BLM2502 Theory of Computation

Spring 2016

BLM2502 Theory of Computation

Course Outline

Week	Content
1	Introduction to Course
2	Computability Theory, Complexity Theory, Automata Theory, Set Theory, Relations, Proofs, Pigeonhole Principle
3	Regular Expressions
4	Finite Automata
5	Deterministic and Nondeterministic Finite Automata
6	Epsilon Transition, Equivalence of Automata
7	Pumping Theorem
8	April 10 - 14 week is the first midterm week
9	Context Free Grammars
10	Parse Tree, Ambiguity,
11	**Pumping Theorem**
12	Turing Machines, Recognition and Computation, Church-Turing Hypothesis
13	Turing Machines, Recognition and Computation, Church-Turing Hypothesis
14	May 22 – 27 week is the second midterm week
15	Review
16	Final Exam date will be announced

The Pumping Lemma for CFL's

Pumping Lemma

- Recall the pumping lemma for regular languages.
- It told 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 and discover an infinite sequence of strings that had to be in the language.
- For CFL's the situation is a little more complicated.
- We can always find two pieces of any sufficiently long string to "pump" in tandem.
  - That is: if we repeat each of the two pieces the same number of times, we get another string in the language.

Stringin 2 kısmını istedigimiz kadar pump edelim accept etmek zorunda.

Statement of the CFL Pumping Lemma

$|z| > n$

$z \in L$

- For every context-free language L There is an integer n, such that For every string z in L of length > n There exists z = uvwxy such that:

1. |vwx| < n.
2. |vx| > 0.
3. For all i > 0, uv^iwx^iy is in L.

v ile stacke koydun

y ile stackten boşaltmalısın

Proof of the Pumping Lemma

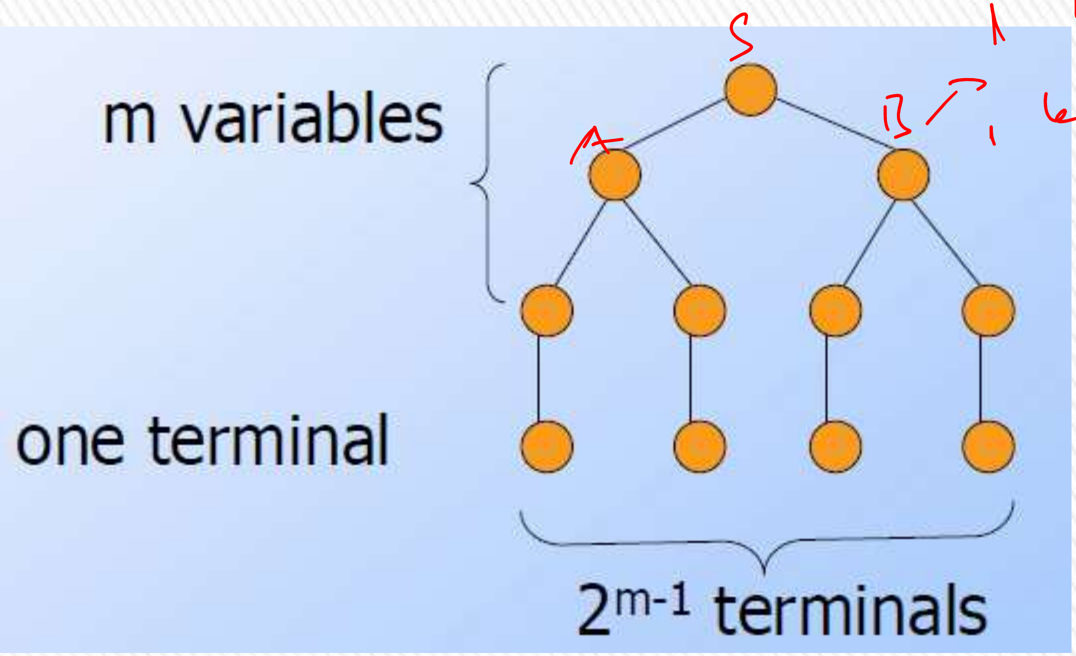
- Start with a CNF grammar for L - {ε}.
- Let the grammar have m variables.
  - Pick n = 2^m.
  - Let |z| > n.
- We claim ("Lemma 1") that a parse tree with yield z must have a path of length m+2 or more.

→ grammarda m tane variable olsun S, A, B

Proof of Lemma 1

S → AB
A → AB
A → a
B → b

- If all paths in the parse tree of a CNF grammar are of length < m+1, then the longest yield has length 2^{m-1}, as in figure:



1 variable kullanılmış?

B → 1 kere

Proof of the Pumping Lemma

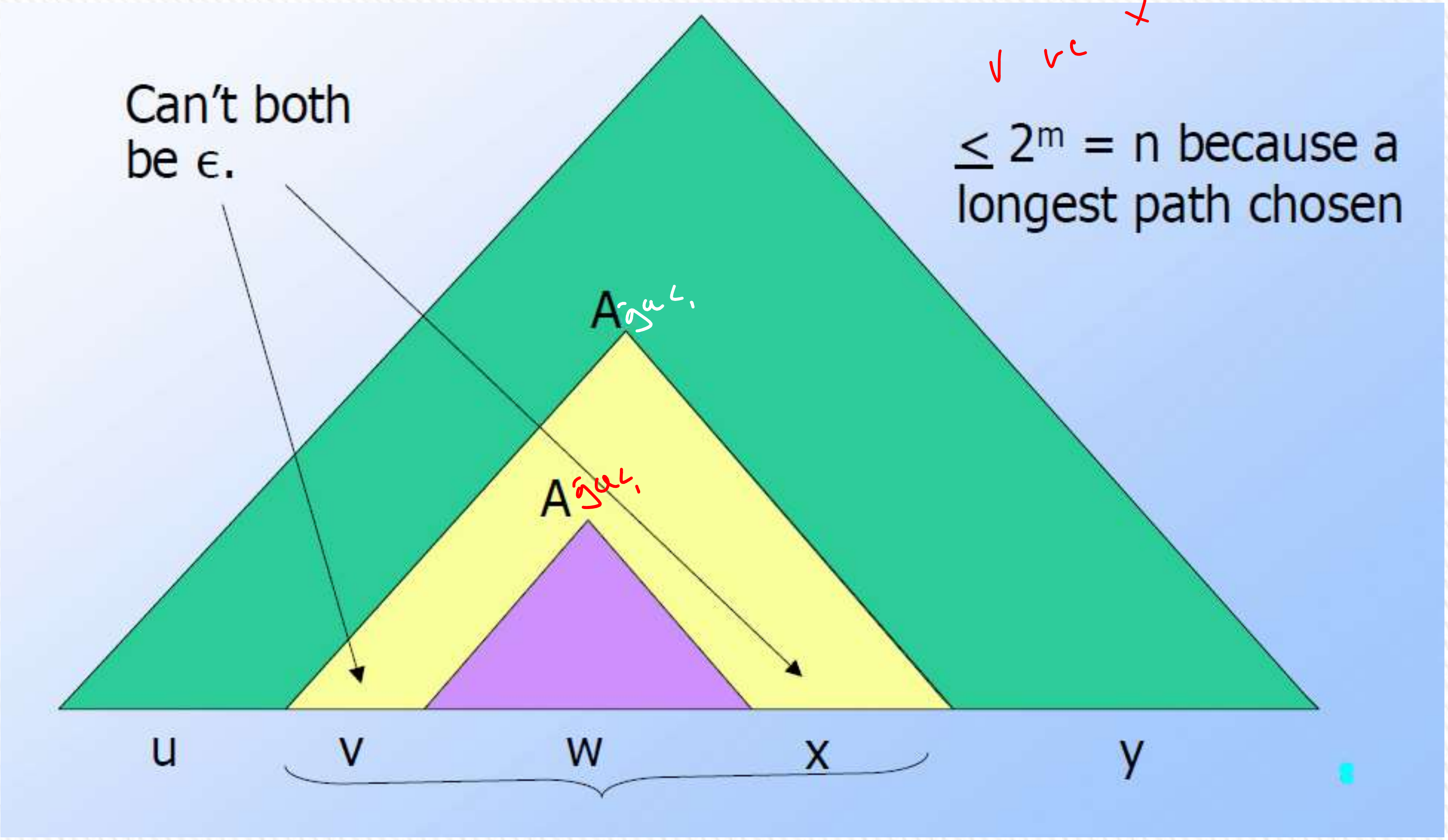
- Now we know that the parse tree for z has a path with at least m+1 variables.

$|z| > 2^m$ olduğu için m+1 değişken olmalı!

- Consider some longest path.
- There are only m different variables, so among the lowest m+1 we can find two nodes with the same label, say A.
- The parse tree thus looks like:

Parse Tree in the Pumping-Lemma

v ve x ε olamaz

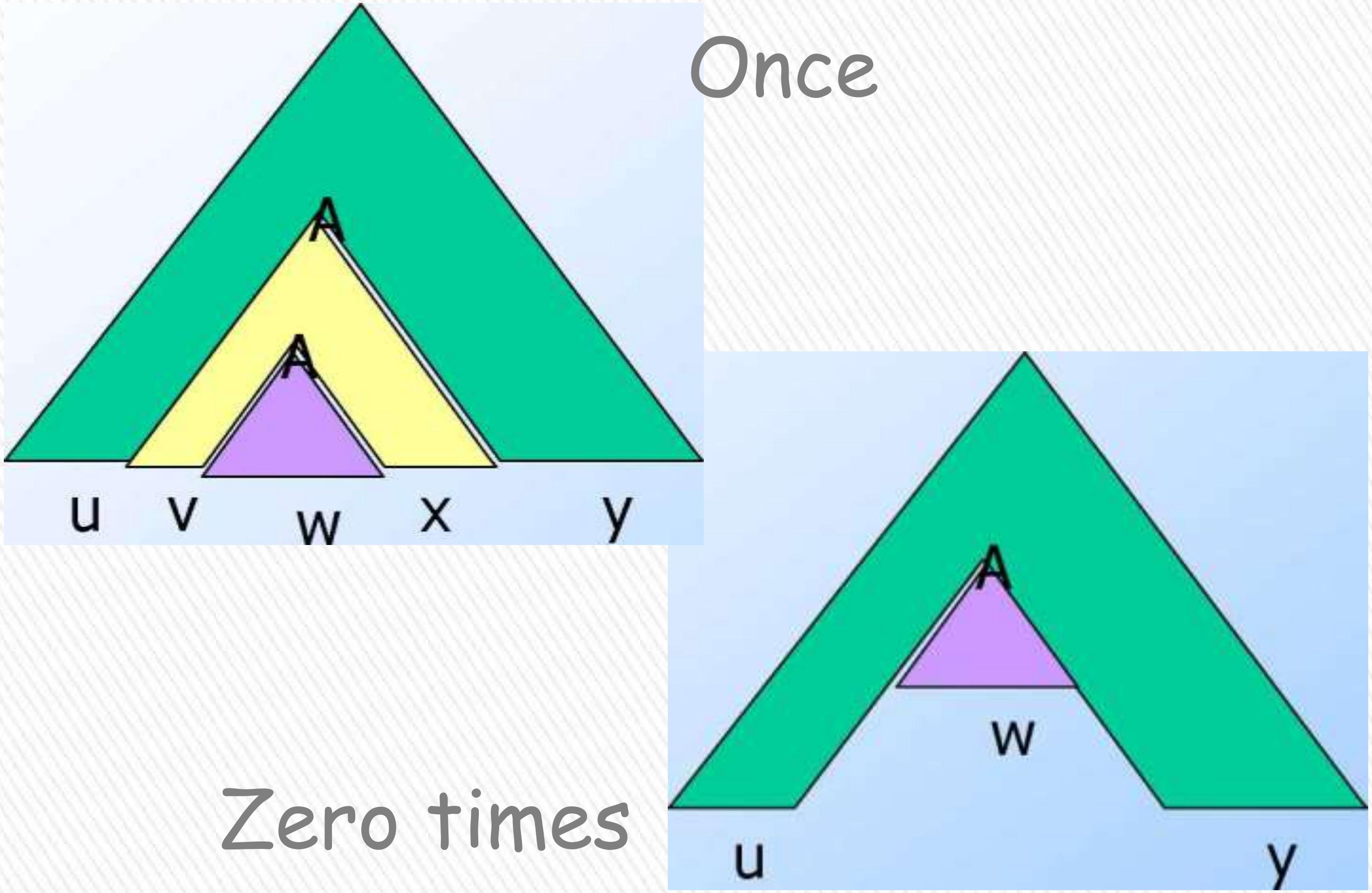


Can't both be ε.

≤ $2^m = n$ because a longest path chosen

Pumping

→ L CFL'e aitse



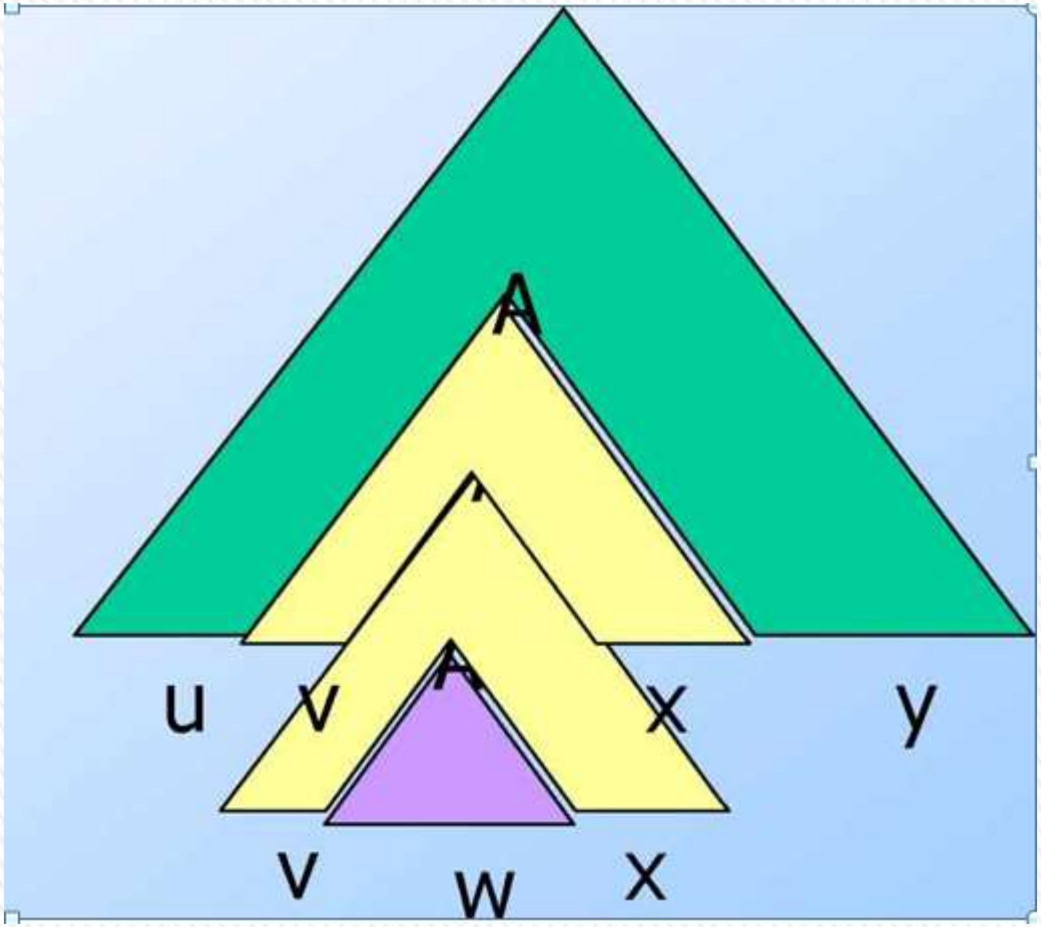
Once

Zero times

12 dn aittir

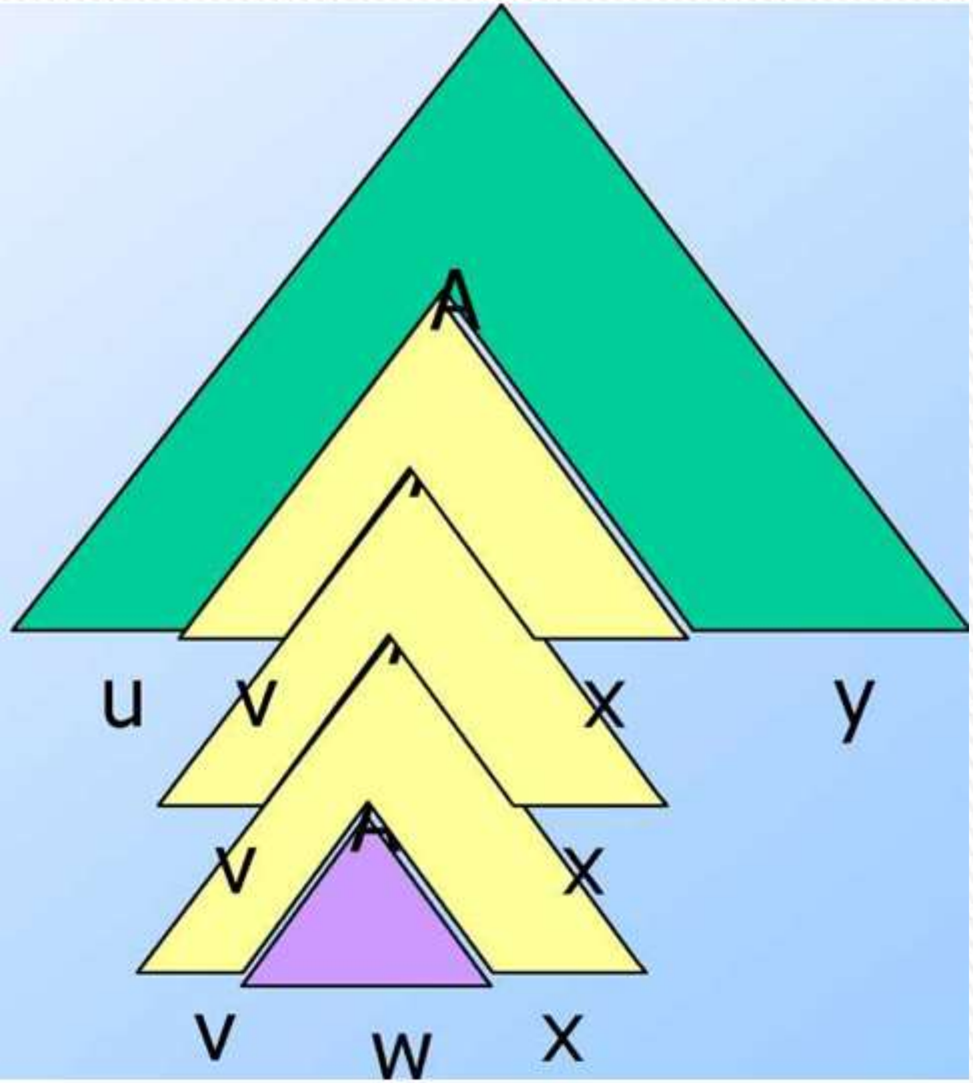
Pumping

Twice



Pumping

Thrice, …



Using the Pumping Lemma

- Non-CFL's typically involve trying to match two pairs of counts or match two strings.
- Example: pumping lemma can be used to show that L = {ww | w in (0+1)*} is not a CFL.
- {0'10' | i >1} is a CFL.
  - We can match one pair of counts.

kendini gösteriyor ondan dolayı CFL değil push pop

$u\,\boxed{v}\,w\,\boxed{x}\,y$

$uv^n w x^n y$

$ab^n c b^n \cancel{b}$

$\underline{a\,b^{n+k}\,c\,b^{n+k}\,b}$ →

Using the Pumping Lemma

- $L = \{0^i10^i10^i \mid i > 1\}$ is not a CFL
  - We can't match two pairs, or three counts as a group.
  - Proof using the pumping lemma.
  - Suppose L were a CFL.
  - Let n be L's pumping-lemma constant.
  - Consider $z = 0^n10^n10^n$.
  - We can write $z = uvwxy$, where $|vwx| < n$, and $|vx| > 1$.
  - Case 1: vx has no 0's.
  - Then at least one of them is a 1, and uwy has at most one 1, which no string in L does.

Using the Pumping Lemma

- Still considering z = $0^n10^n10^n$.
- Case 2: vx has at least one 0.
- vwx is too short (length < n) to extend to all three blocks of 0's in $0^n10^n10^n$.
- Thus, uwy has at least one block of n 0's, and at least one block with fewer than n 0's.
- Thus, uwy is not in L.

$u\ \textcircled{v}\ w\ \textcircled{x}\ y$