Simplifications
of
Context-Free Grammars

A Substitution Rule

$S \to aB$

$A \to aaA$

$A \to abBc$

$B \to aA$

$B \to b$

Substitute
$B \to b$

Equivalent grammar

$S \to aB \mid ab$

$A \to aaA$

$A \to abBc \mid abbc$

$B \to aA$

A Substitution Rule

$$S \to aB \mid ab$$
$$A \to aaA$$
$$A \to abBc \mid abbc$$
$$B \to aA$$

Substitute $B \to aA$

$$S \to \cancel{aB} \mid ab \mid aaA$$
$$A \to aaA$$
$$A \to \cancel{abBc} \mid abbc \mid abaAc$$

Equivalent grammar

In general:

$$A \to xBz$$

$$B \to y_1$$

Substitute $B \to y_1$

$$A \to xBz \mid xy_1z$$

equivalent grammar

Nullable Variables

$\lambda - \text{production}:$ $A \rightarrow \lambda$

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

5

Removing Nullable Variables

Example Grammar:

$$S \rightarrow aMb$$
$$M \rightarrow aMb$$
$$M \rightarrow \lambda$$

Nullable variable

6

Final Grammar

$$S \to aMb$$
$$M \to aMb$$
$$\cancel{M \to \lambda}$$

Substitute $M \to \lambda$

$$S \to aMb$$
$$S \to ab$$
$$M \to aMb$$
$$M \to ab$$

Unit-Productions

Unit Production: $A \to B$

(a single variable in both sides)

Removing Unit Productions

Observation:

$$A \to A$$

Is removed immediately

Example Grammar:

$$S \to aA$$
$$A \to a$$
$$A \to B$$
$$B \to A$$
$$B \to bb$$

$$S \to aA$$
$$A \to a$$
$$\cancel{A \to B}$$
$$B \to A$$
$$B \to bb$$

Substitute $A \to B$

$$S \to aA \mid aB$$
$$A \to a$$
$$B \to A \mid B$$
$$B \to bb$$

$$S \to aA \mid aB$$
$$A \to a$$
$$B \to A \mid \cancel{B}$$
$$B \to bb$$

Remove $B \to B$

$$S \to aA \mid aB$$
$$A \to a$$
$$B \to A$$
$$B \to bb$$

$$S \to aA \mid aB$$
$$A \to a$$
$$\cancel{B \to A}$$
$$B \to bb$$

Substitute
$B \to A$

$$S \to aA \mid aB \mid aA$$
$$A \to a$$
$$B \to bb$$

Remove repeated productions

$$S \to aA \mid aB \mid \cancel{aA}$$
$$A \to a$$
$$B \to bb$$

Final grammar

$$S \to aA \mid aB$$
$$A \to a$$
$$B \to bb$$

Useless Productions

$$S \rightarrow aSb$$
$$S \rightarrow \lambda$$
$$S \rightarrow A$$
$$A \rightarrow aA$$

Useless Production

Some derivations never terminate...

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

15

Another grammar:

$$S \rightarrow A$$
$$A \rightarrow aA$$
$$A \rightarrow \lambda$$
$$B \rightarrow bA$$

Useless Production

Not reachable from S

16

In general:

contains only terminals

if $S \Rightarrow \ldots \Rightarrow xAy \Rightarrow \ldots \Rightarrow w$

$w \in L(G)$

then variable A is useful

otherwise, variable A is useless

17

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

Variables		Productions
	$S \rightarrow aSb$	
	$S \rightarrow \lambda$	
	$S \rightarrow A$	useless
useless	$A \rightarrow aA$	useless
useless	$B \rightarrow C$	useless
useless	$C \rightarrow D$	useless

18

Removing Useless Productions

Example Grammar:

$$S \to aS \mid A \mid C$$
$$A \to a$$
$$B \to aa$$
$$C \to aCb$$

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

$$S \to aS \mid A \mid C$$
$$A \to a$$
$$B \to aa$$
$$C \to aCb$$

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

$S \to A$

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

Keep only the variables
that produce terminal symbols: $\{A, B, S\}$

(the rest variables are useless)

$$S \to aS \mid A \mid \cancel{C}$$
$$A \to a$$
$$B \to aa$$
$$\cancel{C \to aCb}$$

$$S \to aS \mid A$$
$$A \to a$$
$$B \to aa$$

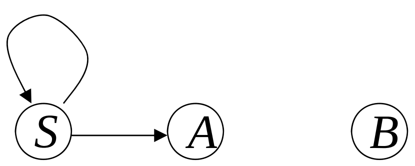
Remove useless productions

21

Second: Find all variables reachable from S

Use a Dependency Graph

$$S \to aS \mid A$$
$$A \to a$$
$$B \to aa$$



not reachable

22

Keep only the variables reachable from S

(the rest variables are useless)

Final Grammar

$$S \to aS \mid A$$
$$A \to a$$
$$\cancel{B \to aa}$$

$$S \to aS \mid A$$
$$A \to a$$

Remove useless productions

23

Removing All

Step 1: Remove Nullable Variables

Step 2: Remove Unit-Productions

Step 3: Remove Useless Variables

24

Normal Forms
for
Context-free Grammars

Chomsky Normal Form

Each productions has form:

$$A \to BC \qquad \text{or} \qquad A \to a$$

variable variable terminal

Examples:

$$S \to AS$$
$$S \to a$$
$$A \to SA$$
$$A \to b$$

Chomsky Normal Form

$$S \to AS$$
$$S \to AAS$$
$$A \to SA$$
$$A \to aa$$

Not Chomsky Normal Form

Convertion to Chomsky Normal Form

Example:

$$S \to ABa$$
$$A \to aab$$
$$B \to Ac$$

Not Chomsky Normal Form

Introduce variables for terminals: T_a, T_b, T_c

$$S \to ABa$$
$$A \to aab$$
$$B \to Ac$$

$$\Rightarrow$$

$$S \to ABT_a$$
$$A \to T_aT_aT_b$$
$$B \to AT_c$$
$$T_a \to a$$
$$T_b \to b$$
$$T_c \to c$$

29

Introduce intermediate variable: V_1

$$S \to ABT_a$$
$$A \to T_aT_aT_b$$
$$B \to AT_c$$
$$T_a \to a$$
$$T_b \to b$$
$$T_c \to c$$

$$\Rightarrow$$

$$S \to AV_1$$
$$V_1 \to BT_a$$
$$A \to T_aT_aT_b$$
$$B \to AT_c$$
$$T_a \to a$$
$$T_b \to b$$
$$T_c \to c$$

30

Introduce intermediate variable: V_2

$$S \to AV_1$$
$$V_1 \to BT_a$$
$$A \to T_aT_aT_b$$
$$B \to AT_c$$
$$T_a \to a$$
$$T_b \to b$$
$$T_c \to c$$

$\Longrightarrow$

$$S \to AV_1$$
$$V_1 \to BT_a$$
$$A \to T_aV_2$$
$$V_2 \to T_aT_b$$
$$B \to AT_c$$
$$T_a \to a$$
$$T_b \to b$$
$$T_c \to c$$

31

Final grammar in Chomsky Normal Form:

Initial grammar

$$S \to ABa$$
$$A \to aab$$
$$B \to Ac$$

$$S \to AV_1$$
$$V_1 \to BT_a$$
$$A \to T_aV_2$$
$$V_2 \to T_aT_b$$
$$B \to AT_c$$
$$T_a \to a$$
$$T_b \to b$$
$$T_c \to c$$

32

In general:

From any context-free grammar
(which doesn't produce λ)
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 $T_a \to a$

In productions: replace a with T_a

New variable: T_a

35

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

with

$$A \to C_1 V_1$$
$$V_1 \to C_2 V_2$$
$$\cdots$$
$$V_{n-2} \to C_{n-1} C_n$$

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

36

Theorem: For any context-free grammar (which doesn't produce λ) 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

Greibach Normal Form

All productions have form:

$$A \rightarrow a\, V_1 V_2 \cdots V_k \qquad k \geq 0$$

symbol variables

39

Examples:

$$S \rightarrow cAB$$
$$A \rightarrow aA \mid bB \mid b$$
$$B \rightarrow b$$

Greinbach Normal Form

$$S \rightarrow abSb$$
$$S \rightarrow aa$$

Not Greinbach Normal Form

40

Conversion to Greinbach Normal Form:

$$S \to abSb$$
$$S \to aa$$

$$S \to aT_bST_b$$
$$S \to aT_a$$
$$T_a \to a$$
$$T_b \to b$$

Greibach Normal Form

Theorem: For any context-free grammar (which doesn't produce λ) there is an equivalent grammar in Greinbach Normal Form

Observations

- Greinbach normal forms are very good for parsing

- It is hard to find the Greinbach normal form of any context-free grammar

The CYK Parser

The CYK Membership Algorithm

Input:

- Grammar G in Chomsky Normal Form
- String w

Output:

find if $w \in L(G)$

45

The Algorithm

Input example:

- Grammar G:

$$S \to AB$$
$$A \to BB$$
$$A \to a$$
$$B \to AB$$
$$B \to b$$

- String w: $aabbb$

46

aabbb

a	a	b	b	b
aa	ab	bb	bb	
aab	abb	bbb		
aabb	abbb			
aabbb				

47

$S \to AB$

$A \to BB$

$A \to a$

$B \to AB$

$B \to b$

a	a	b	b	b
A	A	B	B	B
aa	ab	bb	bb	
aab	abb	bbb		
aabb	abbb			
aabbb				

48

$S \to AB$

$A \to BB$

$A \to a$

$B \to AB$

$B \to b$

a A	a A	b B	b B	b B
aa	ab S,B	bb A	bb A	
aab	abb	bbb		
aabb	abbb			
aabbb				

49

$S \to AB$

$A \to BB$

$A \to a$

$B \to AB$

$B \to b$

a A	a A	b B	b B	b B
aa	ab S,B	bb A	bb A	
aab S,B	abb A	bbb S,B		
aabb A	abbb S,B			
aabbb S,B				

50

Therefore: $aabbb \in L(G)$

Time Complexity: $|w|^3$

Observation: The CYK algorithm can be easily converted to a parser (bottom up parser)