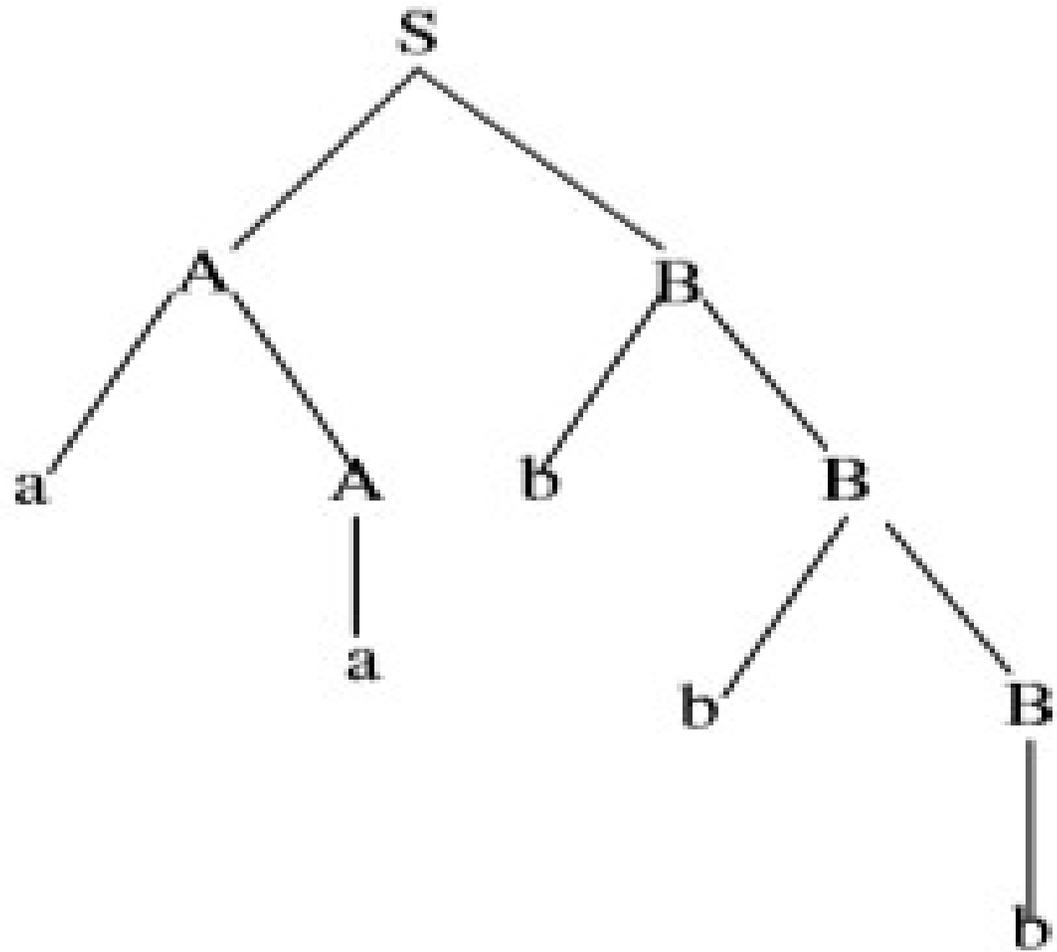
$$B \rightarrow bB \mid b$$

gambarkan pohon penurunan untuk memperoleh untai : "aabbb"

$S \rightarrow AB$

$A \rightarrow aA \mid a$

$B \rightarrow bB \mid b$



## Proses Penurunan ( *PARSING* )

Proses penurunan dapat dilakukan dengan cara :

### A. Penurunan Terkiri ( Leftmost Derivatioan)

Penurunan terkiri dilakukan dengan cara menurunkan symbol variable terkiri terlebih dahulu

### B. Penurunan Terkanan ( Rightmost Derivation)

Penurunan terkiri dilakukan dengan cara menurunkan symbol variable terkanan terlebih dahulu

Contoh :

$$S \rightarrow aAS \mid a$$

$$A \rightarrow SbA \mid ba$$

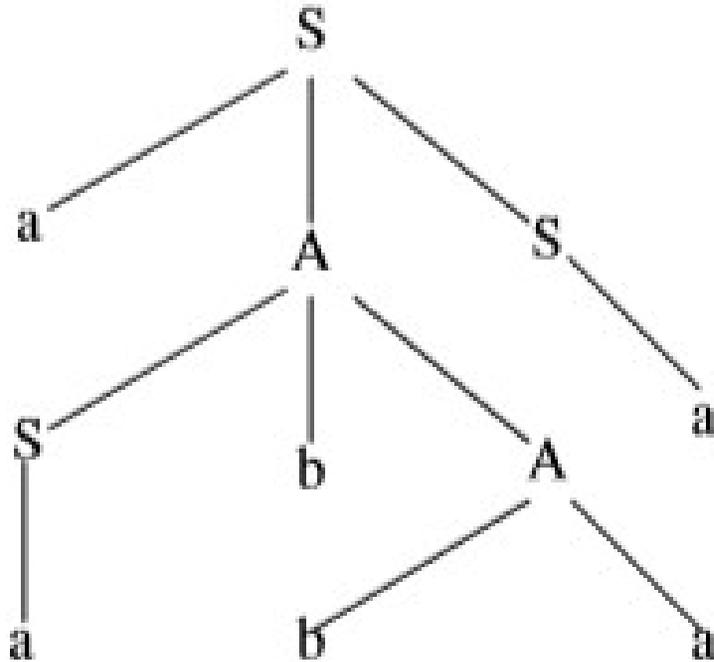
untuk memperoleh untai "aabbaa"

**Penurunan terkiri:  $S \Rightarrow aAS \Rightarrow aSbAS \Rightarrow aabAS \Rightarrow aabbaS \Rightarrow aabbaa$**

**Penurunan terkanan :  $S \Rightarrow aAS \Rightarrow aAa \Rightarrow aSbAa \Rightarrow aAbbaa \Rightarrow aabbaa$**

Penurunan ter kiri:  $S \Rightarrow aAS \Rightarrow aSbAS \Rightarrow aabAS \Rightarrow aabbaS \Rightarrow aabbaa$

Penurunan ter kanan :  $S \Rightarrow aAS \Rightarrow aAa \Rightarrow aSbAa \Rightarrow aAbbaa \Rightarrow aabbaa$



Pohon Penurunan untuk untaian “ aabbaa”

# AMBIGUITAS

- ❑ Ambiguitas disebabkan karena adanya dua atau lebih parse tree, bukan karena banyaknya penurunan
- ❑ CFG  $G = (V, T, P, S)$  dikatakan ambigu jika terdapat sedikitnya satu string  $w$  dalam  $T^*$  dimana kita dapat menentukan dua parse tree yang berbeda
- ❑ Masing-masing parse tree tersebut memiliki root yang diberi label  $S$  dan hasil  $w$
- ❑ Jika setiap string paling banyak satu parse tree dalam grammar, maka grammar tersebut dikatakan tidak ambigu

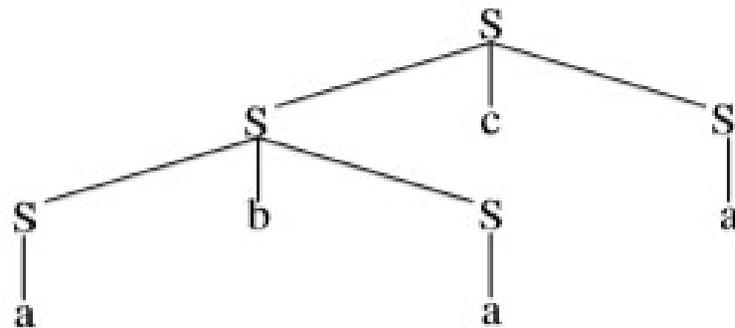
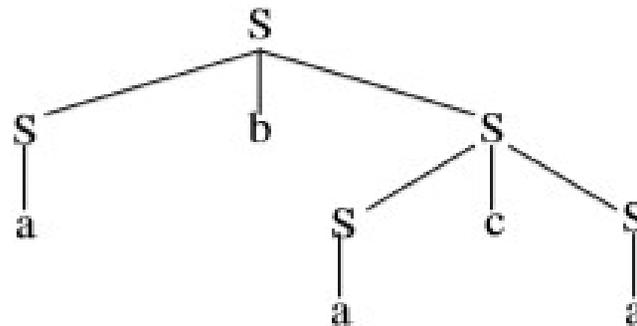
# Ambiguitas

Jika dari aturan produksi tata Bahasa bebas konteks terdapat lebih dari satu cara membuat pohon penurunan untuk memperoleh untai.

Contoh :

$S \rightarrow SbS \mid ScS \mid a$

Kita dapat memperoleh untai "abaca" dalam dua cara berikut ini.



$S \rightarrow AB \mid C$

$A \rightarrow aAb \mid ab$

$B \rightarrow cBd \mid cd$

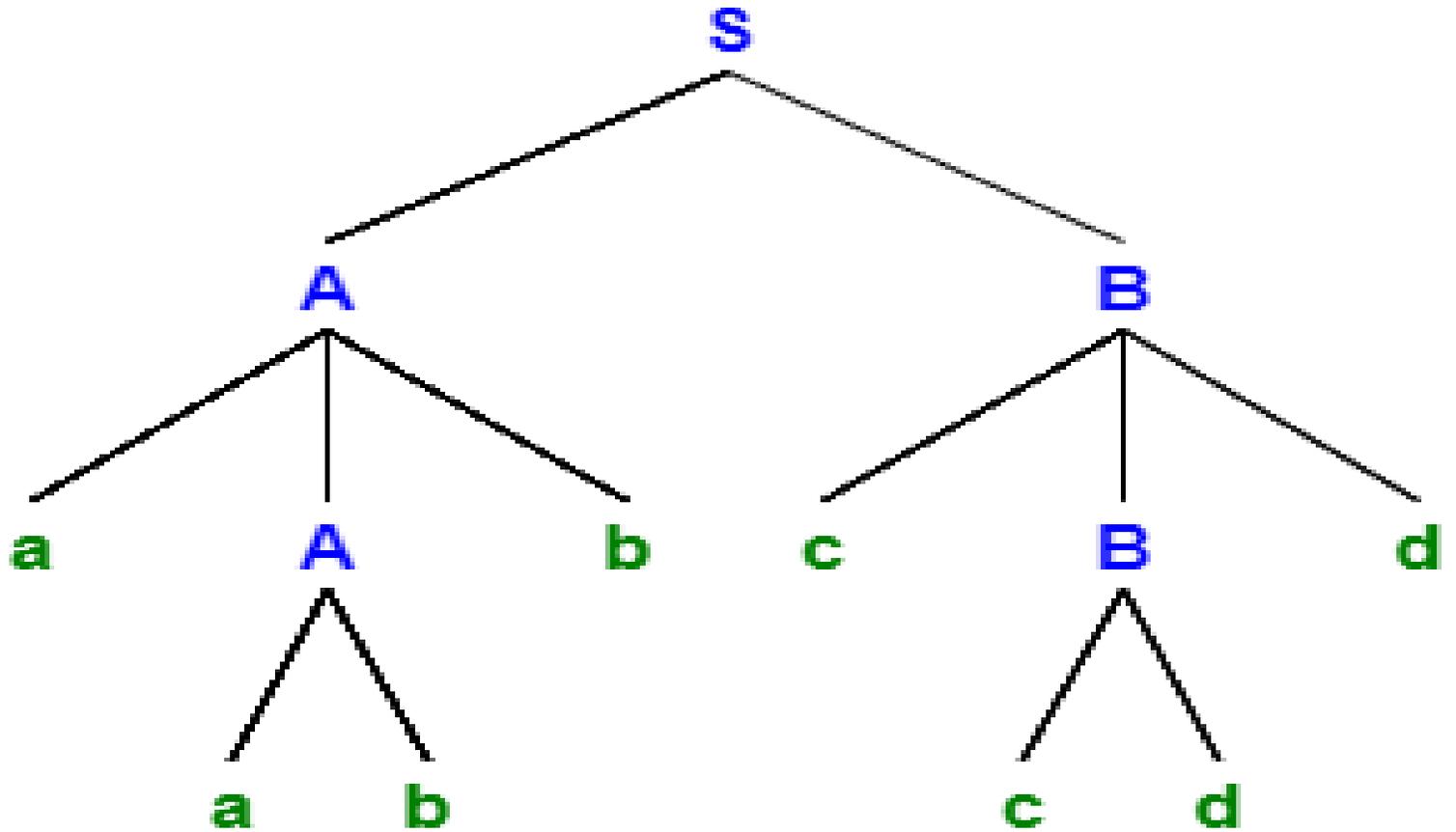
$C \rightarrow aCd \mid aDd$

$D \rightarrow bDc \mid bc$

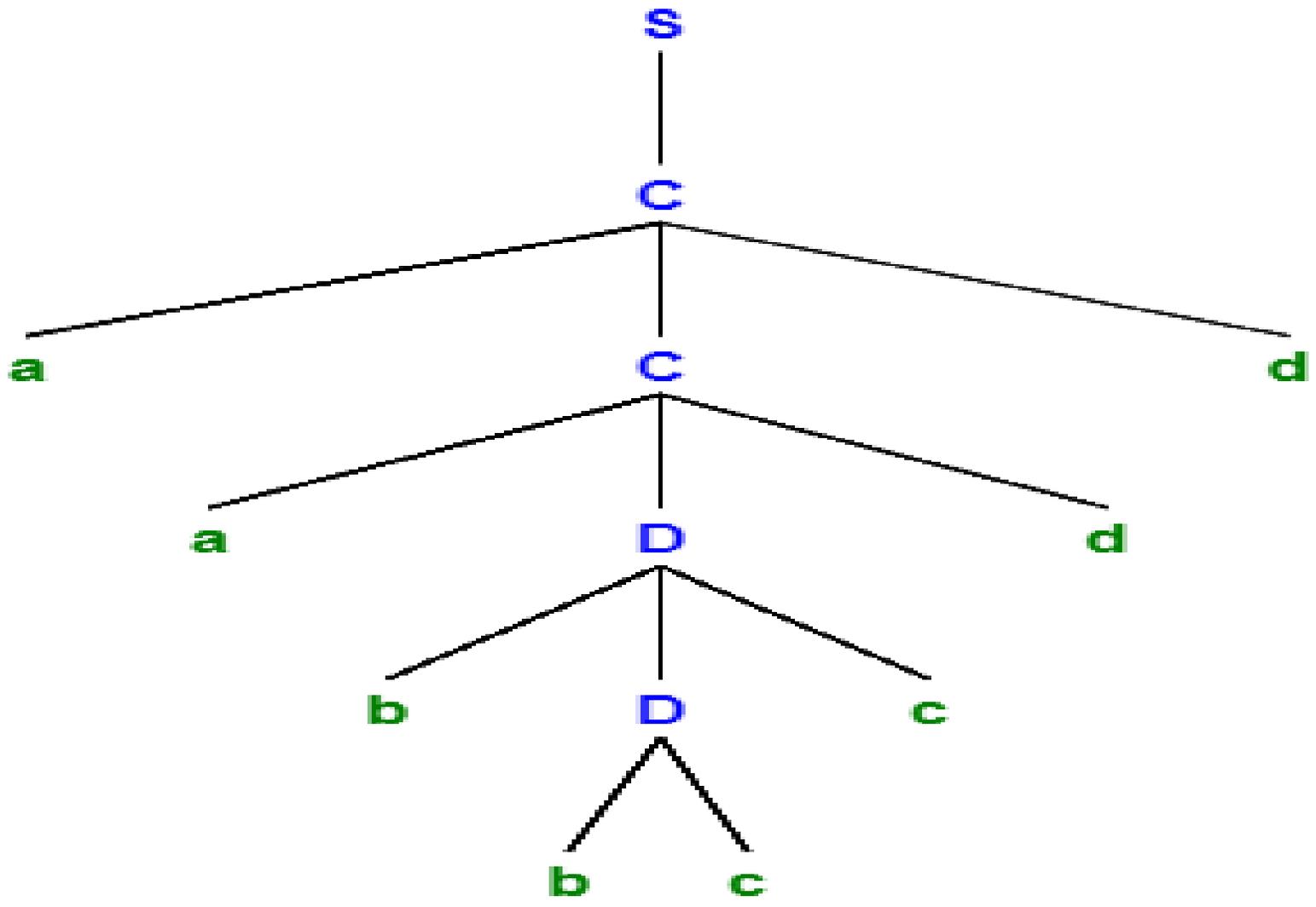
Buatlah pohon penurunan dari himpunan produksi diatas untuk membangkitkan string dengan susunan “**aabbccdd**”.

CFG dikatakan ambigu apabila terdapat penurunan yang dapat dikerjakan dengan dua cara sekaligus, lebih dari satu *leftmost derivation* dan/atau lebih dari satu *rightmost derivation*.

$S \Rightarrow AB \Rightarrow aAbcBd \Rightarrow aabbccdd$



$S \Rightarrow C \Rightarrow aCd \Rightarrow aaDdd \Rightarrow aabDcdd \Rightarrow aabbccdd$



$S \rightarrow ABCb \mid BCbAB$

$A \rightarrow ABa \mid a \mid b \mid c$

$B \rightarrow CBAb \mid BCAb \mid a \mid b \mid c$

$C \rightarrow a \mid b \mid c$

Cek string berikut diterima atau ditolak

1) abcbbcb

2) bbbbbbbac

3) aaabbbcb