Context Free Grammars

Shakir Ullah Shah

Outline

- Removing Λ-Productions
- Removing Unit-Productions
- Removing Useless Variables
- Normal Form of Context Free Grammars
  - Chomsky Normal Form (CNF)

Killing Λ-Productions

Λ-Productions:

In a given CFG, we call a non-terminal N null able

if there is a production N → Λ, or there is a

derivation that starts at N and lead to a Λ.

N ⇒ ……… ⇒ Λ

- Λ-Productions are undesirable.
- We can replace Λ-production with appropriate non-Λ productions

Theorem 23

- If L is CFL generated by a CFG having Λ-productions, then there is a different CFG that has no Λ-production and still generates either the whole language L (if L does not include Λ) or else generate the language of all the words in L other than Λ.
- Replacement Rule.
  1. Delete all Λ-Productions.
  2. Add the following productions:
- For every production of the X ✉ old string

Add new production of the form X ✉ .., where right side will account for every modification of the old string that can be formed by deleting all possible subsets of null-able Non-Terminals, except that we do not allow X ✉ Λ, to be formed if all the character in old string are null-able

Example

Consider the CFG

S → Xa

X → aX | bX | Λ

Old nullable Production	New Production
S → Xa	S → a
X → aX	X → a
X → bX	X → b

So the new CFG is

S → Xa | a

X → aX | bX | a | b

Example

S → XY

X → Zb

Y → bW

Z → AB

W → Z

A → aA | bA | Λ

B → Ba | Bb | Λ

- Null-able Non-terminals are?
- A, B, Z and W

Example Contd.

S → XY
X → Zb
Y → bW
Z → AB
W → Z
A → aA | bA | Λ
B → Ba | Bb | Λ

Old nullable Production	New Production
X → Zb	X → b
Y → bW	Y → b
Z → AB	Z → A and Z → B
W → Z	Nothing new
A → aA	A → a
A → bA	A → b
B → Ba	B → a

So the new CFG is
S → XY
X → Zb | b
Y → bW | b
Z → AB | A | B
W → Z
A → aA | bA | a | b
B → Ba | Ba | a | b

Killing unit-productions

- **Definition:** A production of the form
- Nonterminal ✉ one Nonterminal

is called a **unit production**.

- The following theorem allows us to get rid of unit productions:

Theorem 24:

If there is a CFG for the language L that has no

Λ-productions, then there is also a CFG for L

with

Proof of Theorem 24

- This is another proof by constructive algorithm.
- **Algorithm:** For every pair of nonterminals A and B, if the CFG has a unit production A $\rightarrow$ B, or if there is a chain

A $\Rightarrow$ X1 $\Rightarrow$ X2 $\Rightarrow$ … $\Rightarrow$ B

where X1, X2, … are nonterminals, create new productions

as follows:

- If the non-unit productions from B are

B $\rightarrow$ s1 | s2| …

where s1, s2, … are strings, we create the productions

A $\rightarrow$ s1 | s2 | …

Example

- Consider the CFG

  S → A| bb

  A → B | b

  B → S | a

- The non-unit productions are

  S → bb A → b B → a

- And unit productions are

  S → A

  A → B

Example contd.

- Let's list all unit productions and their sequences and create new productions:

S → A	gives	S → b
S → A → B	gives	S → a
A → B	gives	A → a
A → B → S	gives	A → bb
B → S	gives	B → bb
B → S → A	gives	B → b

- Eliminating all unit productions, the new CFG is

S → bb | b | a

A → b | a | bb

B → a | bb | b

Consider the CFG

S → A| bb

A → B | b

B → S | a

unit productions are

S → A

A → B

B → S

Another grammar:

$$S \to A$$

$$A \to aA$$

$$A \to \lambda$$

$$B \to bA$$

Useless Production

Not reachable from S

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

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

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

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)

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

$$A \to a$$

$$B \to aa$$

$$\cancel{C \to aCb}$$

$$S \to aS \mid A$$

$$A \to a$$

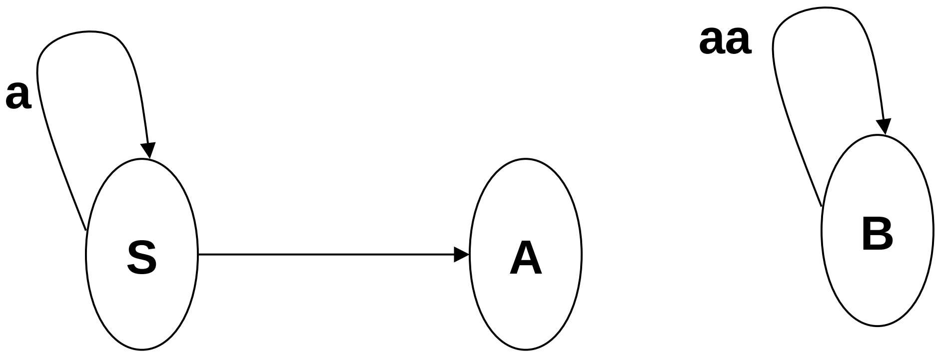
$$B \to aa$$

Remove useless productions

Second: Find all variables reachable from S

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

Use a Dependency Graph



not reachable

Keep only the variables reachable from S

(the rest variables are useless)

$$S \to aS \mid A$$

$$A \to a$$

$$\cancel{B \to aa}$$

Final Grammar

$$S \to aS \mid A$$

$$A \to a$$

Remove useless productions

Normal Form of Context Free Grammars

Normal Form

- To convert a grammar in normal form means “standardizing the productions without changing the resulting language”.

THE CHOMSKY NORMAL FORM

A context-free grammar is in Chomsky normal form if every rule is of the form:

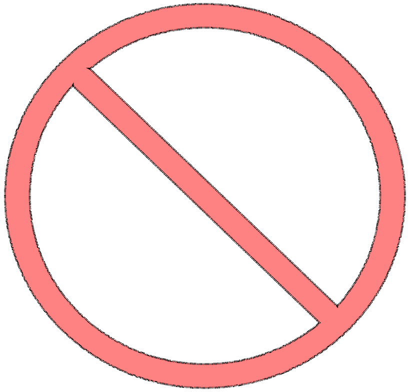
$A \to BC$	**B and C are not start variables**
$A \to a$	**a is a terminal**
$S \to \varepsilon$	**S is the start variable and** if **ε is in the language**

Any variable A that is not the start variable can only generate strings of length > 0

THE CHOMSKY NORMAL FORM

A context-free grammar is in Chomsky normal form if every rule is of the form:

$A \rightarrow BC$	**B and C are not start variable**
$A \rightarrow a$	**a is a terminal**
$S \rightarrow \varepsilon$	**S is the start variable**



- $S \rightarrow AS \mid a$
- $A \rightarrow SA \mid b$

Is in Chomsky Normal Form. The Grammar

- $S \rightarrow AS \mid ASS$
- $A \rightarrow SA \mid aa$
- Is not; both production $S \rightarrow AS \mid ASS$ and $A \rightarrow SA \mid aa$ violate the conditions of definition.

Theorem 25

- If L is a language generated by some CFG, then there exist another CFG that generate all the non-null words of L, all of whose productions are of one of the two basic forms:

Non-Terminal ✉ string of only Non-terminals

Or

Non-Terminal ✉ one Terminal

Proof

- The proof is a constructive algorithm by converting a given CFG in a CFG of the production of designed format.

- Let us suppose in given CFG the non-terminal are S, X1, X2, X3, … . Let us also assume {a,b} are two terminals.

STEP-1

We now add two new non-terminals A and B

Proof Theorem 25 contd…

STEP-2

now for every previous production involving terminals, we replace each ‘a’ with the non-terminal A and each ‘b’ with B.

e.g. X3 → X4aX1SbbX7a

Becomes

X3 → X4AX1SBBX7A

Which is a string of solid non-terminals

X6 → abaab

Becomes

X6 → ABAAB

Example

S → X1 | X2aX2 | aSb | b

X1 → X2X2 | b

X2 → aX2 | aaX2

S → X1 | X2AX2 | ASB | B

X1 → X2X2 | B

X2 → AX2 | AAX2

A → a

B → b

Theorem: Any context-free language can be generated by a context-free grammar in Chomsky normal form

“Can transform any CFG into Chomsky normal form”

Chomsky Normal Form

There are 5 steps to follow in order to transform a grammar into CNF:

1. Add the a new start variable S0 and the production rule S0 $\longrightarrow \mathrm{S}$.
2. Eliminate the ε-rules.
3. Eliminate the unary productions A $\longrightarrow$ B.
4. Add rules of the form Vt $\longrightarrow$ t for every terminal t and replace t with the variable Vt.
5. Transform the remaining of the rules to the

1. Add a new start variable

- We have to make sure that the start variable doesn’t occur to the right side of some rule.
- Thus, we add a new start variable S0 and the rule S0 $\longrightarrow$ S, where S is the old start variable.

2. Eliminate ε-rules

- We have to eliminate all productions of the form $A \longrightarrow \varepsilon$, for A being any non-start variable.
- To do so we should remove the rule $A \longrightarrow \varepsilon$ and replace every appearance of A with ε in all other rules.

3. Eliminate unary productions

- A unary production is a production of the form A $\longrightarrow$ B (with both A, B being variables).
- There should only be productions of the form V1 $\longrightarrow$ V2V3 involving variables, thus we have to eliminate unary productions.
- To do so, we replace B in A $\longrightarrow$ B with the right parts of the rules involving B in the left part.

4. Add Vt $\longrightarrow$ t and replace t with Vt

- There should only be rules of the form A $\longrightarrow$ t involving terminals, thus terminals should disappear from every other rule involving more than just one single literal.
- To do so, we add a new variable Vt for every terminal t and we replace every appearance of t with Vt , except those in rules of the form A $\longrightarrow$ t.

5. Transform rules to A $\longrightarrow$ BC

- All the rules involving only variables should be of the form A $\longrightarrow$ BC. Thus we should take care of all the rules involving more than 2 variables in the right part
- For the rule V $\longrightarrow$ A1A2A3...An,we start reducing the size of the right part by replacing every two variables with one new variable (resulting in the creation of n-2 new variables).

5. Transform rules to A $\longrightarrow$ BC

V $\longrightarrow$ $A1A2A3A4A5A6...An$

5. Transform rules to A $\longrightarrow$ BC

V $\longrightarrow \mathrm{B1A3A4A5A6...An}$

B1 $\longrightarrow \mathrm{A1A2}$

5. Transform rules to A $\longrightarrow$ BC

V $\longrightarrow \mathrm{B2A4A5A6...An}$

B2 $\longrightarrow \mathrm{B1A3}$

B1 $\longrightarrow \mathrm{A1A2}$

5. Transform rules to A $\longrightarrow$ BC

V $\longrightarrow$ $\mathrm{B3A5A6...An}$

B3 $\longrightarrow$ $\mathrm{B2A4}$

B2 $\longrightarrow$ $\mathrm{B1A3}$

B1 $\longrightarrow$ $\mathrm{A1A2}$

5. Transform rules to A $\longrightarrow$ BC

V $\longrightarrow \mathrm{B4A6...An}$

B4 $\longrightarrow \mathrm{B3A5}$

B3 $\longrightarrow \mathrm{B2A4}$

B2 $\longrightarrow \mathrm{B1A3}$

B1 $\longrightarrow \mathrm{A1A2}$

5. Transform rules to A $\longrightarrow$ BC

V $\longrightarrow$ Bn-2An

Bn-2 $\longrightarrow$ Bn-3An-1

...

B4 $\longrightarrow$ B3A5

B3 $\longrightarrow$ B2A4

B2 $\longrightarrow$ B1A3

B1 $\longrightarrow$ A1A2

Example

$S \longrightarrow CSC \mid B$

$C \longrightarrow 00 \mid \varepsilon$

$B \longrightarrow 01B \mid 1$

Example

1. <u>Add new start variablle</u>

$S0 \longrightarrow S$

$S \longrightarrow CSC \mid B$

$C \longrightarrow 00 \mid \varepsilon$

$B \longrightarrow 01B \mid 1$

Example

<u>2. Eliminate ε-moves</u>

$S0 \longrightarrow S$

$S \longrightarrow CSC \mid B$

$C \longrightarrow 00 \mid \cancel{\varepsilon}$

$B \longrightarrow 01B \mid 1$

Example

2. Eliminate ε-moves

$S0 \longrightarrow S$

$S \longrightarrow CSC \mid B \mid CS \mid SC \mid S$

$C \longrightarrow 00$

$B \longrightarrow 01B \mid 1$

Example

3. Eliminate Unary Productions

$S0 \longrightarrow S$

$S \longrightarrow CSC \mid B \mid CS \mid SC \mid \cancel{S}$

$C \longrightarrow 00$

$B \longrightarrow 01B \mid 1$

Example

3. Eliminate Unary Productions

$S0 \longrightarrow S$

$S \longrightarrow CSC \mid B \mid CS \mid SC$ -

$C \longrightarrow 00$

$B \longrightarrow 01B \mid 1$

Example

<u>3. Eliminate Unary Productions</u>

$S0 \longrightarrow S$

$S \longrightarrow CSC \mid \cancel{B} \mid CS \mid SC$

$C \longrightarrow 00$

$B \longrightarrow 01B \mid 1$

Example

3. Eliminate Unary Productions

$S0 \longrightarrow S$

$S \longrightarrow CSC \mid 01B \mid 1 \mid CS \mid SC$

$C \longrightarrow 00$

$B \longrightarrow 01B \mid 1$

Example

<u>3. Eliminate Unary Productions</u>

S0 $\longrightarrow$ $\cancel{S}$

S $\longrightarrow$ CSC | 01B | 1 | CS | SC

C $\longrightarrow$ 00

B $\longrightarrow$ 01B | 1

Example

3. Eliminate Unary Productions

$S0 \longrightarrow CSC \mid 01B \mid 1 \mid CS \mid SC$

$S \longrightarrow CSC \mid 01B \mid 1 \mid CS \mid SC$

$C \longrightarrow 00$

$B \longrightarrow 01B \mid 1$

Example

4. <u>Create Vt for every terminal t</u>

$S0 \longrightarrow$ CSC | 01B | 1 | CS | SC

$S \longrightarrow$ CSC | 01B | 1 | CS | SC

$C \longrightarrow$ 00

$B \longrightarrow$ 01B | 1

$Z \longrightarrow 0$

Example

4. Create Vt for every terminal t

$S0 \longrightarrow CSC \mid Z1B \mid 1 \mid CS \mid SC$

$S \longrightarrow CSC \mid Z1B \mid 1 \mid CS \mid SC$

$C \longrightarrow ZZ$

$B \longrightarrow Z1B \mid 1$

$Z \longrightarrow 0$

Example

4. Create Vt for every terminal t

S0 $\longrightarrow$ CSC | Z1B | 1 | CS | SC

S $\longrightarrow$ CSC | Z1B | 1 | CS | SC

C $\longrightarrow$ ZZ

B $\longrightarrow$ Z1B | 1

$\text{Z} \longrightarrow 0$

$\text{A} \longrightarrow 1$

Example

4. Create Vt for every terminal t

S0 $\longrightarrow$ CSC | ZAB | 1 | CS | SC

S $\longrightarrow$ CSC | ZAB | 1 | CS | SC

C $\longrightarrow$ ZZ

B $\longrightarrow$ ZAB | 1

$Z \longrightarrow 0$

$A \longrightarrow 1$

Example

5. Take care of long rules

S0 $\longrightarrow$ CSC | ZAB | 1 | CS | SC

S $\longrightarrow$ CSC | ZAB | 1 | CS | SC

C $\longrightarrow$ ZZ

B $\longrightarrow$ ZAB | 1

$Z \longrightarrow 0$

$A \longrightarrow 1$

$D \longrightarrow CS$

Example

5. Take care of long rules

S0 $\longrightarrow$ DC | ZAB | 1 | CS | SC

S $\longrightarrow$ DC | ZAB | 1 | CS | SC

C $\longrightarrow$ ZZ

B $\longrightarrow$ ZAB | 1

$\mathrm{Z} \longrightarrow 0$

$\mathrm{A} \longrightarrow 1$

$\mathrm{D} \longrightarrow \mathrm{CS}$

Example

5. Take care of long rules

S0 $\longrightarrow$ DC | ZAB | 1 | CS | SC

S $\longrightarrow$ DC | ZAB | 1 | CS | SC

C $\longrightarrow$ ZZ

B $\longrightarrow$ ZAB | 1

$\mathrm{Z} \longrightarrow 0$

$\mathrm{A} \longrightarrow 1$

$\mathrm{D} \longrightarrow \mathrm{CS}$

Example

5. Take care of long rules

S0 $\longrightarrow$ DC | EB | 1 | CS | SC

S $\longrightarrow$ DC | EB | 1 | CS | SC

C $\longrightarrow$ ZZ

B $\longrightarrow$ EB | 1

$Z \longrightarrow 0$

$A \longrightarrow 1$

$D \longrightarrow CS$

Convert the follow[redacted]nto Chomsky normal form:

A [redacted] BAB | B | ε

B [redacted] 00 | ε

S0 → BC | DD | BB | AB | BA | ε, C → AB,

A → BC | DD | BB | AB | BA , B → DD, D → 0