Theory of Automata
Context Free Grammars

Hafiz Tayyeb Javed

Week-10-Lecture-02

Removal of Null and Unit Productions and Useless Productions

Contents

- Simplification of CFGs
  - Killing Λ-Productions
  - Killing unit-productions
  - Removing Useless Variables

Killing Λ-Productions

Λ-Productions:

In a given CFG, we call a non-terminal N ***nullable***

- if there is a production N → Λ, or
- there is a derivation that starts at N and lead to a Λ.

$$N \Rightarrow \ldots\ldots\ldots \Rightarrow \Lambda$$

- 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(s) 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

$S \rightarrow a \mid Xb \mid aYa$

$X \rightarrow Y \mid \Lambda$

$Y \rightarrow b \mid X$

X is nullable

Y is nullable

Old nullable Production	New Production
$X \rightarrow Y$	nothing
$X \rightarrow \Lambda$	nothing
$Y \rightarrow X$	nothing
$S \rightarrow Xb$	$S \rightarrow b$
$S \rightarrow aYa$	$S \rightarrow aa$

So the new CFG is

$S \rightarrow a \mid Xb \mid aa \mid aYa \mid b$

$X \rightarrow Y$

$Y \rightarrow b \mid X$

Example

Consider the CFG

$S \rightarrow Xa$

$X \rightarrow aX \mid bX \mid \Lambda$

X is nullable

Old nullable Production	New Production
$S \rightarrow Xa$	$S \rightarrow a$
$X \rightarrow aX$	$X \rightarrow a$
$X \rightarrow bX$	$X \rightarrow b$

So the new CFG is

$S \rightarrow a \mid Xa$

$X \rightarrow aX \mid bX \mid a \mid b$

Example

$S \rightarrow XY$

$X \rightarrow Zb$

$Y \rightarrow bW$

$Z \rightarrow AB$

$W \rightarrow Z$

$A \rightarrow aA \mid bA \mid \Lambda$

$B \rightarrow Ba \mid Bb \mid \Lambda$

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

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

Example Contd.

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

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
B → Bb	B → b

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

Remove Nulls

(a + b)*bb(a + b)*

$S \to XY$
$X \to Zb$
$Y \to bW$
$Z \to AB$
$W \to Z$
$A \to aA \mid bA \mid \Lambda$
$B \to Ba \mid Bb \mid \Lambda$

Old	**Additional New Productions Derived from Old**
$X \to Zb$	$X \to b$
$Y \to bW$	$Y \to b$
$Z \to AB$	$Z \to A$ and $Z \to B$
$W \to Z$	Nothing
$A \to aA$	$A \to a$
$A \to bA$	$A \to b$
$B \to Ba$	$B \to a$
$B \to Bb$	$B \to b$

$$S \rightarrow XY$$
$$X \rightarrow Zb \mid b$$
$$Y \rightarrow bW \mid b$$
$$Z \rightarrow AB \mid A \mid B$$
$$W \rightarrow Z$$
$$A \rightarrow aA \mid bA \mid a \mid b$$
$$B \rightarrow Ba \mid Bb \mid a \mid b$$

Killing unit-productions

- **Definition:** A production of the form
  - non-terminal → one non-terminal

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 no Λ-productions and **no unit productions**.

Proof of Theorem 24

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

  $A \rightarrow X_1 \rightarrow X_2 \rightarrow \ldots \rightarrow B$

  where X_1, X_2, ... are non-terminals, create new productions as follows:
- If the non-unit productions from B are

  $B \rightarrow s_1 \mid s_2 \mid \ldots$

where s_1, s_2, ... are strings, we create the productions

$A \rightarrow s_1 \mid s_2 \mid \ldots$

Example

- Consider the CFG

  $S \rightarrow A \mid bb$

  $A \rightarrow B \mid b$

  $B \rightarrow S \mid a$

- The non-unit productions are

  $S \rightarrow bb$ $\quad$ $A \rightarrow b$ $\quad$ $B \rightarrow a$

- And unit productions are

$S \rightarrow A$

$A \rightarrow B$

$B \rightarrow S$

Example contd.

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

$S \rightarrow A$	gives	$S \rightarrow b$
$S \rightarrow A \rightarrow B$	gives	$S \rightarrow a$
$A \rightarrow B$	gives	$A \rightarrow a$
$A \rightarrow B \rightarrow S$	gives	$A \rightarrow bb$
$B \rightarrow S$	gives	$B \rightarrow bb$
$B \rightarrow S \rightarrow A$	gives	$B \rightarrow b$

The CFG

$S \rightarrow A \mid bb$
$A \rightarrow B \mid b$
$B \rightarrow S \mid a$

- Eliminating all unit productions, the new CFG is

  $S \rightarrow bb \mid b \mid a$

  $A \rightarrow b \mid a \mid bb$

  $B \rightarrow a \mid bb \mid b$

- This CFG generates a finite language since there are no non-terminals in any strings produced from S.

Useless Symbols

- A symbol that is not useful is useless
- Let a CFG G. A symbol X ε (V U ∑) is useful if there is a derivation

$$S \overset{*}{\underset{G}{\Rightarrow}} UxV \overset{*}{\underset{G}{\Rightarrow}} w$$

Where U and V ε (V U ∑) and w ε ∑*.

- A terminal is useful if it occurs in a string of the language of G.
- A variable is useful if it occurs in a derivation that begins from S and generates a terminal string

For a variable to be useful two conditions must be satisfied.

1. The variable must occur in a sentential form of the grammar
2. There must be a derivation of a terminal string from the variable.

- A variable that occurs in a sentential form is said to be reachable from S.
- A two part procedure is presented to eliminate useless symbols.

Algorithm to remove useless symbols

PART-I

Identify variables that derive terminal strings

Remove non-terminals that do not derive terminal strings.

e.g. following grammar G = $S \rightarrow aS \mid A \mid C$

$A \rightarrow a$

$B \rightarrow aa$

$C \rightarrow CB$

Example

S → aS | A | C
A → a
B → aa
C → CB

We can identify variables that derive terminal strings.
i.e. A → a B → aa
And S ➔A ➔ a TERM = {S, A, B}
But not C. thus C is useless

Example

PART - II

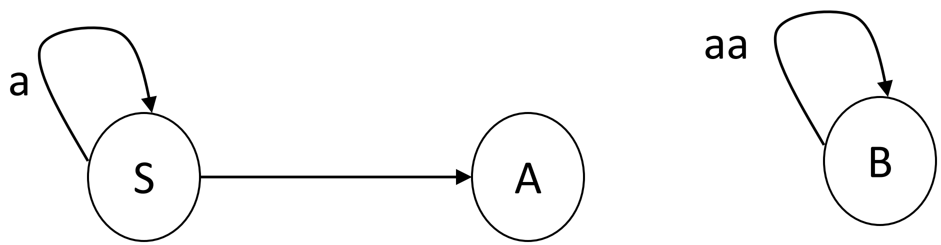
Rename the grammar

GT = $S \rightarrow aS \mid A$

$A \rightarrow a$

$B \rightarrow aa$

Now draw a graph and delete nodes not reachable from S.



As B is unreachable so delete it and the final grammar will be G_V=

$S \rightarrow aS \mid A$

$A \rightarrow a$

Useless Productions

$$S \rightarrow aSb$$

$$S \rightarrow \lambda$$

$$S \rightarrow A$$

$$A \rightarrow aA$$ Useless Production

Some derivations never terminate...

Another grammar:

$$S \to A$$

$$A \to aA$$

$$A \to \lambda$$

$$B \to bA$$ Useless Production

Not reachable from S

In general:

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

and w contains only terminals

$w \in L(G)$

then variable A is useful

otherwise, variable A is useless

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

$$S \rightarrow aSb$$

$$S \rightarrow \lambda$$

Variables		Productions
	$S \rightarrow A$	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

$$S \to aS \mid A \mid C$$

$$A \to a$$

$$B \to aa$$

$$C \to aCb$$

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

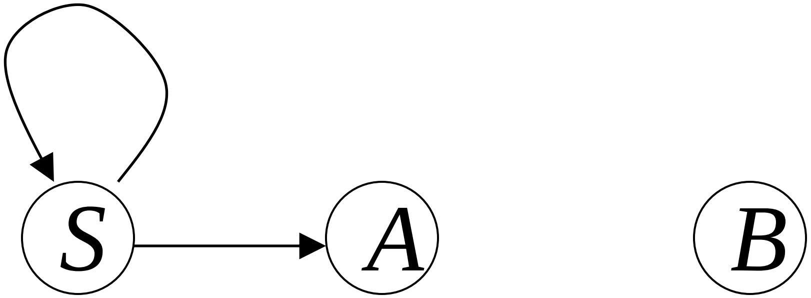
Second: Find all variables reachable from S

Use a Dependency Graph

$$S \rightarrow aS \mid A$$

$$A \rightarrow a$$

$$B \rightarrow aa$$



not reachable

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

Set of variables that Derive terminal symbols

- Input = CFG (V, ∑, P , S)

- TERM = { A | there is a rule $A \rightarrow w$ ε P with w ε ∑*
- repeat
  - PREV = TERM
  - For each variable in A ε V do
    - If there is a rule $A \rightarrow w$ and w ε (PREV U ∑)* then TERM = TERM U {A}
- Until PREV = TERM

Example

- Consider following CFG

G: $S \rightarrow AC \mid BS \mid B$

$A \rightarrow aA \mid aF$

$B \rightarrow CF \mid b$

$C \rightarrow cC \mid D$

$D \rightarrow aD \mid BD \mid C$

$E \rightarrow aA \mid BSA$

$F \rightarrow bB \mid b$

S → AC | BS | B
A → aA | aF
B → CF | b
C → cC | D
D → aD | BD | C
E → aA | BSA
F → bB | b

- New Grammar from TERM will be

G_T:
S → BS | B
A → aA | aF
B → b
E → aA | BSA
F → bB | b

Iteration	TERM	PREV
0	{B, F}	{}
1	{B, F, A, S}	{B, F}
2	{B, F, A, S, E}	{B, F, A, S}
3	{B, F, A, S, E}	{B, F, A, S, E}