

# Simplifications of Context-Free Grammars

# A Substitution Rule

$$S \rightarrow aB$$

$$A \rightarrow aaA$$

$$A \rightarrow abBc$$

$$B \rightarrow aA$$

$$B \rightarrow b$$

Substitute

$$B \rightarrow b$$

Equivalent  
grammar

$$S \rightarrow aB \mid ab$$

$$A \rightarrow aaA$$

$$A \rightarrow abBc \mid abbc$$

$$B \rightarrow aA$$

# A Substitution Rule

$$S \rightarrow aB \mid ab$$

$$A \rightarrow aaA$$

$$A \rightarrow abBc \mid abbc$$

$$B \rightarrow aA$$

Substitute

$$B \rightarrow aA$$

$$S \rightarrow \cancel{aB} \mid ab \mid aaA$$

$$A \rightarrow aaA$$

$$A \rightarrow \cancel{abBc} \mid abbc \mid abaAc$$

Equivalent  
grammar

In general:

$$A \rightarrow xBz$$

$$B \rightarrow y_1$$

Substitute

$$B \rightarrow y_1$$

$$A \rightarrow xBz \mid xy_1z$$

equivalent  
grammar

# Nullable Variables

$\lambda$  – production :  $A \rightarrow \lambda$

Nullable Variable:  $A \Rightarrow \dots \Rightarrow \lambda$

# Removing Nullable Variables

Example Grammar:

$$S \rightarrow aMb$$

$$M \rightarrow aMb$$

$$M \rightarrow \lambda$$



Nullable variable

$S \rightarrow aMb$

$M \rightarrow aMb$

~~$M \rightarrow \lambda$~~

Substitute  
 $M \rightarrow \lambda$

## Final Grammar

$S \rightarrow aMb$

$S \rightarrow ab$

$M \rightarrow aMb$

$M \rightarrow ab$

# Unit-Productions

Unit Production:  $A \rightarrow B$

(a single variable in both sides)



# Removing Unit Productions

Observation:

$$A \rightarrow A$$

Is removed immediately

## Example Grammar:

$$S \rightarrow aA$$

$$A \rightarrow a$$

$$A \rightarrow B$$

$$B \rightarrow A$$

$$B \rightarrow bb$$

$$S \rightarrow aA$$

$$A \rightarrow a$$

~~$$A \rightarrow B$$~~

$$B \rightarrow A$$

$$B \rightarrow bb$$

Substitute

$$A \rightarrow B$$

$$S \rightarrow aA \mid aB$$

$$A \rightarrow a$$

$$B \rightarrow A \mid B$$

$$B \rightarrow bb$$

$$S \rightarrow aA \mid aB$$

$$A \rightarrow a$$

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

$$B \rightarrow bb$$

Remove

$$B \rightarrow B$$

$$S \rightarrow aA \mid aB$$

$$A \rightarrow a$$

$$B \rightarrow A$$

$$B \rightarrow bb$$

$$S \rightarrow aA \mid aB$$

$$A \rightarrow a$$

~~$$B \rightarrow A$$~~

$$B \rightarrow bb$$

Substitute

$$B \rightarrow A$$

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

$$A \rightarrow a$$

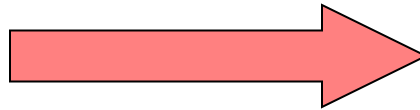
$$B \rightarrow bb$$

# Remove repeated productions

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

$A \rightarrow a$

$B \rightarrow bb$



## Final grammar

$S \rightarrow aA \mid aB$

$A \rightarrow a$

$B \rightarrow bb$

# Useless Productions

$$S \rightarrow aSb$$

$$S \rightarrow \lambda$$

$$S \rightarrow A$$

$$A \rightarrow aA \text{ Useless Production}$$

Some derivations never terminate...

$$S \Rightarrow A \Rightarrow aA \Rightarrow aaA \Rightarrow \dots \Rightarrow aa \dots aA \Rightarrow \dots$$

Another grammar:

$$S \rightarrow A$$

$$A \rightarrow aA$$

$$A \rightarrow \lambda$$

$$B \rightarrow bA \quad \text{Useless Production}$$

Not reachable from  $S$



In general:

contains only  
terminals

if  $S \Rightarrow \dots \Rightarrow xAy \Rightarrow \dots \Rightarrow w$

  
 $w \in L(G)$

then variable  $A$  is useful

otherwise, variable  $A$  is useless

A production  $A \rightarrow x$  is useless  
if any of its variables is useless

$$S \rightarrow aSb$$

$$S \rightarrow \lambda$$

Productions

Variables

$$S \rightarrow A$$

useless

useless

$$A \rightarrow aA$$

useless

useless

$$B \rightarrow C$$

useless

useless

$$C \rightarrow D$$

useless

# Removing Useless Productions

Example Grammar:

$$S \rightarrow aS \mid A \mid C$$

$$A \rightarrow a$$

$$B \rightarrow aa$$

$$C \rightarrow aCb$$

**First:** find all variables that can produce strings with only terminals

$$S \rightarrow aS \mid A \mid C$$

Round 1:  $\{A, B\}$

$$A \rightarrow a$$

$$S \rightarrow A$$

$$B \rightarrow aa$$

$$C \rightarrow aCb$$

Round 2:  $\{A, B, S\}$

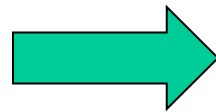
Keep only the variables  
that produce terminal symbols:  $\{A, B, S\}$   
(the rest variables are useless)

$$S \rightarrow aS \mid A \mid \cancel{C}$$

$$A \rightarrow a$$

$$B \rightarrow aa$$

$$\cancel{C \rightarrow aCb}$$



$$S \rightarrow aS \mid A$$

$$A \rightarrow a$$

$$B \rightarrow aa$$

Remove useless productions

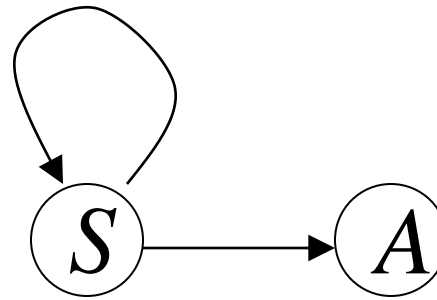
**Second:** Find all variables  
reachable from  $S$

Use a Dependency Graph

$S \rightarrow aS \mid A$

$A \rightarrow a$

$B \rightarrow aa$



$B$

not  
reachable

Keep only the variables  
reachable from  $S$

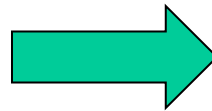
(the rest variables are useless)

Final Grammar

$$S \rightarrow aS \mid A$$

$$A \rightarrow a$$

~~$$B \rightarrow aa$$~~



$$S \rightarrow aS \mid A$$

$$A \rightarrow a$$

Remove useless productions

# Removing All

**Step 1:** Remove Nullable Variables

**Step 2:** Remove Unit-Productions

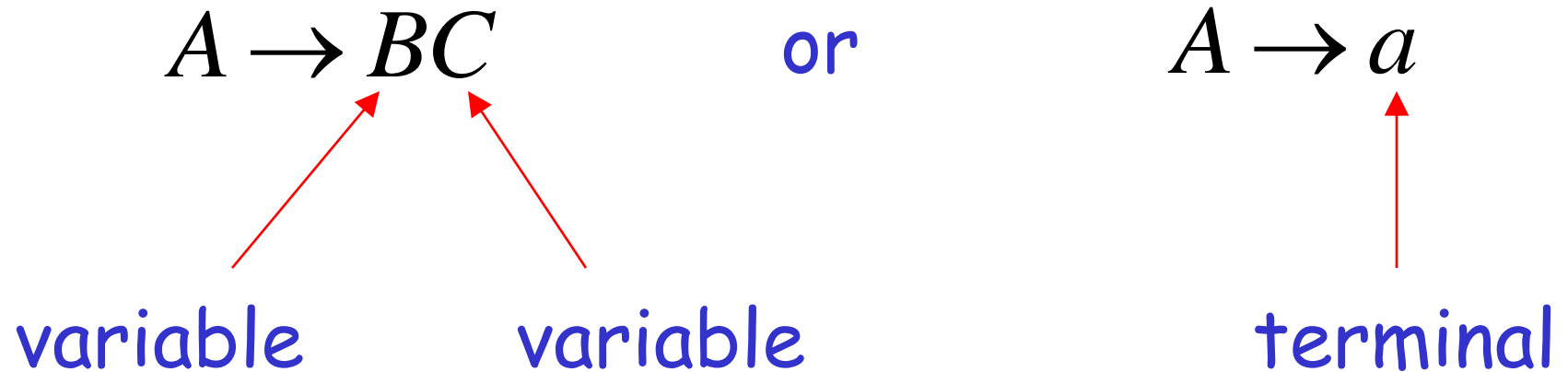
**Step 3:** Remove Useless Variables



# Normal Forms for Context-free Grammars

# Chomsky Normal Form

Each productions has form:



# Examples:

$$S \rightarrow AS$$

$$S \rightarrow a$$

$$A \rightarrow SA$$

$$A \rightarrow b$$

Chomsky  
Normal Form

$$S \rightarrow AS$$

$$S \rightarrow AAS$$

$$A \rightarrow SA$$

$$A \rightarrow aa$$

Not Chomsky  
Normal Form

# Conversion to Chomsky Normal Form

Example:  $S \rightarrow ABa$

$A \rightarrow aab$

$B \rightarrow Ac$

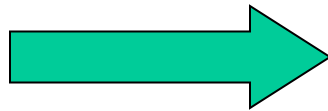
Not Chomsky  
Normal Form

Introduce variables for terminals:  $T_a, T_b, T_c$

$$S \rightarrow ABa$$

$$A \rightarrow aab$$

$$B \rightarrow Ac$$



$$S \rightarrow ABT_a$$

$$A \rightarrow T_aT_aT_b$$

$$B \rightarrow AT_c$$

$$T_a \rightarrow a$$

$$T_b \rightarrow b$$

$$T_c \rightarrow c$$

Introduce intermediate variable:  $V_1$

$$S \rightarrow ABT_a$$

$$A \rightarrow T_aT_aT_b$$

$$B \rightarrow AT_c$$

$$T_a \rightarrow a$$

$$T_b \rightarrow b$$

$$T_c \rightarrow c$$



$$S \rightarrow AV_1$$

$$V_1 \rightarrow BT_a$$

$$A \rightarrow T_aT_aT_b$$

$$B \rightarrow AT_c$$

$$T_a \rightarrow a$$

$$T_b \rightarrow b$$

$$T_c \rightarrow c$$

Introduce intermediate variable:  $V_2$

$$S \rightarrow AV_1$$

$$V_1 \rightarrow BT_a$$

$$A \rightarrow T_aT_aT_b$$

$$B \rightarrow AT_c$$

$$T_a \rightarrow a$$

$$T_b \rightarrow b$$

$$T_c \rightarrow c$$



$$S \rightarrow AV_1$$

$$V_1 \rightarrow BT_a$$

$$A \rightarrow T_aV_2$$

$$V_2 \rightarrow T_aT_b$$

$$B \rightarrow AT_c$$

$$T_a \rightarrow a$$

$$T_b \rightarrow b$$

$$T_c \rightarrow c$$

# Final grammar in Chomsky Normal Form:

$$S \rightarrow AV_1$$

$$V_1 \rightarrow BT_a$$

$$A \rightarrow T_aV_2$$

$$V_2 \rightarrow T_aT_b$$

$$B \rightarrow AT_c$$

$$T_a \rightarrow a$$

$$T_b \rightarrow b$$

$$T_c \rightarrow c$$

## Initial grammar

$$S \rightarrow ABa$$

$$A \rightarrow aab$$

$$B \rightarrow Ac$$



In general:

From any context-free grammar  
(which doesn't produce  $\lambda$ )  
not in Chomsky Normal Form

we can obtain:

An equivalent grammar  
in Chomsky Normal Form

# The Procedure

First remove:

Nullable variables

Unit productions

Then, for every symbol  $a$ :

Add production  $T_a \rightarrow a$

In productions: replace  $a$  with  $T_a$

New variable:  $T_a$

Replace any production  $A \rightarrow C_1 C_2 \cdots C_n$

with  $A \rightarrow C_1 V_1$

$V_1 \rightarrow C_2 V_2$

...

$V_{n-2} \rightarrow C_{n-1} C_n$

New intermediate variables:  $V_1, V_2, \dots, V_{n-2}$

**Theorem:** For any context-free grammar  
(which doesn't produce  $\lambda$ )  
there is an equivalent grammar  
in Chomsky Normal Form

# Observations

- Chomsky normal forms are good for parsing and proving theorems
- It is very easy to find the Chomsky normal form for any context-free grammar