CS5371
Theory of Computation

Lecture 9: Automata Theory VII
(Pumping Lemma, Non-CFL, DPDA $\neq$ PDA)

Objectives

- Introduce the Pumping Lemma for CFL
- Show that some languages are non-CFL
- Discuss the DPDA, which is a PDA with "deterministic transition function"

Pumping Lemma for CFL

Theorem: If L is a context-free language, then there is a number p (pumping length) where, if s is any string in L of length at least p, we can find u,v,x,y,z such that s = uvxyz and

- For each $i \geq 0$, uv^ixy^iz is in L
- $|vy| > 0$, and
- $|vxy| \leq p$

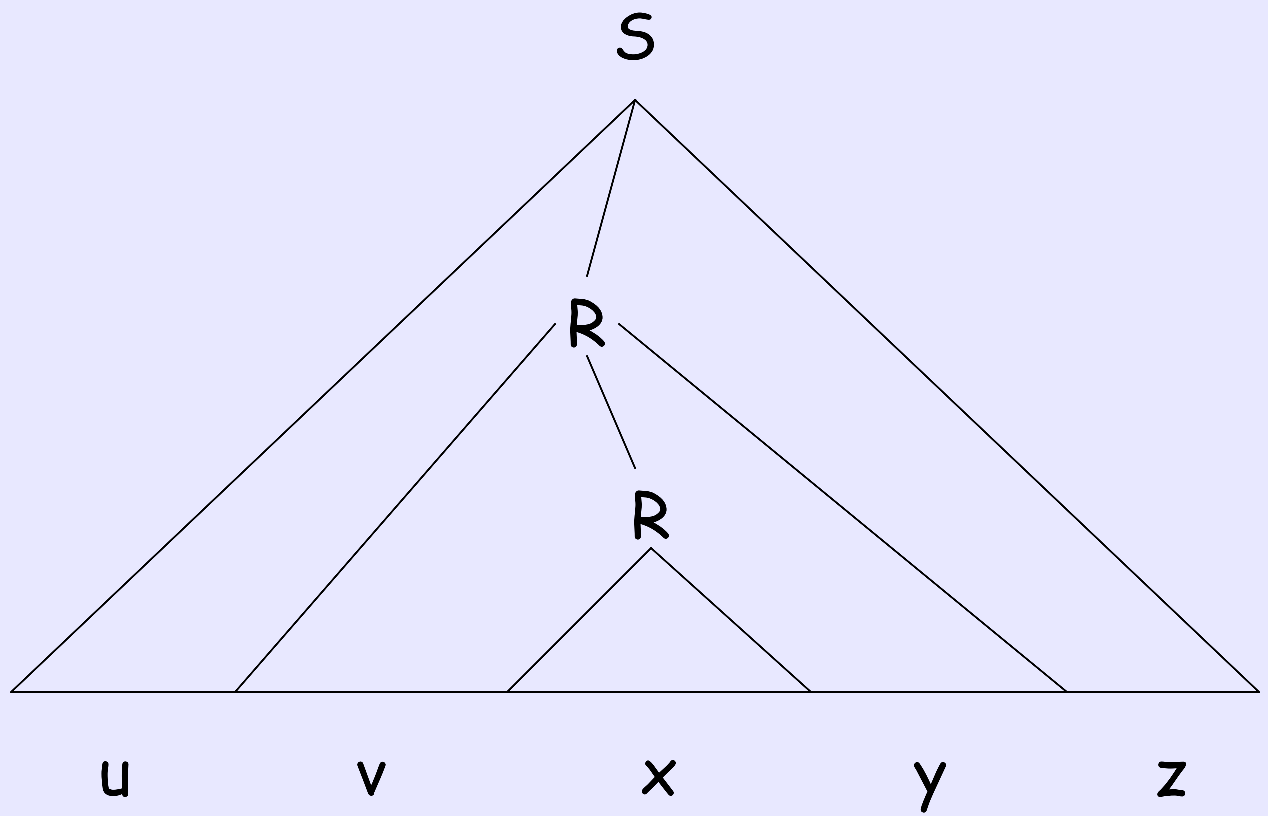
Proof of Pumping Lemma

- Let b be the maximum branching factor in the parse tree of any string in L (that is, the right side of any rule has at most b terminals and variables)
- Let $p = b^{|V|+1}$
- What is the minimum height of the parse tree for a string longer than p?

Proof of Pumping Lemma (2)

- The height of the tree is at least |V| + 1 → There is a path in tree with length |V|+2 nodes
  - Only one can be terminal, so that there are |V|+1 variable on the path
- What does that mean?
  - Some variable appear at least two times!

Proof of Pumping Lemma (3)



Proof of Pumping Lemma (4)

- At this point, we note that uv^ixy^iz is in L for any $i \geq 0$
  - From the parse tree, we know that R can derive x, and R can derive vxy
  - As S derives uRz, S can derive uxz
  - As S derives uvRyz, S can derive uvvxyyz
- To complete the prove, we need to show
  - $|vy| > 0$, and
  - $|vxy| \leq p$
- The current construction cannot, but…

Proof of Pumping Lemma (5)

- If we let the above parse tree to be the smallest among all that generates the string, and
- If we use the longest path from S to any leaf, and R be the repeating variable from the lowest |V|+1 in this path
- Then,
  - $|vy| > 0$ (why?) and
  - $|vxy| \leq p$ (why?)

Why is |vy| > 0?

- Suppose on the contrary that $|vy| = 0$. This occurs when both v and y are empty strings
- If this happen, we can replace the subtree rooted at R that generates vxy by the subtree rooted at R that generates x
  - Resulting parse tree also geneartes uvxyz, but it has fewer nodes → contradiction occurs (why?)

Why is $|vxy| \leq p$?

- R is chosen from the lowest $|V| + 1$ variable
- The height of the subtree at R that generates vxy has at most $|V|+1$ (why?)
  - It has at most $b^{|V|+1}$ leaves
  - Thus, it can generate at most p characters (as $p = b^{|V|+1}$)

Recall: b = maximum branching factor

Non-CFL (example 1)

Theorem: The language

$$A = \{a^n b^n c^n \mid n \geq 0\}$$

is not a context-free language.

How to prove?

Use pumping lemma on $a^p b^p c^p$

Proof (example 1)

- We apply pumping lemma on $a^p b^p c^p$, what can be the corresponding vxy?
  - Case 1: if both v and y contain only one type of char
  - Case 2: if either v or y contain more than one type of char
- In both cases, uvvxyyz cannot be in *A* (why?)
- Thus, a string longer than p in *A* does not satisfy pumping lemma → *A* is not CFL

Non-CFL (example 2)

Theorem: The language

$$B = \{a^i b^j c^k \mid 0 \leq i \leq j \leq k\}$$

is not a context-free language.

How to prove?

Use pumping lemma on $a^p b^p c^p$

Proof (example 2)

- We apply pumping lemma on $a^p b^p c^p$, what can be the corresponding vxy?
  - Case 1: if both v and y contain only one type of char
  - Case 2: if either v or y contain more than one type of char
- We can guarantee that for Case 2, uvvxyyz cannot be in B
- However, for Case 1, if v = b, y = c, then the string uvvxyyz can always be in B… (so, how to get a contradiction??)

Proof (example 2)

- We can divide Case 1 into two subcases:
  - When char 'a' does not appear in v or y
  - When char 'a' appears in v or y
- For the first subcase, uxz (here, we pump down the string) cannot be in B (why?)
- For the second subcase, uvvxyyz (here, we pump up the string) cannot be in B (why?)
- Thus, a string longer than p in B does not satisfy pumping lemma → B is not CFL

Non-CFL (example 3)

Theorem: The language

$$C = \{ww \mid w \text{ in } \{0,1\}^*\}$$

is not a context-free language.

How to prove?

Use pumping lemma on $0^p1^p0^p1^p$

Proof (example 3)

- We apply pumping lemma on $0^p1^p0^p1^p$, what can be the corresponding vxy?
  - Case 1: vxy appears in the first half
  - Case 2: vxy appears in the second half
  - Case 3: vxy includes the middle '10'
- For Cases 1 or 2, uvvxyyz not in C (why?)
- For Case 3, u must start with 0^p, and z must end with 1^p (because $|vxy| \leq p$ and vxy includes the middle '10'). Then, uxz cannot be in C (why?)

Is CFL closed under complement?

- What is the complement of *A* = $\{a^n b^n c^n \mid n \geq 0\}$?

Assume $\Sigma = \{a, b, c\}$

- They include:
  - strings containing ba, ca, or cb;
  - strings $a^i b^j c^k$ with $i \neq j$ or $j \neq k$
- Thus, the complement of *A* is context free. (why??)
- As *A* is not context free, what can we conclude?

Is CFL closed under intersection?

- Is $A = \{a^n b^n c^m \mid n,m \geq 0\}$ a CFL?
- Is $B = \{a^m b^n c^n \mid n,m \geq 0\}$ a CFL?
- What is the intersection of A and B? Is it a CFL?
- What can we conclude?

DPDA

- Roughly speaking, a deterministic PDA is a PDA with only one choice at any time
- Precisely, for current state q, input char a, and stack symbol s,

  (1) $|\delta(q, a, s)| = 1$ for all $a \in \Sigma$ and

  $|\delta(q, \varepsilon, s)| = 0$, or

  (2) $|\delta(q, a, s)| = 0$ for all $a \in \Sigma$ and

  $|\delta(q, \varepsilon, s)| = 1$
- Then, at the end of reading the input, the PDA can be in at most 1 state

DPDA (2)

- We also need to assume that all input strings for DPDA ends with a special char # (which does not appear at other places in the input string)
  - Since, unlike PDA, we cannot guess the end of the input string

DPDA = PDA??

- Suppose L is a language recognized by a DPDA D. Will the complement of L be recognized by some DPDA D'?
- Yes (… rough idea only…)
  - Consider all the states that correspond to the end of reading the input strings (which has an incoming transition arrow labeling with (#, b → c)). If it is accept/reject state in D, reverse it in D'

DPDA = PDA??

- As language recognized by DPDA is closed under complement, but language recognized by PDA (that is, CFL) is not closed, we have

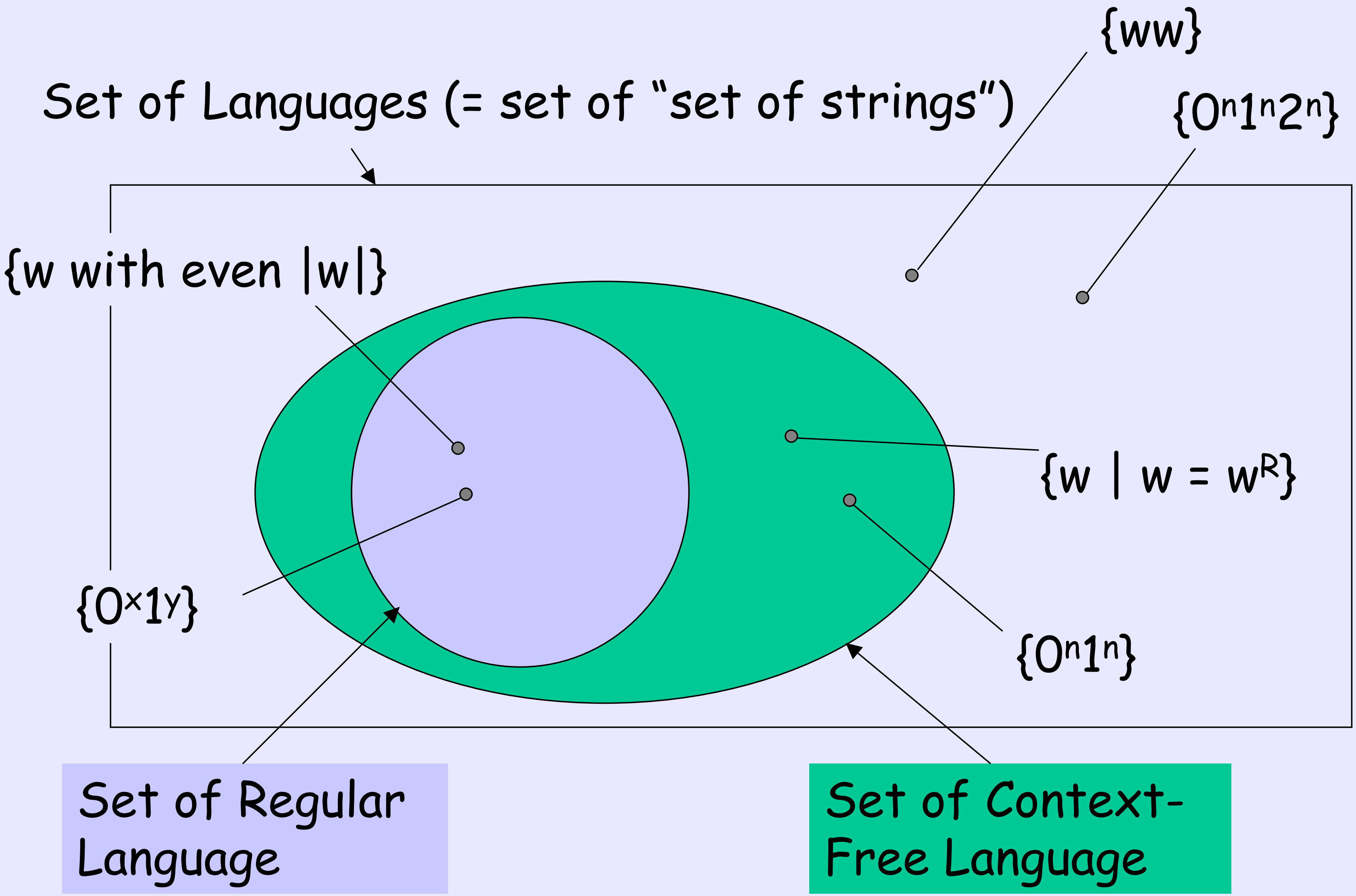
$$\text{DPDA} \neq \text{PDA}$$

in terms of descriptive power.

What we have learnt so far?

- PDA = CFG
  - Prove by Construction
- Properties of CFG (Ambiguous, CNF)
  - More on these in Homework 2
- Pumping Lemma
  - Prove by Construction (using Parse Tree)
- Existence of non-CFL
- DPDA $\neq$ PDA

Language Hierarchy



Next Time

- Turing Machine
  - A even more power computer