COM3064
Automata Theory

Week 9: The Pumping Lemma for CFLs

Lecturer: Dr. Sevgi YİĞİT SERT
Spring 2023

Resources: Introduction to The Theory of Computation, M. Sipser,
Introduction to Automata Theory, Languages, and Computation, J.E. Hopcroft, R. Motwani, and J.D. Ullman
BBM401 Automata Theory and Formal Languages, İlyas Çiçekli
CENG280 Formal Languages and Abstract Machines, Halit Oğuztüzün

The Pumping Lemma for CFLs

Intuition:

- The **pumping lemma of regular languages** tell us that
  - **If there was a string long enough to cause a cycle in the DFA for the language, then we could "pump" the cycle (a piece of the string) and discover an infinite sequence of strings that had to be in the language.**

- The **pumping lemma of context-free languages** tell us that
  - **If there was a string long enough to cause a cycle (same variable appears more than once in the derivation), then we can always find two pieces of this sufficiently long string to "pump" in tandem and discover an infinite sequence of strings that had to be in the language.**
  - **That is: if we repeat each of the two pieces the same number of times, we get another string in the language.**

The Pumping Lemma for CFLs

- Let's take an **infinite** context-free language which generates an infinite number of different strings

Example:

$$S \rightarrow AB$$

$$A \rightarrow aBb$$

$$B \rightarrow Sb$$

$$B \rightarrow b$$

The Pumping Lemma for CFLs

- In a derivation of a long string, variables are repeated

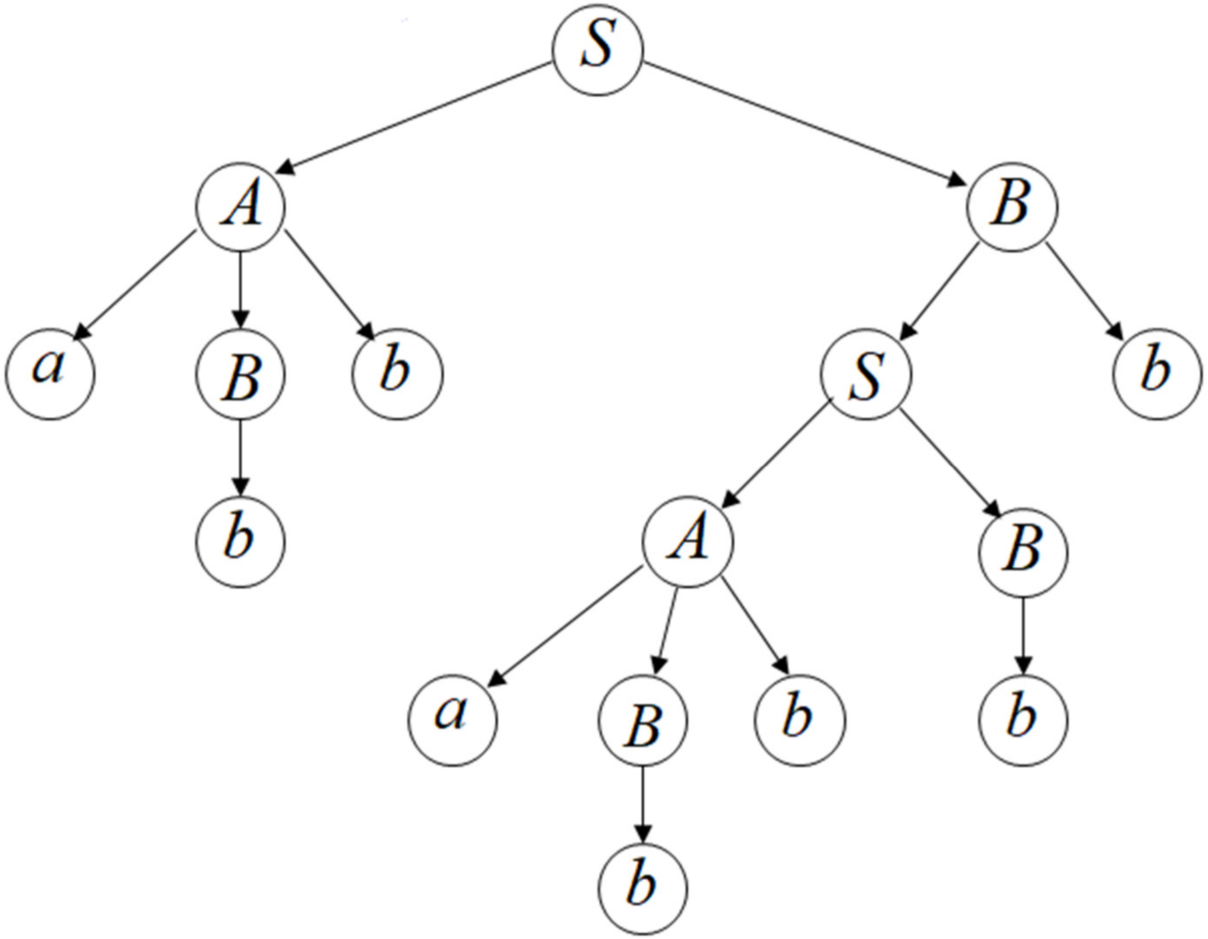
$$S \rightarrow AB$$
$$A \rightarrow aBb$$
$$B \rightarrow Sb$$
$$B \rightarrow b$$

A derivation:

$$S \Rightarrow AB \Rightarrow aBbB \Rightarrow abbB$$
$$\Rightarrow abbSb \Rightarrow abbABb \Rightarrow abbaBbBb \Rightarrow$$
$$\Rightarrow abbabbBb \Rightarrow abbabbbb$$

The Pumping Lemma for CFLs

Derivation tree of string : $abbabbbb$



The Pumping Lemma for CFLs

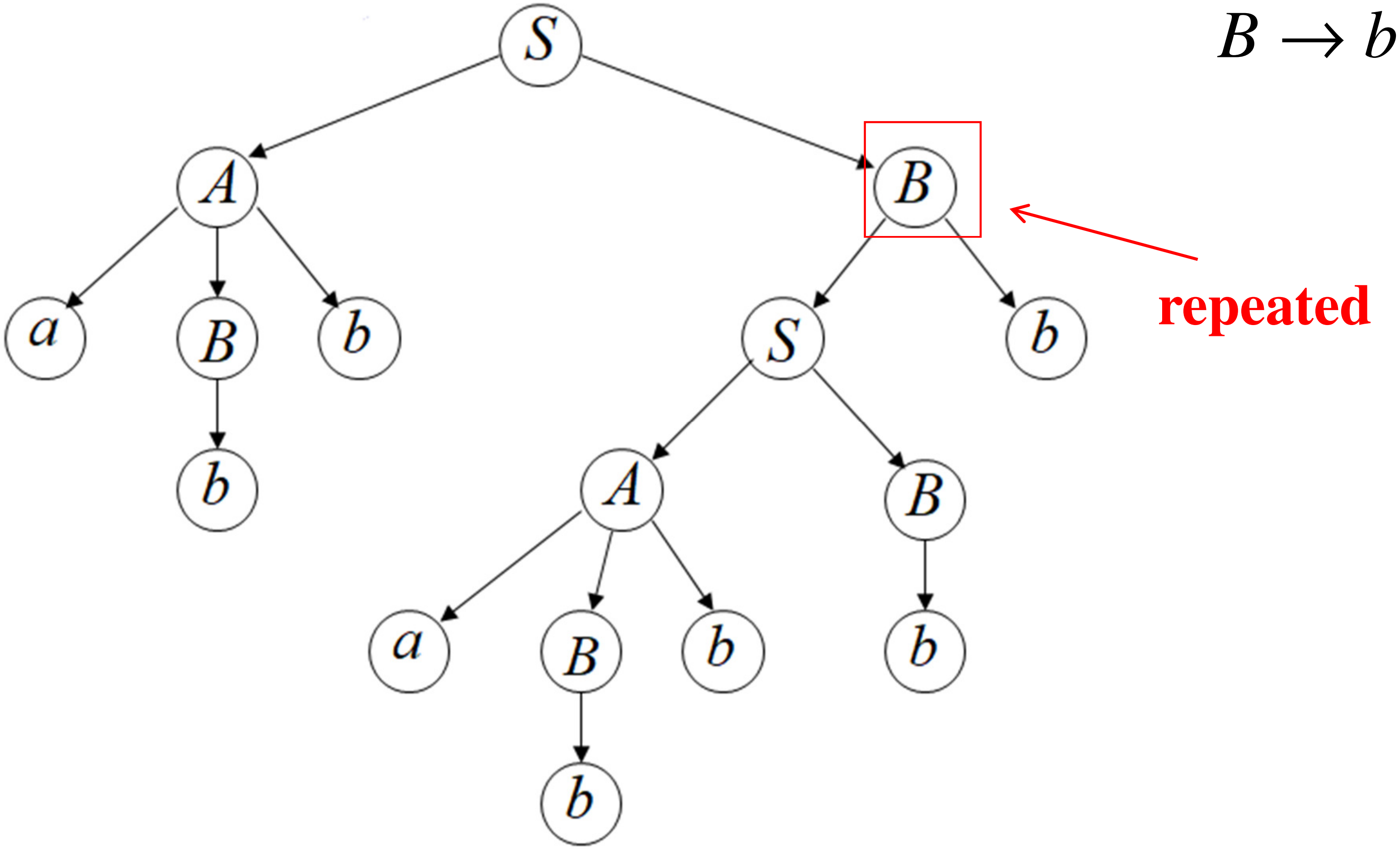
$S \rightarrow AB$

$A \rightarrow aBb$

$B \rightarrow Sb$

$B \rightarrow b$

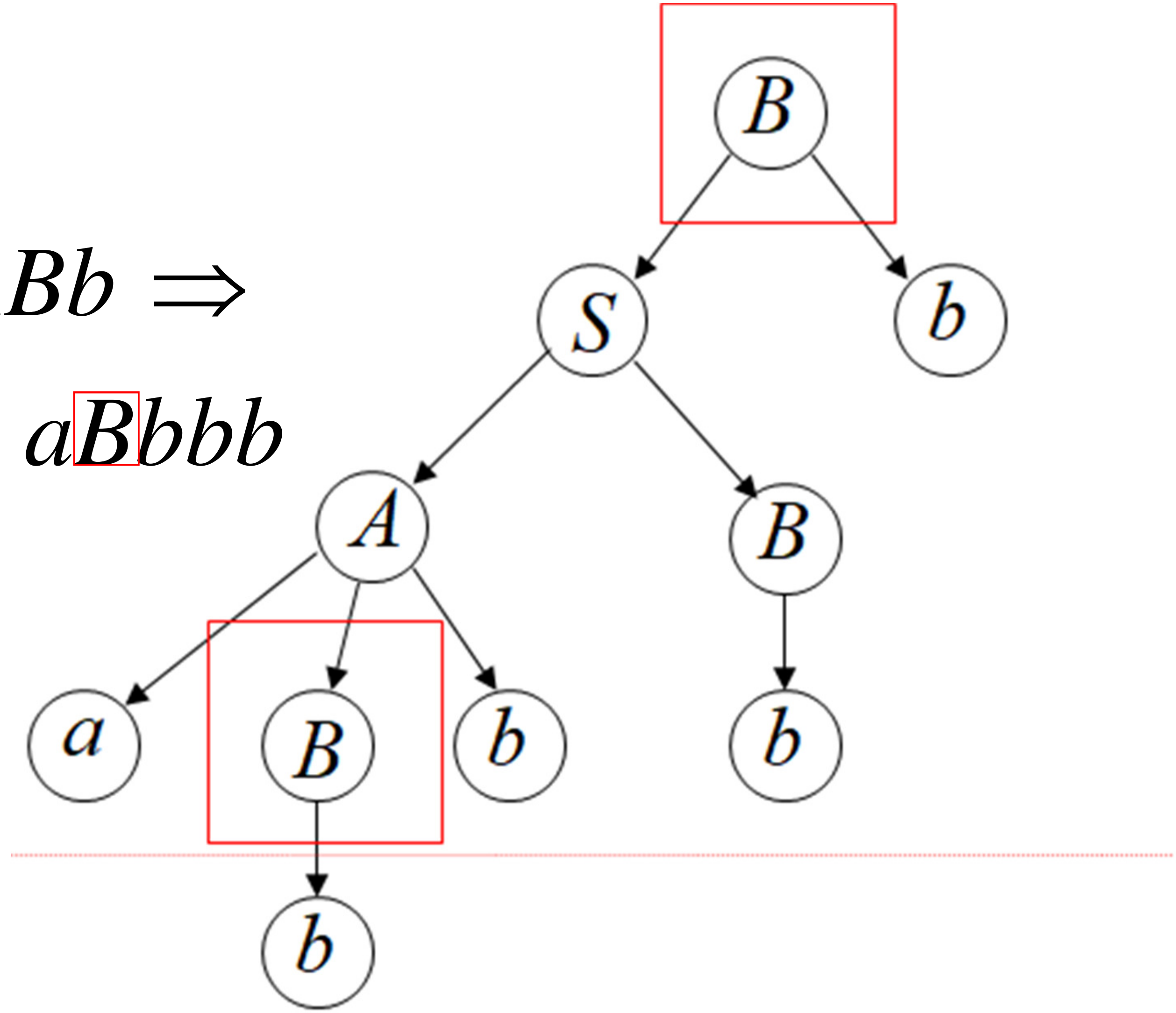
Derivation tree of string : $abbabbbb$



Repeated Part

$$\boxed{B} \Rightarrow Sb \Rightarrow ABb \Rightarrow$$

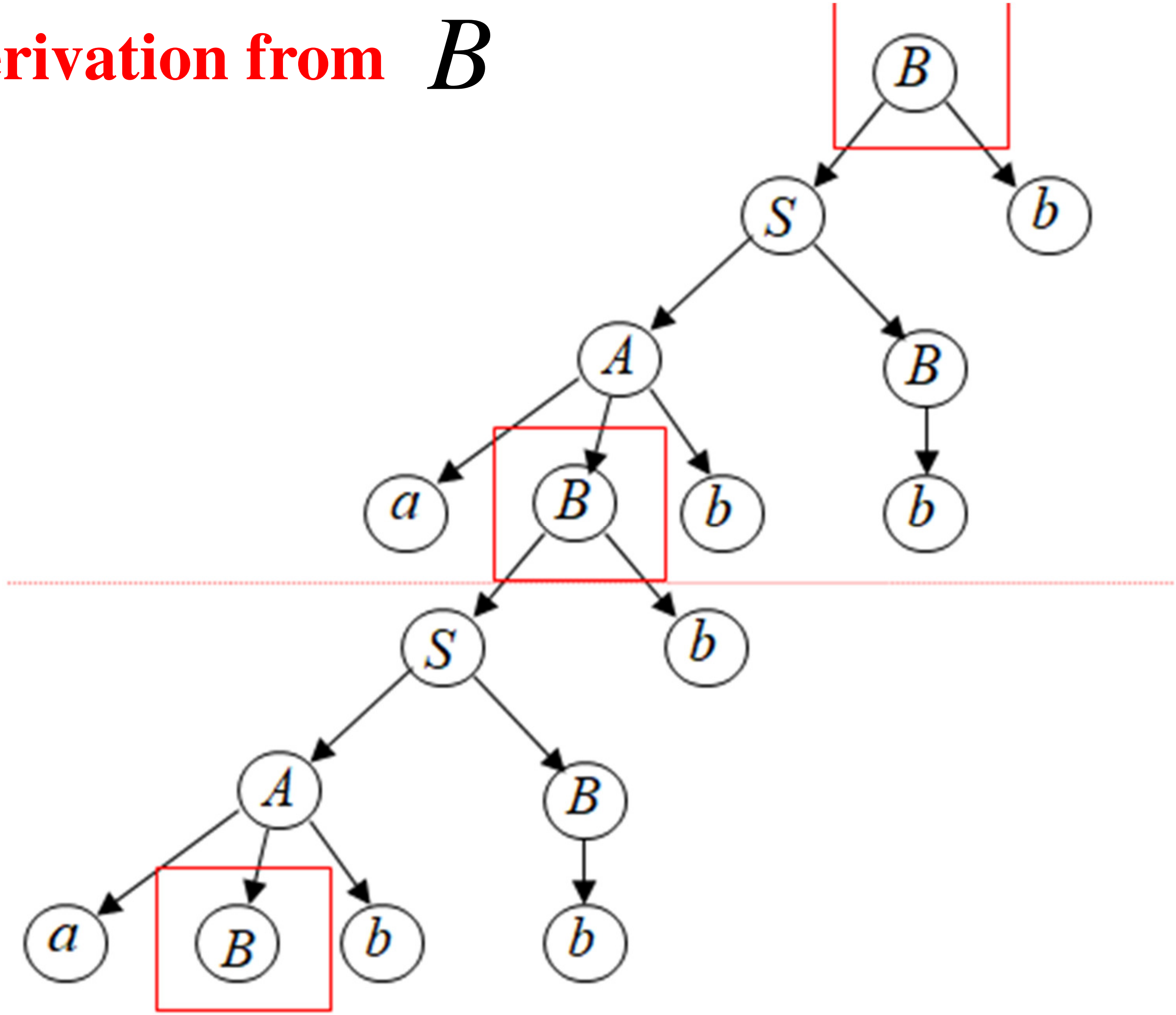
$$\Rightarrow aBbBb \Rightarrow a\boxed{B}bbb$$



$$B \overset{*}{\Rightarrow} aBbbb$$

$$B \Rightarrow b$$

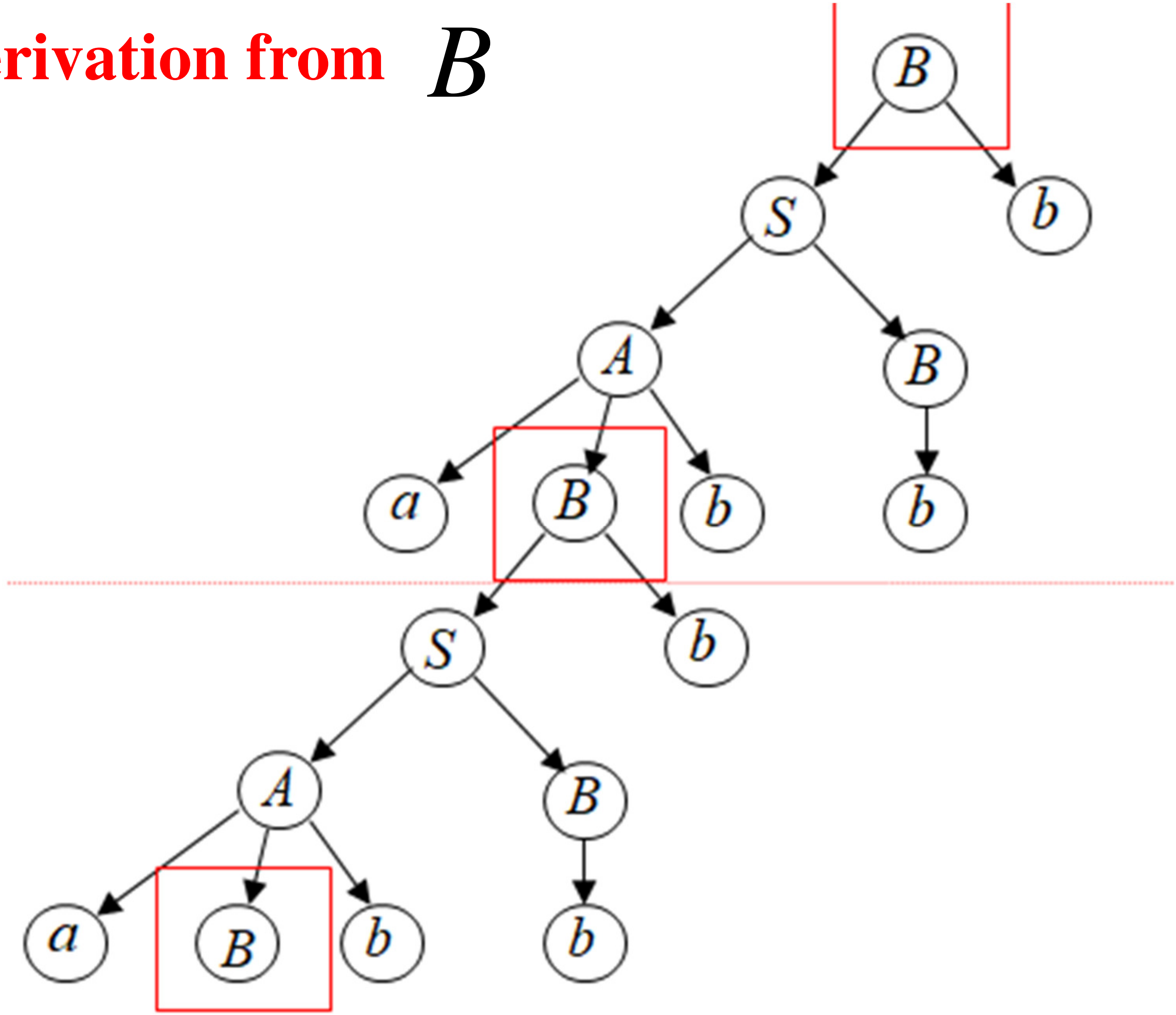
Another possible derivation from B



$$B \overset{*}{\Rightarrow} aBbbb$$

$$B \overset{*}{\Rightarrow} aBbbb \overset{*}{\Rightarrow} aaBbbbbbb$$

Another possible derivation from B



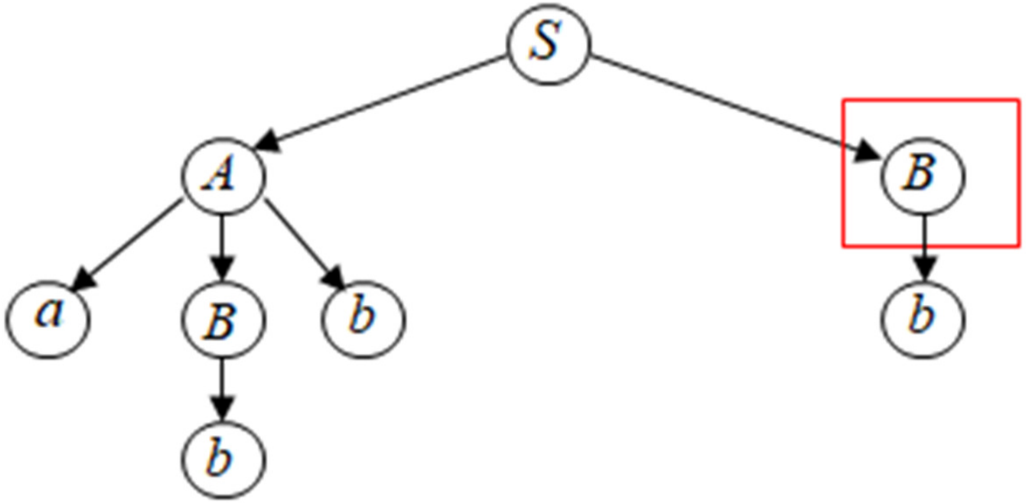
$$B \overset{*}{\Rightarrow} aBbbb$$

$$B \overset{*}{\Rightarrow} (a)B(bbb) \overset{*}{\Rightarrow} (a)^2 B(bbb)^2$$

A derivation from S

$$B \overset{*}{\Rightarrow} aBbbb$$

$$B \Rightarrow b$$



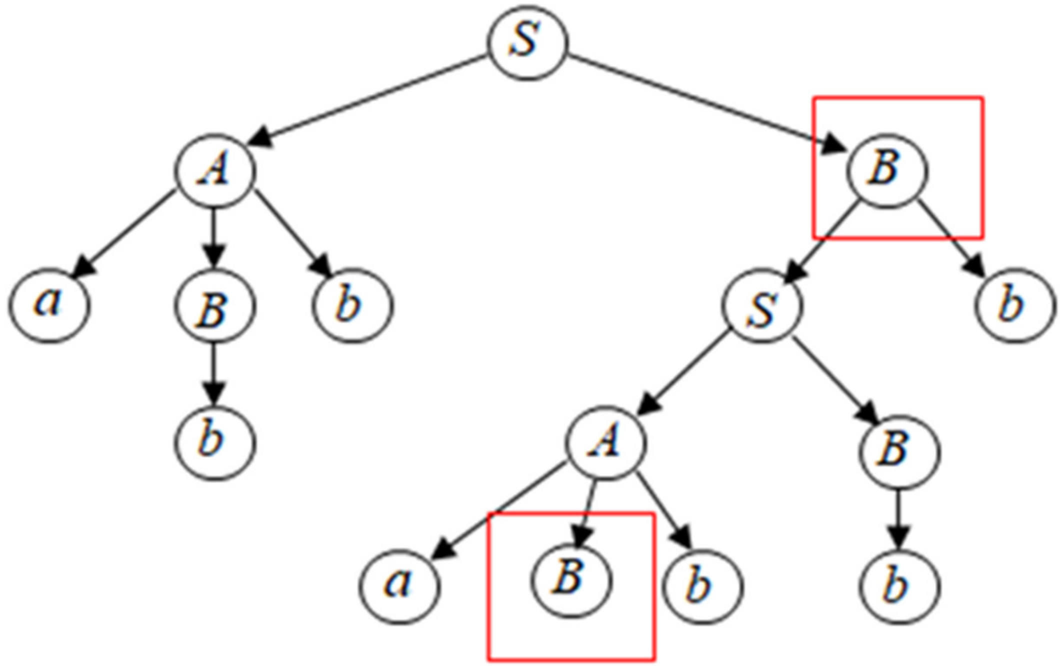
$$S \overset{*}{\Rightarrow} abbB = abbb = abb(a)^0 b(bbb)^0$$

$$abb(a)^0 b(bbb)^0 \in L(G)$$

A derivation from S

$$B \overset{*}{\Rightarrow} aBbbb$$

$$B \Rightarrow b$$

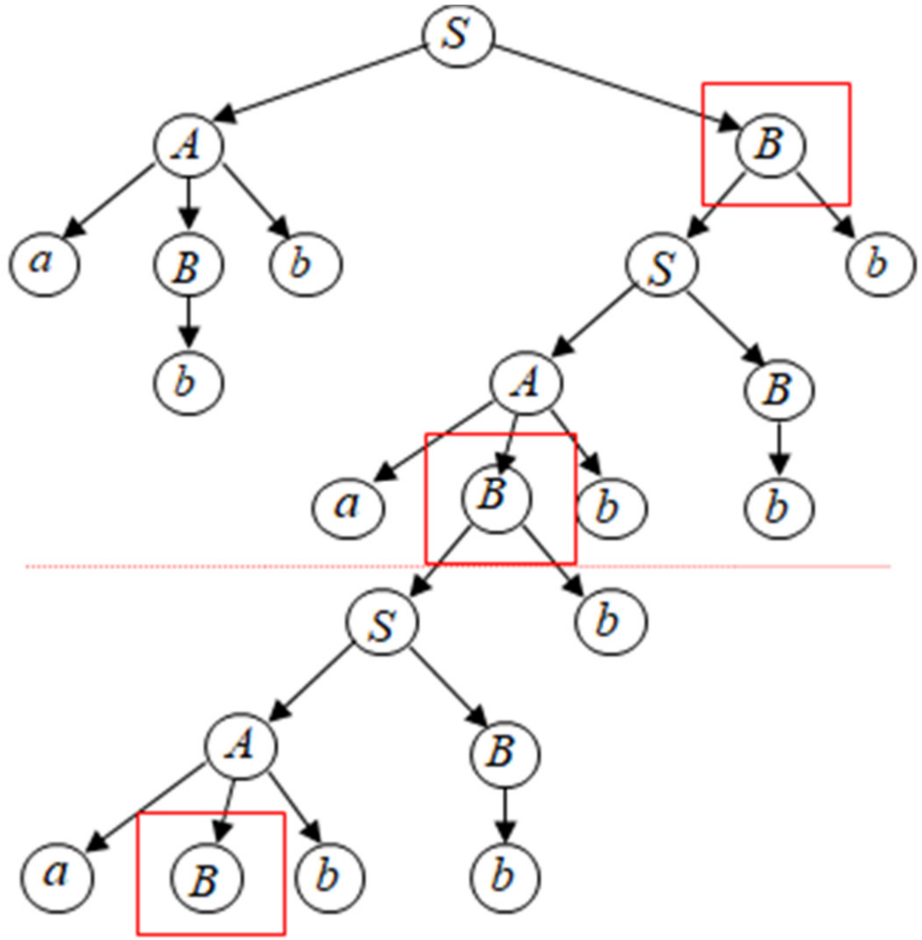


$$S \overset{*}{\Rightarrow} abbBb \overset{*}{\Rightarrow} abbaBbbb \overset{*}{\Rightarrow} abb(a)B(bbb)$$

A derivation from S

$$B \overset{*}{\Rightarrow} aBbbb$$

$$B \Rightarrow b$$



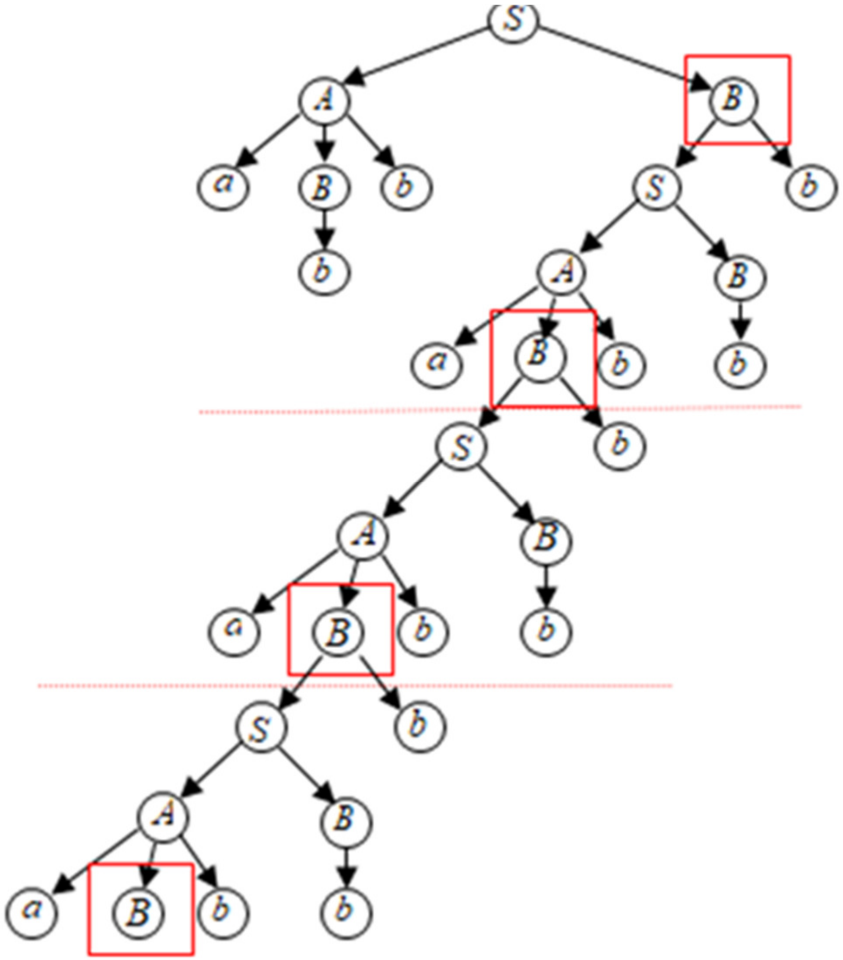
$$S \overset{*}{\Rightarrow} abb(a)B(bbb) \overset{*}{\Rightarrow} abb(a)^2 B(bbb)^2$$

$$abb(a)^2 b(bbb)^2 \in L(G)$$

A derivation from S

$$B \overset{*}{\Rightarrow} aBbbb$$

$$B \Rightarrow b$$

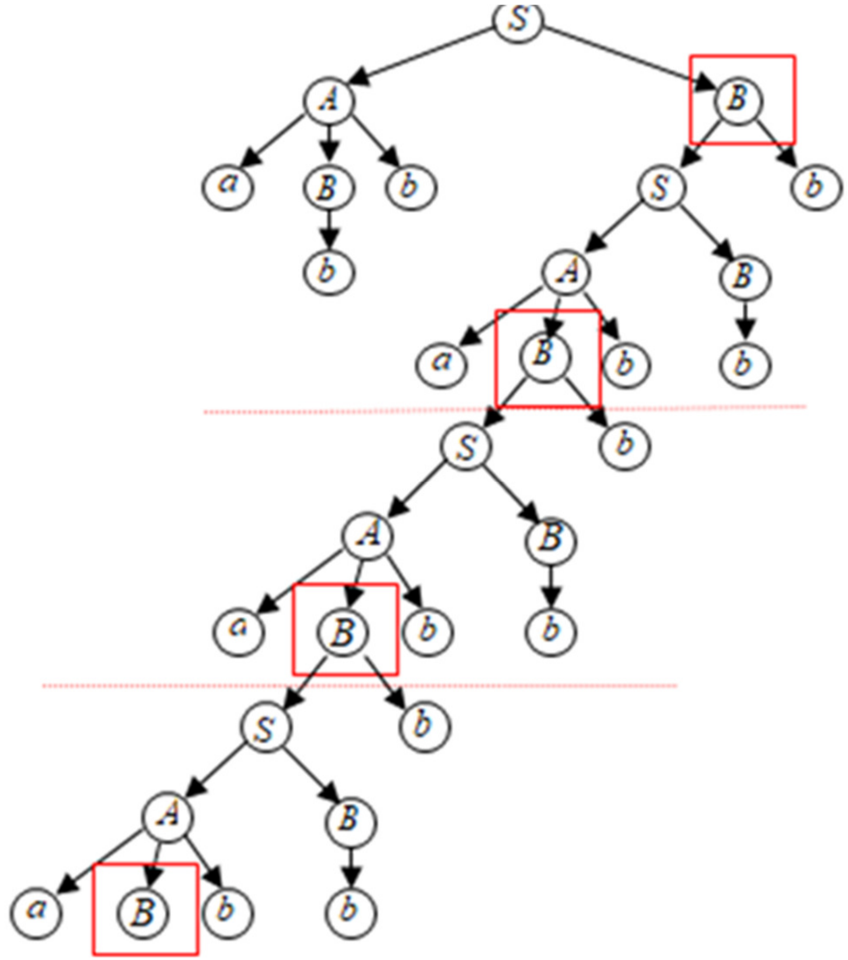


$$S \overset{*}{\Rightarrow} abb(a)^2 B(bbb)^2 \overset{*}{\Rightarrow} abb(a)^3 B(bbb)^3$$

A derivation from S

$$B \overset{*}{\Rightarrow} aBbbb$$

$$B \Rightarrow b$$



$$S \overset{*}{\Rightarrow} abb(a)^3 B(bbb)^3 \Rightarrow abb(a)^3 b(bbb)^3$$

$$abb(a)^3 b(bbb)^3 \in L(G)$$

In general

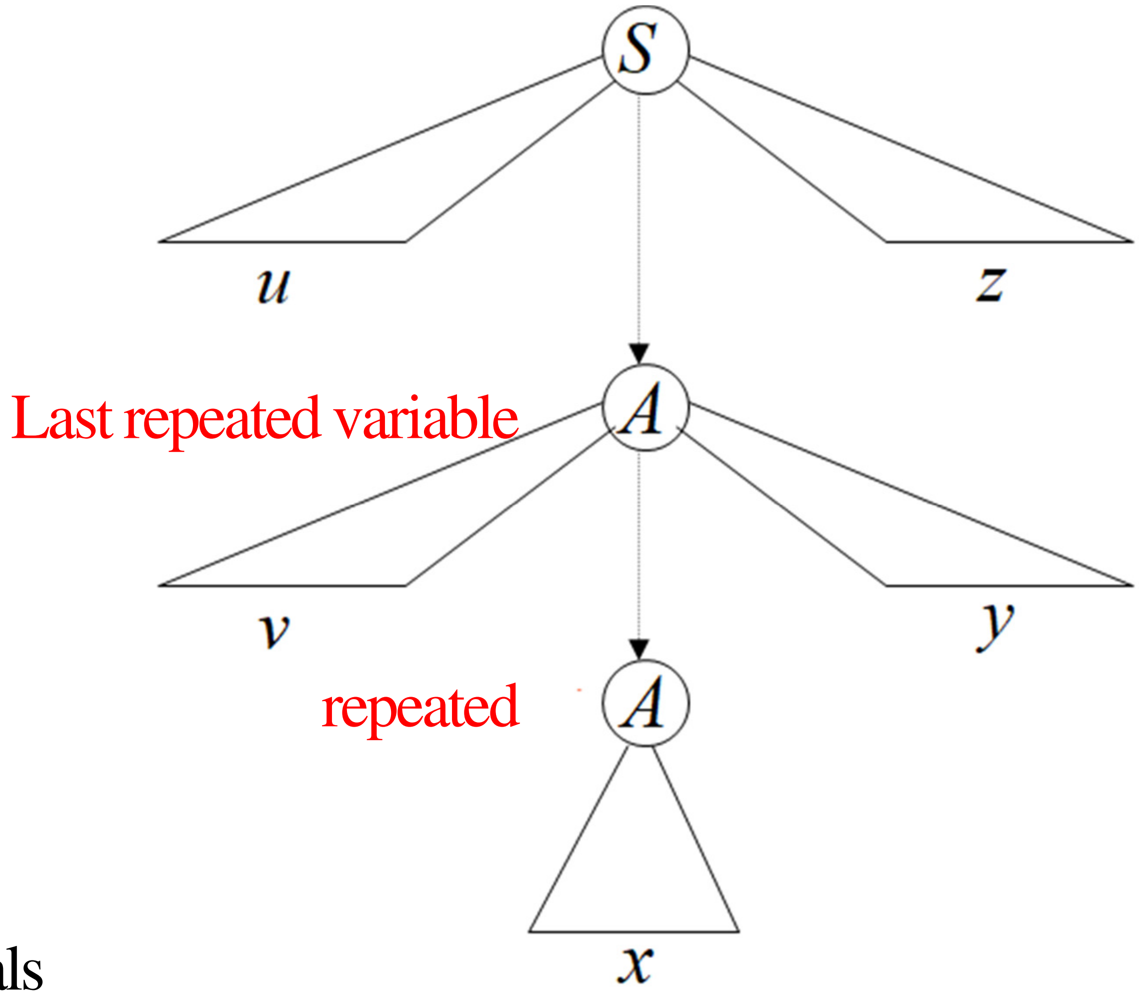
$$S \overset{*}{\Rightarrow} abb(a)^i b(bbb)^i$$

$$abb(a)^i b(bbb)^i \in L(G) \qquad i \geq 0$$

The Pumping Lemma for CFLs

- Consider an infinite context-free language L
- Let G be the grammar of $\mathrm{L} - \{\varepsilon\}$ and it has no unit-productions and ε-productions.
- Take a string $w \in L(G)$ with the length $|\mathrm{w}| \geq n$, where
  $n = p + 1$ and
  $p = \ (Number\ of\ productions) x\ (Largest\ right\ side\ of\ a\ production)$
- We will show in the derivation w of a variable of G is repeated.

Derivation tree of string *w*



$w = uvxyz$

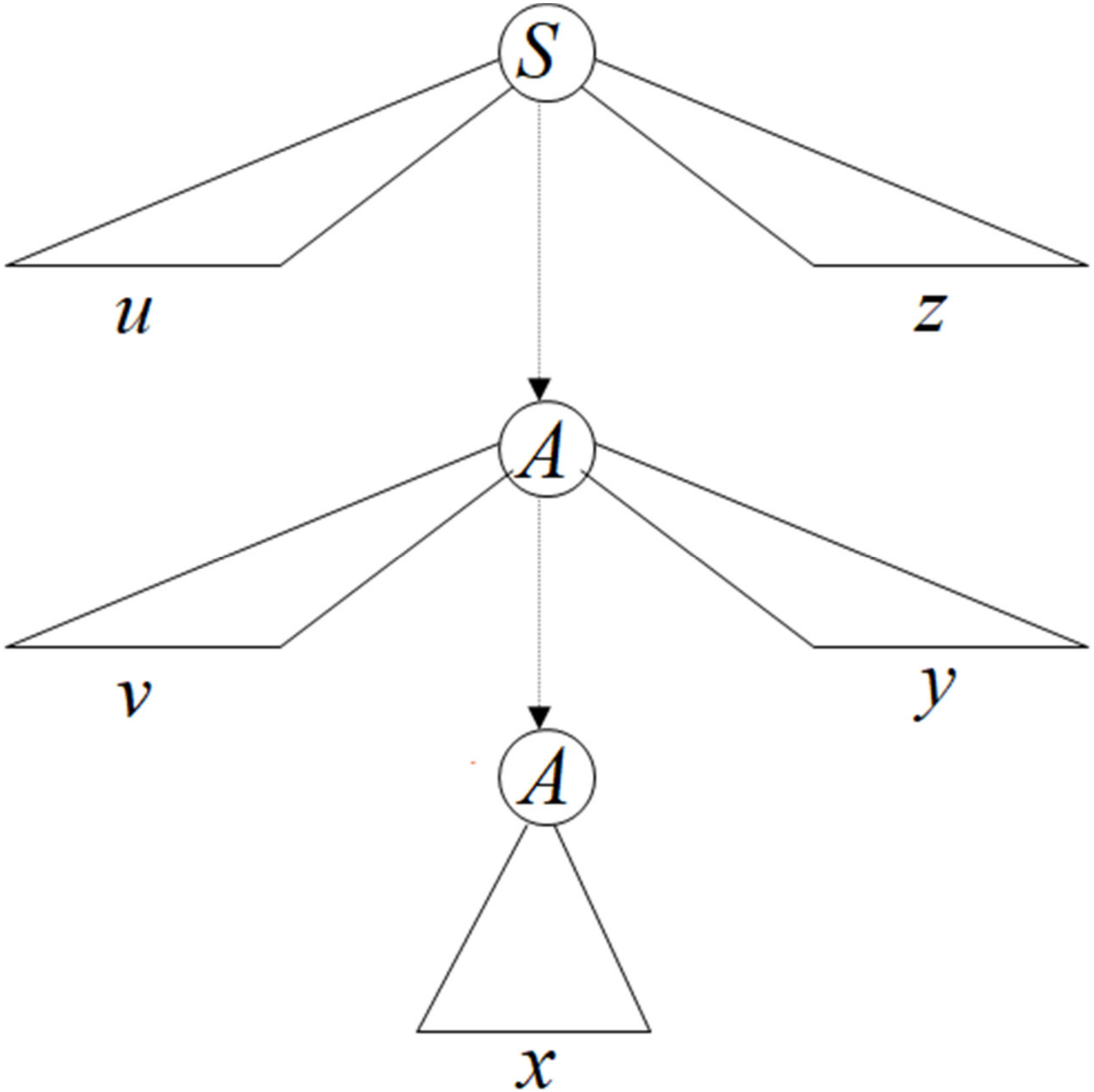
u, v, x, y, z are strings of terminals

Possible derivations:

$$S \overset{*}{\Rightarrow} uAz$$

$$A \overset{*}{\Rightarrow} vAy$$

$$A \overset{*}{\Rightarrow} x$$



The Pumping Lemma for CFLs

$$S \overset{*}{\Rightarrow} uAz \qquad A \overset{*}{\Rightarrow} vAy \qquad A \overset{*}{\Rightarrow} x$$

- These strings can be generated:

$$S \overset{*}{\Rightarrow} uAz \overset{*}{\Rightarrow} uxz \qquad uv^0xy^0z$$

$$S \overset{*}{\Rightarrow} uAz \overset{*}{\Rightarrow} uvAyz \overset{*}{\Rightarrow} uvxyz \qquad w = uv^1xy^1z$$

$$S \overset{*}{\Rightarrow} uAz \overset{*}{\Rightarrow} uvAyz \overset{*}{\Rightarrow} uvvAyyz \overset{*}{\Rightarrow} uvvxyyz \qquad uv^2xy^2z$$

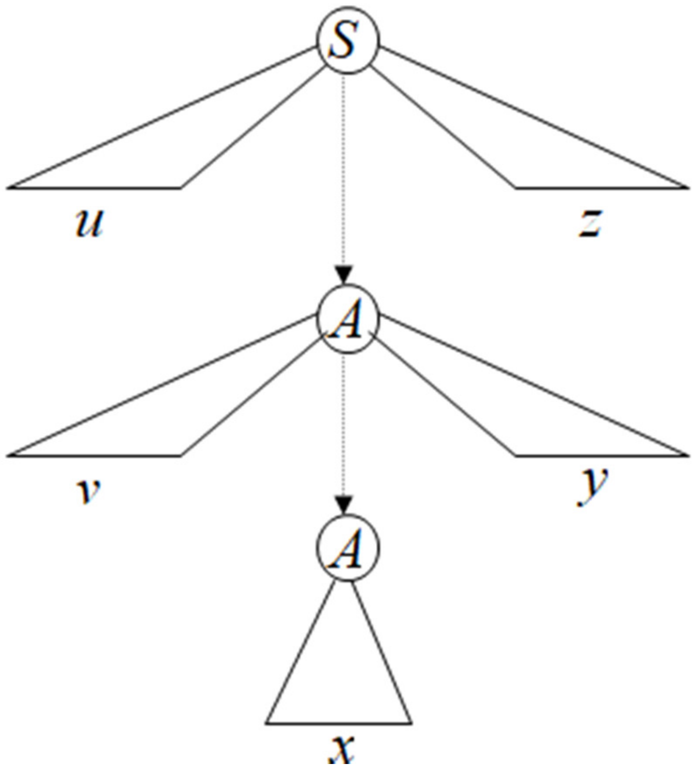
$$\vdots \qquad \vdots$$

$$uv^ixy^iz$$

The Pumping Lemma for CFLs

- Therefore, any string in the following form is generated by the grammar *G*

$$uv^i xy^i z \qquad i \geq 0$$



- We can observe that

$$|\,vxy\,| \leq n$$

- Since *A* is the last repeated variable

$$|\,vy\,| \geq 1$$

- Since there is no unit-productions and ε-productions

The Pumping Lemma for CFLs

Let L be a CFL, then there exists a constant n such that if w is any string in L such that $|w|$ is at least n, then we can write $w = uvxyz$, subject to the following conditions:

1. $|vxy| \leq n$
   - **That is, the middle portion is not too long.**
2. $|vy| \geq 1$
   - **Since v and y are the pieces to be "pumped," this condition says that at least one of the strings we pump must not be empty.**
3. uv^ixy^iz **is in L, for all $i \geq 0$.**

   That is, the two strings v and y may be "pumped" any number of times, including 0, and the resulting string will still be a member of L.

Using the Pumping Lemma

- **In order to show that a language *L* is NOT a CFL using the Pumping Lemma:**

1. **Suppose *L* were a CFL.**
2. **Then there is an integer n given us by the pumping lemma, which we do not know, we must plan for any possible n.**
3. **Pick a string w which must be in *L*, it must be defined using n and $|w| \geq n$.**
    - **Tricky Part 1: You should find a string w so that you can create a contradiction in step 5. YOU CANNOT SELECT A SPECIFIC STRING.**
4. **Break w into $uvxyz$, subject only to the constraints that $|vxy| \leq n$ and $|vy| \geq 1$**
5. **Pick i and show that uv^ixy^iz is NOT in *L* to create a contradiction.**
    - **Tricky Part 2: You have to show that uv^ixy^iz is NOT in *L* using only the constraints that $|vxy| \leq n$ and $|vy| \geq 1$. You may need to look at more than one cases. YOU CANNOT GIVE A SPECIFIC EXAMPLE.**
6. **Conclude that *L* is NOT a CFL.**

Using the Pumping Lemma – Example

Example: Let L be the language $\{0^k 1^k 2^k \mid k \geq 1\}$. Show that this language is NOT a CFL using the Pumping Lemma:

1. Suppose L were a CFL.
2. Then there is an integer n given us by the pumping lemma.
3. Let us pick a string $w = 0^n 1^n 2^n$ and $0^n 1^n 2^n$ is in L.
4. Break w into $uvxyz$, where $|vxy| \leq n$ and $vy \neq \varepsilon$.
5. Pick 0 for i and we have show that uxz is NOT in L to create a contradiction.
   - Since $|vxy| \leq n$, we know that vxy cannot involve both 0's and 2's, since the last 0 and the first 2 are separated by $n + 1$ positions. So, there are two cases.

Using the Pumping Lemma – Example

- Case 1: vxy has no 2's:
  - Then vy consists of only 0's and 1's.
  - Since v or/and y has at least one of these symbols.
  - Pick 0 for i, $\boldsymbol{uv^ixy^iz}$ is equal to $\boldsymbol{uxz}$
  - Then w has n 2's but it has fewer than n 0's or fewer than n 1's or both.
  - Therefore, vxy does not belong to L and creates a contradiction with our assumption that L were a CFL.
  - We conclude that L is NOT a CFL in case 1.
- Case 2: vxy has no 0's:
  - Then vy consists of only 1's and 2's.
  - Since, v or/and y has at least one of these symbols.
  - Then w has n 0's but it has fewer than n 1's or fewer than n 2's or both.
  - Therefore, vxy does not belong to L and creates a contradiction with our assumption that L were a CFL.
  - We also conclude that L is NOT a CFL in case 2.
- Whichever case holds, we conclude that L has a string we know NOT to be in L.

6. This contradiction allows us to conclude that our assumption was wrong; and L is not a CFL.

Using the Pumping Lemma – Exercise

Exercise: **Let L be the language $\{ww \mid w \in \{0, 1\}^*\}$. Show that this language is NOT a CFL using the Pumping Lemma.**