PART 2:

Automata:

PDA

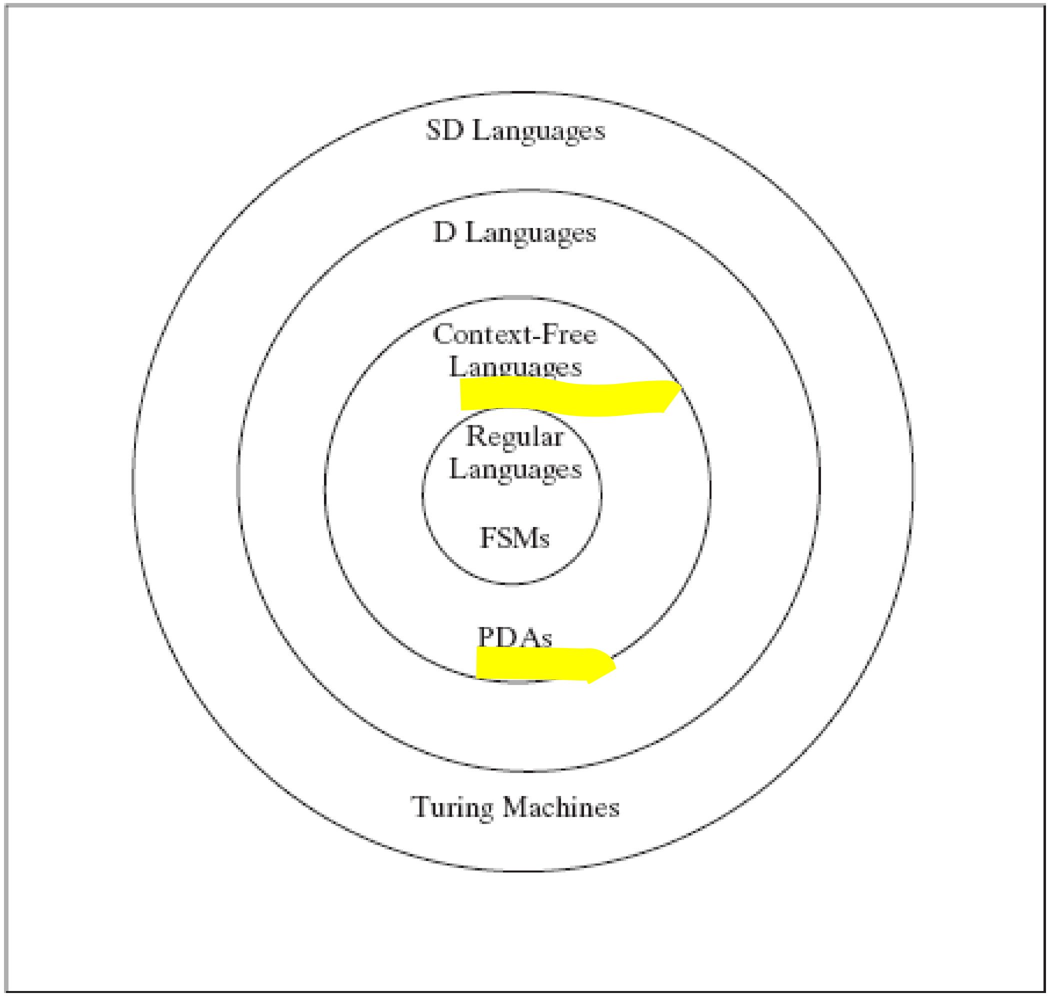
Formal Language:

Context-Free Languages
Non-Context-Free Languages

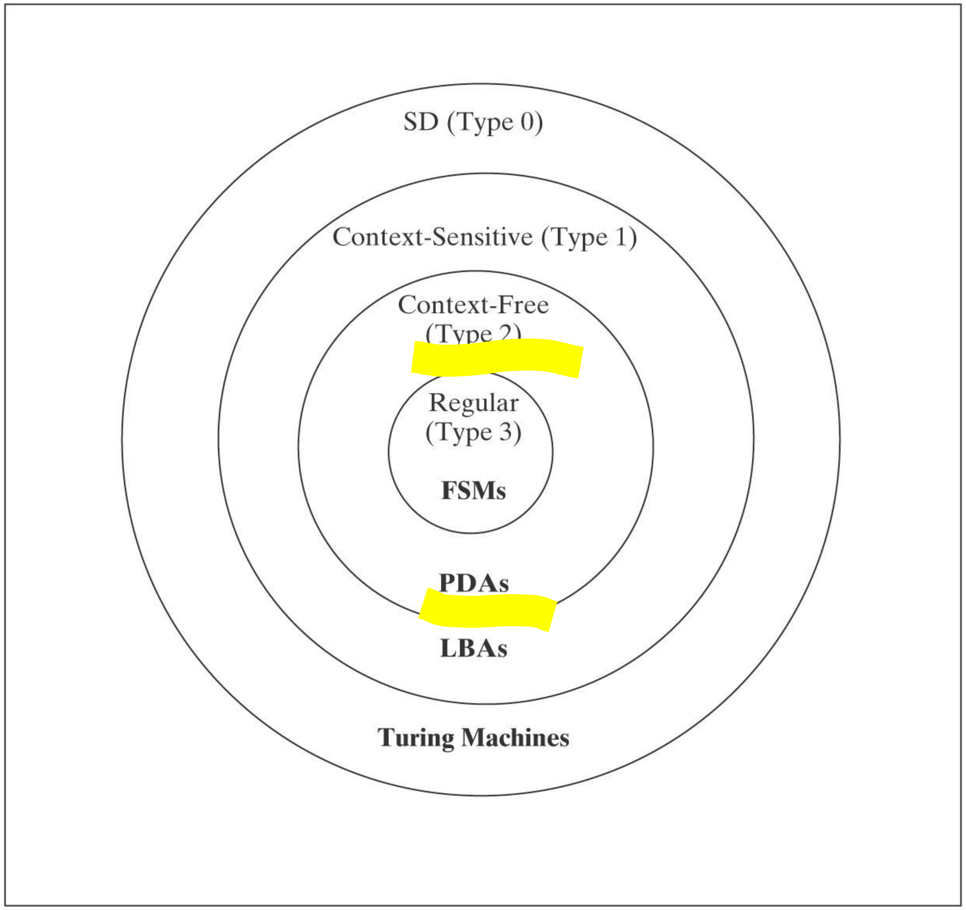
Grammar:

Context-Free Grammars

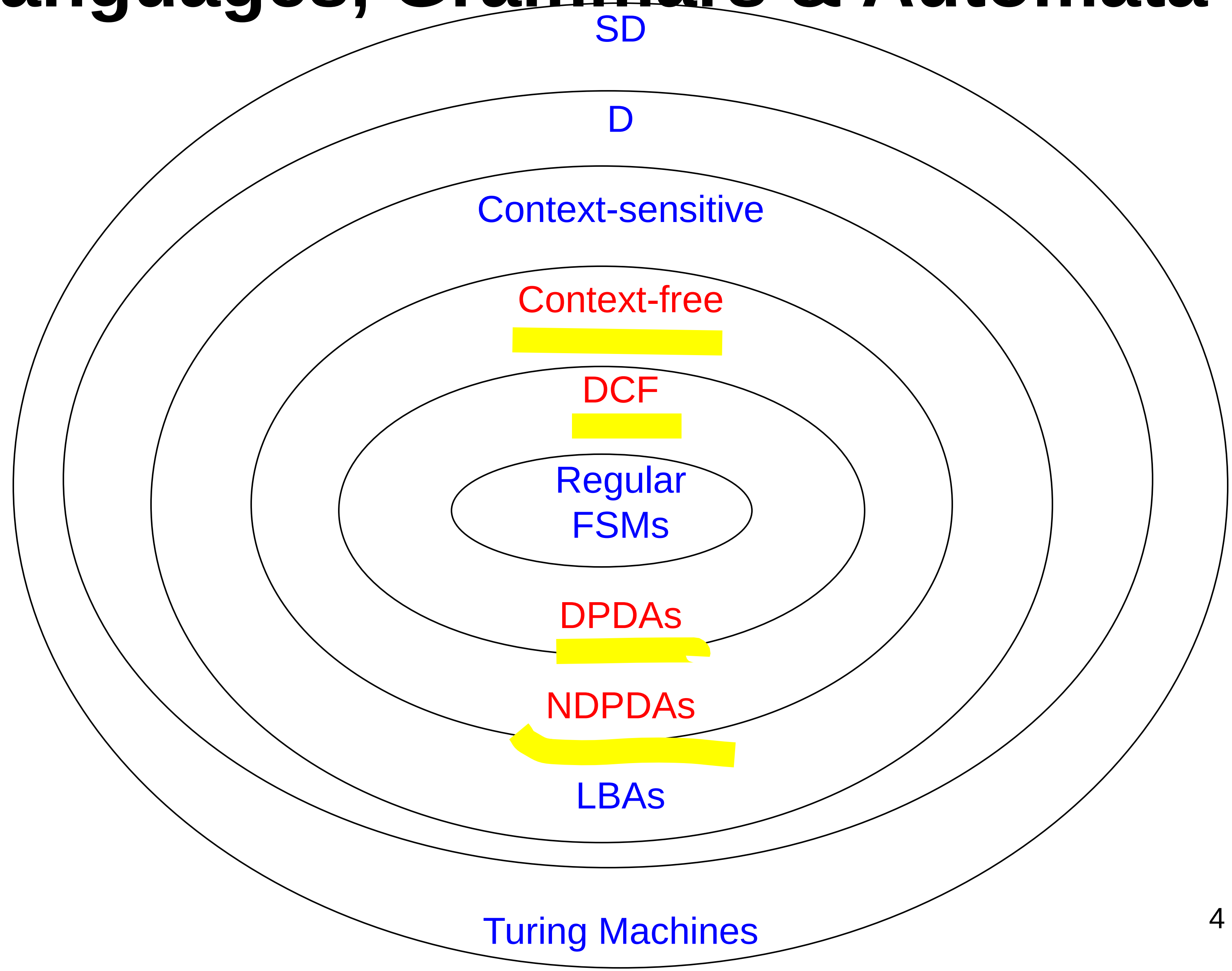
Languages, Grammars & Automata



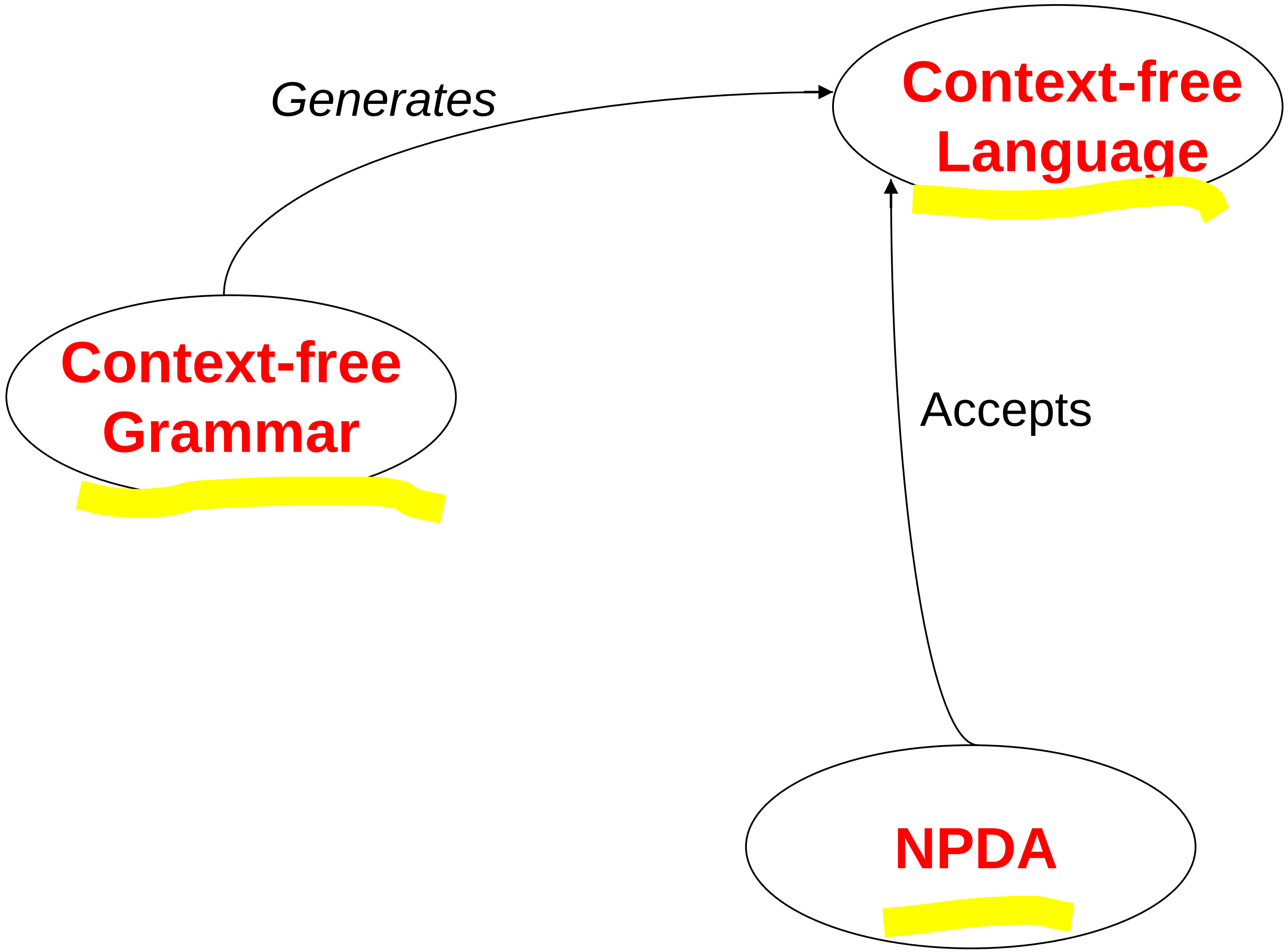
Languages, Grammars & Automata



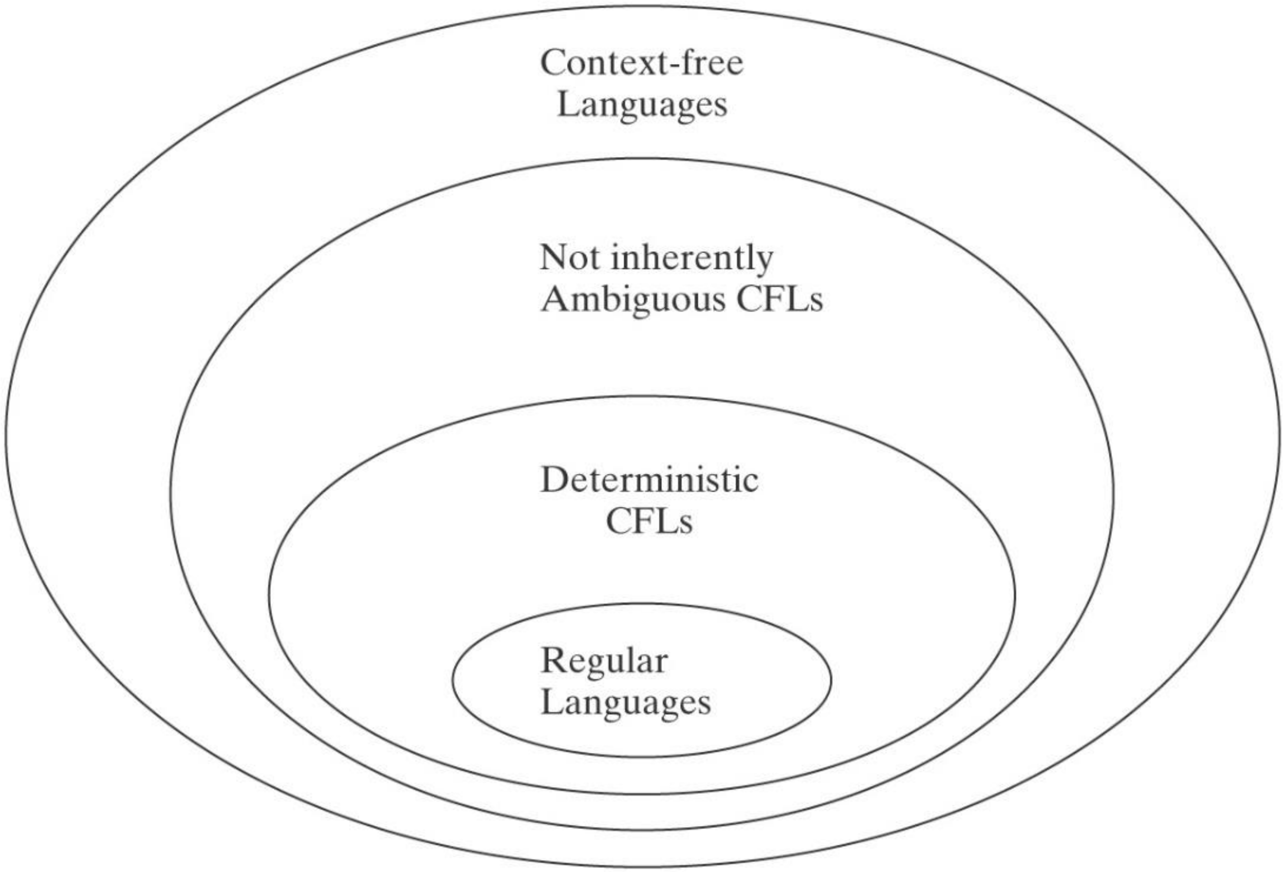
Languages, Grammars & Automata



Context-free Grammars, Languages, and PDAs



The CFL Hierarchy



Closure Properties & Pumping of CFLs

Non-Context-Free Languages

Languages That Are and Are Not Context-Free?

- $\mathtt{a}^*\mathtt{b}^*$ context-free?
- $A^nB^n = \{\mathtt{a}^n\mathtt{b}^n : n \geq 0\}$ context-free?
- $A^nB^nC^n = \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n : n \geq 0\}$ context-free?
- PalEven = $\{ww^R : w \in \{\mathtt{a}, \mathtt{b}\}^*\}$ context-free?
- WW = $\{ww : w \in \{\mathtt{a}, \mathtt{b}\}^*\}$ context-free?

Every RL is CFL

Lemma: Every regular language is CF.

Proof Idea:

Every FSM is (trivially) a PDA:

Given an FSM $M = (K, \Sigma, \Delta, s, A)$ and elements of δ of the form:

(p,	c,	q)
old state,	input,	new state

Construct a PDA $M' = (K, \Sigma, \{\varnothing\}, \Delta, s, A)$. Each (p, c, q) becomes:

((p,	c,	ε),	(q,	ε))
old state,	input,	don't look at stack	new state	don't push on stack

Just don't use the stack!

There Exists at Least One Language that is CF but Not Regular

Lemma: There exists at least one language that is CF but not regular

Proof Idea:

Proof by Counterexample

L= $\{a^n b^n, n \geq 0\}$ is context-free but not regular.

CFLs Properly Contain RLs

Theorem 13.1: The regular languages are a <u>proper subset</u> of the context-free languages.

Proof Idea:

In two parts:

- Every regular language is CF.
- There exists at least one language that is CF but not regular.

How Many CFLs Are There?

Theorem 13.2: There is a countably infinite number of CFLs.

Proof Idea:

- We can lexicographically enumerate all the CFGs.
- There is a countably infinite number of CFGs.
- Thus, a countably infinite number of CFLs.

There Exist Non-Context-Free Languages

- There is a countably infinite number of CFLs.
- There is an uncountable number of languages.

✓ Thus there are more languages than there are context-free languages.

✓ So there must exist some languages that are not context-free.

Example: $\{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n : n \geq 0\}$

Showing that L is Context-Free

Showing that L is Context-Free

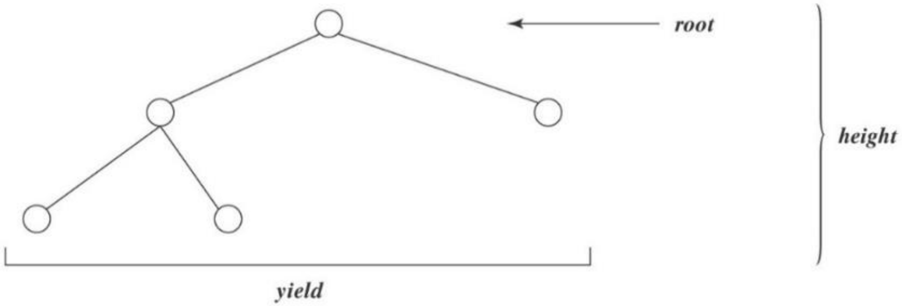
Techniques for showing that a language *L* is context-free:

- ✓ Exhibit a context-free grammar for *L*.
- ✓ Exhibit a PDA for *L*.
- ✓ Use the closure properties of context-free languages. (weaker than they are for regular languages.)

Showing that *L* is Not Context-Free

Parse Tree

- The *height* of a parse tree is *the length of the longest path from the root to any leaf*.
- The *branching factor* of a parse tree is *the largest number of daughter nodes* associated with any node in the tree.
- The *yield* of a parse tree is *the ordered sequence of its leaf nodes*.

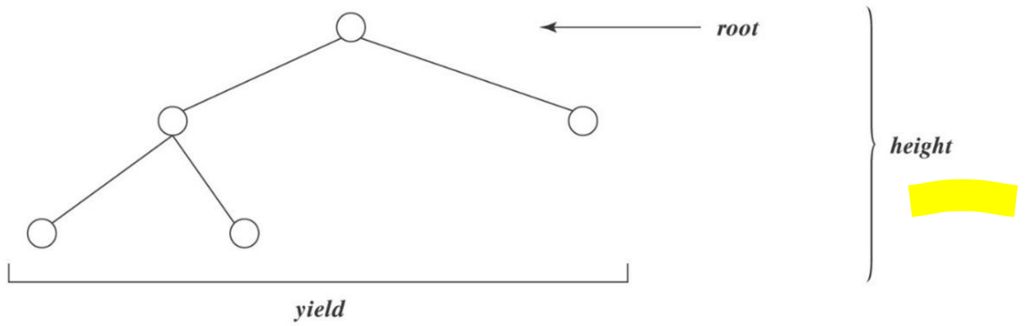


Height, Balancing Factor & Yield

Theorem 13.3: The **length of the yield** of *any parse tree T* with height ***h*** and branching factor ***b*** is ≤ $\boldsymbol{b^h}$.

Proof Idea:

Proof by Induction on h.



Height, Balancing Factor & Yield

Proof by Induction on h:

- Prove when h=1:

The length of the yield of *any parse tree T* with height *h=1* and branching factor *b* is

$\leq b$

$\leq b^1$

- Assume it is true for h=n (Inductive Hypothesis):

The length of the yield of any parse tree T with height h=n and branching factor b is $\leq b^h$ *.*

Height, Balancing Factor & Yield

- Prove when h=n+1 using the Inductive Hypothesis:

The length of the yield of *any parse tree T* with height $h=n+1$ and branching factor b is

$\leq (b^n)(b^1)$

$\leq b^{n+1}$

$\leq b^h$

Context-Free Pumping

CFG G = (V, Σ, R, S)

n = the # of nonterminals
b = the branching factor
= the length of the longest RHS of any rule

Consider any parse tree T where no nonterminal appears no more than once:

- **The height of T $\leq$ n**
- **The longest string (the yield of T) $\leq b^n$**

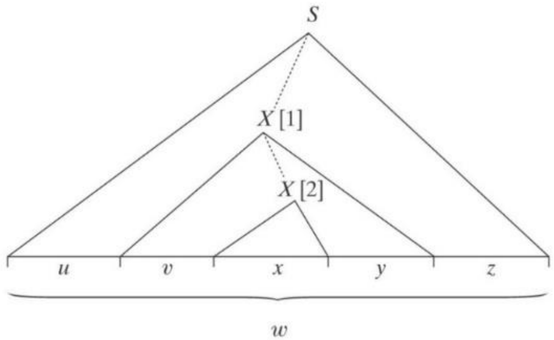
Context-Free Pumping

Consider a string w in L(G) st **$|w| > b^n$**

Any parse tree that G generates for w

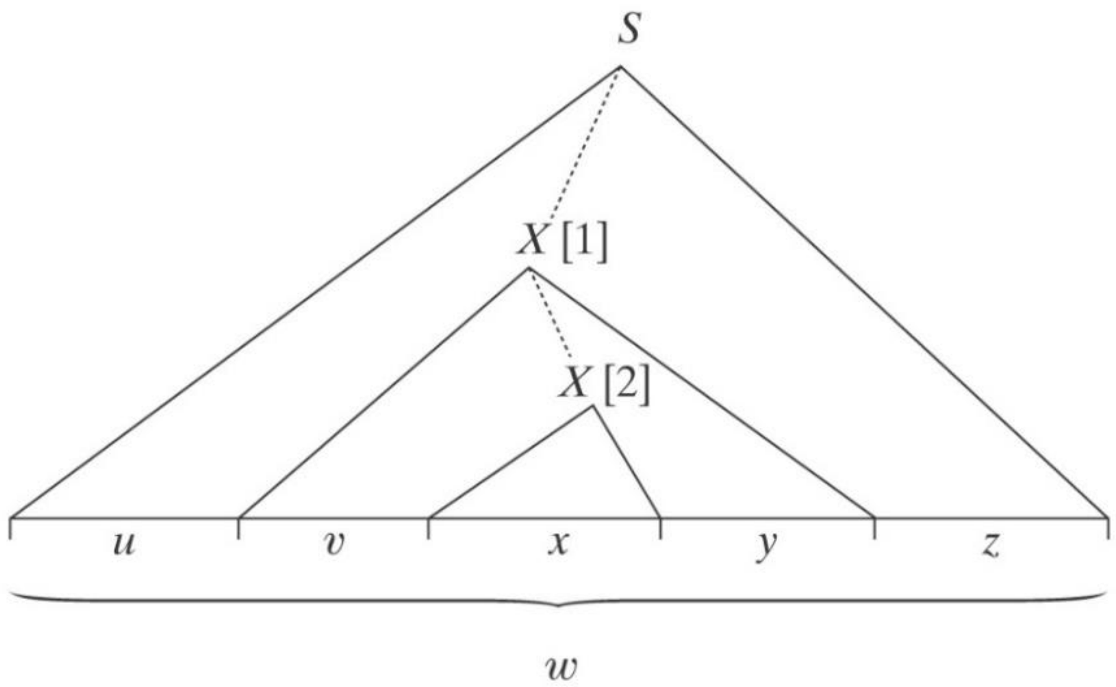
- must contain at least one path that contains at least one **repeated** nonterminal!
- must use at least one **recursive rule**!

The parse tree must look like:



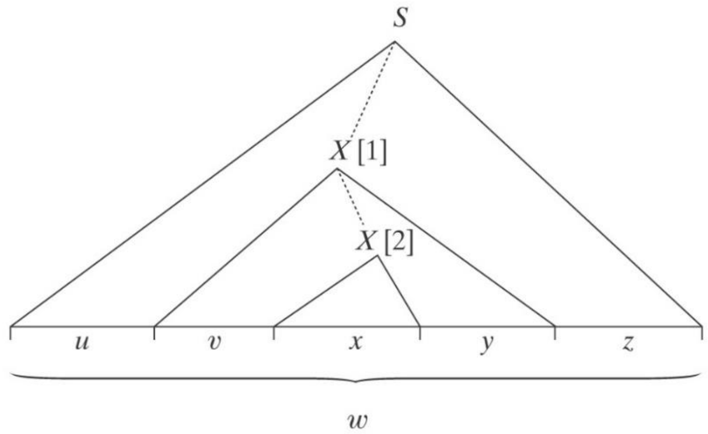
Context-Free Pumping

The parse tree must look like:



$S \Rightarrow^* uXz \Rightarrow^* uvXyz \Rightarrow^* uvvxyyz$

Context-Free Pumping



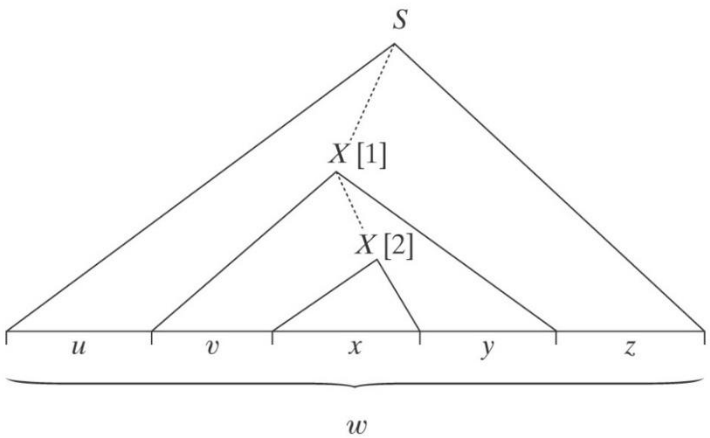
There is another derivation in *G*:

$S \Rightarrow^* uXz \Rightarrow^* uxz$,

At the point labeled [1], the non-recursive $rule_2$ is used.

uxz is also in *L*(*G*).

Context-Free Pumping



There are infinitely many derivations in *G*, such as:

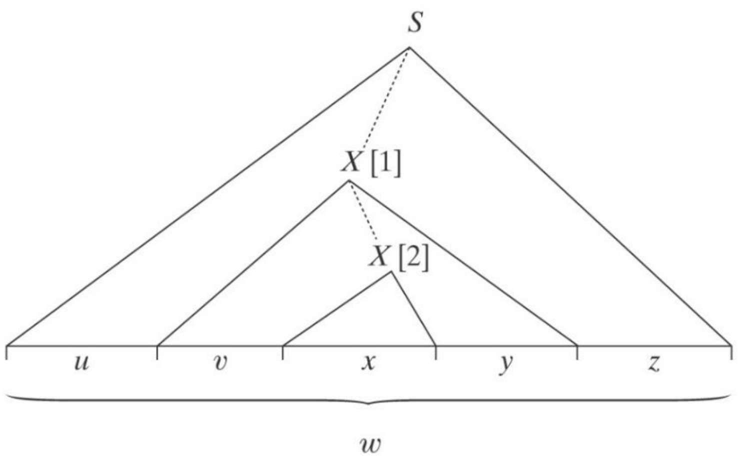
$S \Rightarrow^* uXz \Rightarrow^* uvXyz \Rightarrow^* uvvXyyz \Rightarrow^* uvvxyyz$

Those derivations produce the strings:

uv^2xy^2z, uv^3xy^3z, …

All of those strings are also in *L*(*G*).

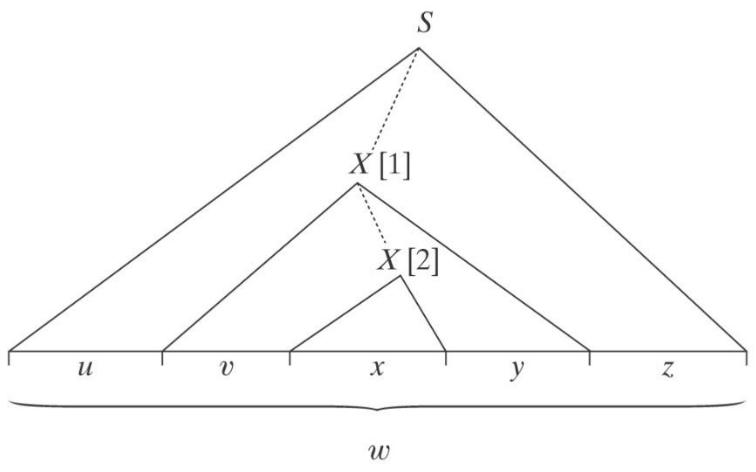
Context-Free Pumping



If $rule_1 = X \rightarrow X$a, we could get $v = \varepsilon$.
If $rule_1 = X \rightarrow$ aX, we could get $y = \varepsilon$.
But it is not possible that both *v* and *y* are ε.

If they were, then the derivation $S \Rightarrow^* uXz \Rightarrow^* uxz$ would also yield *w* and it would create a parse tree with fewer nodes. But, that contradicts the assumption that we started with a tree with the smallest possible number of nodes.

Context-Free Pumping



The height of the subtree rooted at [1] is at most $n + 1$.

So $|vxy| \leq b^{n+1}$.

The Context-Free Pumping Theorem

***Theorem 13.4*:** If L is a context-free language, then

$\exists k \geq 1$ ($\forall$ strings $w \in L$, where $|\boldsymbol{w}| \geq \boldsymbol{k}$

($\exists u, v, x, y, z$ ($w = uvxyz$,

$\boldsymbol{vy} \neq \boldsymbol{\varepsilon}$,

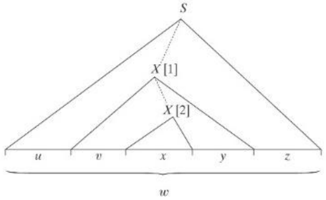
$|\boldsymbol{vxy}| \leq \boldsymbol{k}$ and

$\forall \boldsymbol{q} \geq \mathbf{0}$ **(** $\boldsymbol{uv^q xy^q z}$ **is in** $\boldsymbol{L}$**))))**

The Context-Free Pumping Theorem

Proof Idea:

L is generated by some CFG $G = (V, \Sigma, R, S)$ with n nonterminal symbols and branching factor b.



Let k be b^{n+1}.

The longest string that can be generated by G with no repeated nonterminals in the resulting parse tree has length b^n.

Assuming that $b \geq 2$, it must be the case that $b^{n+1} > b^n$. So let w be any string in $L(G)$ where $|w| \geq k$.

The Context-Free Pumping Theorem

Let *w* be any string in *L*(*G*) where **|*w*| ≥ *k*.**

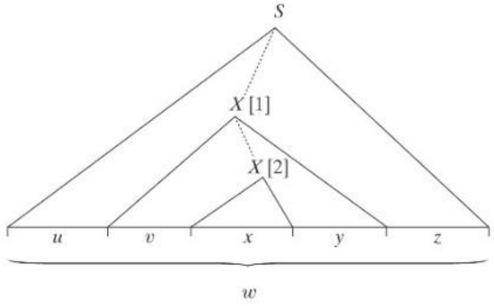
Let *T* be any smallest parse tree for *w*.

***T* must have height at least *n* + 1.**

Choose some path in *T* of length at least *n* + 1.

Let *X* be the bottom-most repeated nonterminal along that path.

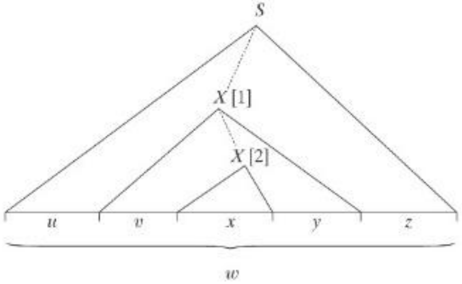
Then *w* can be rewritten as *uvxyz*.



The Context-Free Pumping Theorem

Then *w* can be rewritten as *uvxyz*.

The tree rooted at [1] has height at most *n* + 1.

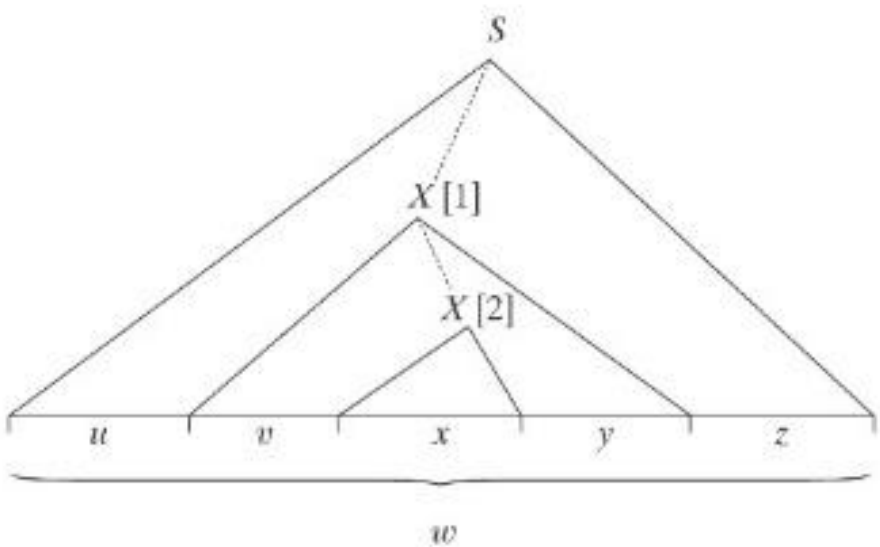


- Thus its yield, *vxy*, has length less than or equal to b^{n+1}, which is *k*.
- $vy \neq \varepsilon$ since if *vy* were ε then there would be a smaller parse tree for *w* and we chose *T* so that that wasn't so.

The Context-Free Pumping Theorem

- *uxz* must be in *L* because $rule_2$ could have been used immediately at [1].

- For any $q \geq 1$, uv^qxy^qz must be in *L* because $rule_1$ could have been used *q* times before finally using $rule_2$.



Showing That *L* is not Context-free

- The pumping theorem is true for every context-free language!
- If we could show the pumping theorem is not true of some language L, then L is not context-free!

- **Proof by Contraction:**

  1. Suppose some language L is contet-free, then it would possess certain properties.
  2. Show that L does not posses those properties.
  3. Therefore, L is not context-free.

Regular vs CF Pumping Theorems

Similarities:

- We choose *w*, the string to be pumped.
- We choose a value for *q* that shows that *w* isn't pumpable.
- We may apply closure theorems before we start.

Differences:

- Two regions, *v* and *y*, must be pumped in tandem.
- We don't know anything about where in the strings *v* and *y* will fall. All we know is that they are reasonably "close together", i.e., $|vxy| \le k$.
- Either *v* or *y* could be empty, although not both.

Example 13.1

$L = A^nB^nC^n = \{a^n b^n c^n, n \geq 0\}$ is **not context-free**!

Proof Idea: Proof by Contradiction.

Suppose $L = A^nB^nC^n$ is CFL.
There exists k st any string w where $|w| \geq k$ must satisfy the CFL pumping theorem.

We will show one string that does not satisfy the CFL pumping theorem.

Choose $w = a^k b^k c^k$

Example 13.1

Choose $w = \mathtt{a}^k \mathtt{b}^k \mathtt{c}^k$

1 | 2 | 3

- If either *v* or *y* spans regions, then let *q* = 2 (i.e., pump in once). The resulting string will have letters out of order and thus not be in $A^nB^nC^n$.

- If both *v* and *y* each contain only one distinct character, then set *q* to 2. Additional copies of at most two different characters are added, leaving the third unchanged. There are no longer equal numbers of the three letters, so the resulting string is not in $A^nB^nC^n$.

So, $A^nB^nC^n$ is not context-free!

Example 13.2

L = $\{a^{n^2}, n \geq 0\}$ is **not context-free**!

Proof Idea: Proof by Contradiction.

Suppose L is CFL.

Choose $n = k^2$, then $n^2 = k^4$. Choose $w = a^{k^4}$.

Example 13.2

Choose $n = k^2$, then $n^2 = k^4$. Choose $w = a^{k^4}$.

$vy = \mathtt{a}^p$, for some nonzero p.
Set q to 2. The resulting string, s, is a^{k^4+p}. It must be in L.
But it isn't because it is too short:

w:	next longer string in L:
$(k^2)^2$ a's	$(k^2 + 1)^2$ a's
k^4 a's	$k^4 + 2k^2 + 1$ a's

For s to be in L, $p = |vy|$ would have to be at least $2k^2 + 1$.
But $|vxy| \leq k$, so p can't be that large. Thus s is not in L.

So, L is not context-free.

Example 13.3

L = $\{a^n b^m a^n$, n, $m \geq 0$ and $n \geq m\}$ is **not context-free**!

Proof Idea: Proof by Contradiction.

Suppose L is CFL.

Choose $w = a^k b^k a^k$

Example 13.3

Choose $w = \mathtt{a}^k\mathtt{b}^k\mathtt{a}^k$

```
aaa … aaabbb … bbbaaa … aaa
|       1      |     2       |      3      |
```

So, *L* is not context-free.

Example 13.4

WcW = $\{w\mathtt{c}w : w \in \{\mathtt{a}, \mathtt{b}\}^*\}$ is **not context-free**!

Proof Idea: Proof by Contradiction.

Suppose WcW is CFL.

Choose $w = \mathtt{a}^k\mathtt{b}^k\mathtt{c}\mathtt{a}^k\mathtt{b}^k$.

Example 13.4

Choose w = $a^k b^k c a^k b^k$.

```
aaa … aaabbb …  bbbcaaa … aaabbb … bbb
|      1       |       2        |3|    4          |      5        |
```

- If v or y overlaps region 3, set q to 0. The resulting string will no longer contain a c.
- If both v and y occur before region 3 or they both occur after region 3, then set q to 2. One side will be longer than the other.
- If either v or y overlaps region 1, then set q to 2. In order to make the right side match, something would have to be pumped into region 4. Violates $|vxy| \leq k$.
- If either v or y overlaps region 2, then set q to 2. In order to make the right side match, something would have to be pumped into region 5. Violates $|vxy| \leq k$.

So WcW is not context-free!

Closure and Non-Closure Properties of CFLs

Operations that preserve the Property of being a Context-Free Language!

Closure Theorems for CFLs

The context-free languages are closed under:

- Union
- Concatenation
- Kleene star

- Reverse
- Letter substitution

Closure Theorems for CFLs

Theorem 13.5 The CFLs are closed under union, concatenation, Kleene star, reverse, and letter substitution.

Proof Idea:

Proof by Construction.

Closure Under Union

Let $G_1 = (V_1, \Sigma_1, R_1, S_1)$, and $G_2 = (V_2, \Sigma_2, R_2, S_2)$.

Assume that G_1 and G_2 have disjoint sets of nonterminals, not including S.

Let $L = L(G_1) \cup L(G_2)$.

We can show that L is CF by exhibiting a CFG for it:

$$G = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2,\\ R_1 \cup R_2 \cup \{S \rightarrow S_1, S \rightarrow S_2\},\\ S)$$

Closure Under Concatenation

Let $G_1 = (V_1, \Sigma_1, R_1, S_1)$, and $G_2 = (V_2, \Sigma_2, R_2, S_2)$.

Assume that G_1 and G_2 have disjoint sets of nonterminals, not including S.

Let $L = L(G_1)L(G_2)$.

We can show that L is CF by exhibiting a CFG for it:

$$G = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2,\\ R_1 \cup R_2 \cup \{S \to S_1\, S_2\},\\ S)$$

Closure Under Kleene Star

Let $G = (V, \Sigma, R, S_1)$.

Assume that G does not have the nonterminal S.

Let $L = L(G)^*$.

We can show that L is CF by exhibiting a CFG for it:

$$G = (V_1 \cup \{S\}, \Sigma_1, R_1 \cup \{S \rightarrow \varepsilon, S \rightarrow S\, S_1\}, S)$$

Non-Closure Theorems for CFLs

The context-free languages are **not closed** under:

- Intersection
- Complement
- Difference

Non-Closure Theorems for CFLs

Theorem 13.6 The CFLs are NOT closed under intersection, complement or difference.

Proof Idea:

Proof by Counterexample.

Non-Closure Under Intersection

$L_1 = \{a^n b^n c^m : n, m \geq 0\}$
$L_2 = \{a^m b^n c^n : n, m \geq 0\}$

Both L_1 and L_2 are context-free.

But now consider:

$L = L_1 \cap L_2$

$= \{a^n b^n c^n : n \geq 0\}$ not context-free!

Non-Closure Under Complement

$L_1 \cap L_2 = \neg(\neg L_1 \cup \neg L_2)$

- The context-free languages are closed under union.
- So if they were closed under complement, they would be closed under intersection (which they are not).

Example: $\neg A^nB^nC^n$ is context-free. But $\neg(\neg A^nB^nC^n) = A^nB^nC^n$ is not context-free.

Non-Closure Under Difference

$\neg L = \Sigma^* - L$.

- Σ^* is context-free.
- If the context-free languages were closed under difference, the complement of any context-free language would necessarily be context-free.
- But that is not so.

Closure Theorems for CFLs with RLs

Theorem 13.7 The CFLs are closed under intersection **with the regular languages**.

Proof Idea:

Proof by Construction.

Closure Theorems for CFLs with RLs

Theorem 13.8 The difference L1 – L2 between a CFL L1 and a **RL L2** is context-free.

Proof Idea:

Proof by Construction.

Example 13.5

L = {$\mathtt{a}^n\mathtt{b}^n$: $n \geq 0$ <u>and</u> $n \neq 1776$} is **context-free**!

L = {$\mathtt{a}^n\mathtt{b}^n$: $n \geq 0$} – {$\mathtt{a}^{1776}\mathtt{b}^{1776}$}.

Here,
{$\mathtt{a}^n\mathtt{b}^n$: $n \geq 0$} is context-free.
{$\mathtt{a}^{1776}\mathtt{b}^{1776}$} is regular.

So, L is context-free.

Deterministic CFLs

Deterministic PDAs

A PDA *M* is *deterministic* iff:

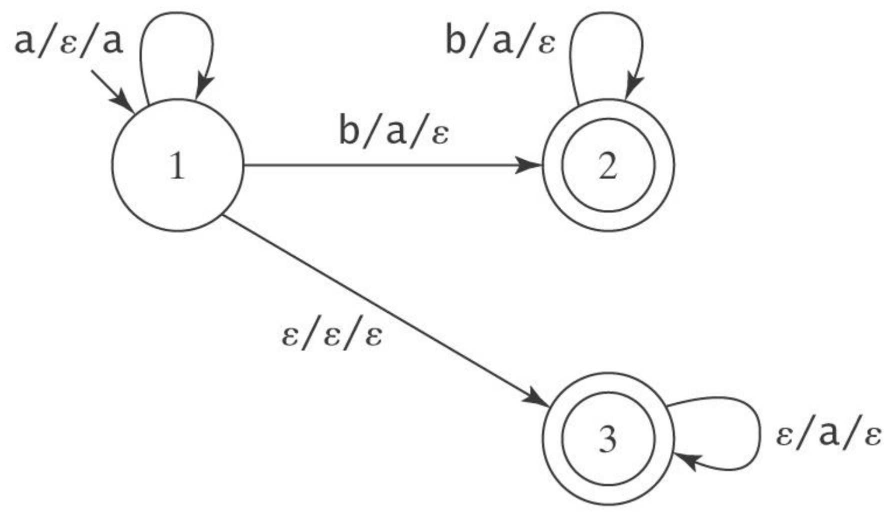
- Δ_M contains no pairs of transitions that compete with each other, and

- Whenever *M* is in an accepting configuration it has no available moves.

An NDPDA for *L*

$L = a^* \cup \{a^n b^n : n > 0\}$.

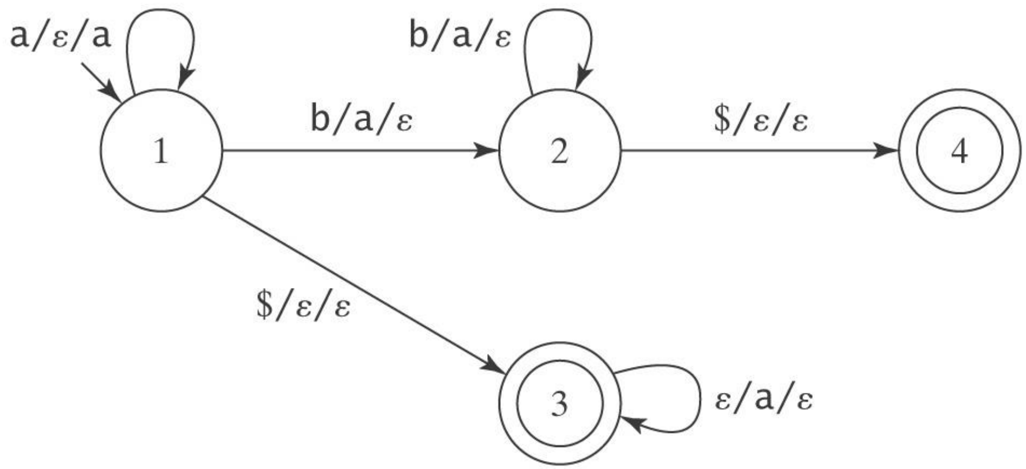
NDPDA?



A DPDA for *L*$

L = a* ∪ {$a^n b^n$: $n > 0$}.

DPDA?



Deterministic CFLs

A language *L* is ***deterministic context-free*** iff *L*$ ($= an end-of-string marker) can be accepted by some **deterministic PDA**.

DCFL = DPDA

Non-Equivalence of NPDA and DPDA

- NPDA ≠ DPDA
- DPDA is weaker than NPDA!

DCFLs and CFLs

Theorem 13.13 There exist CLFs that are not deterministic.

Proof Idea:

Proof By Example.

Let $L = \{a^i b^j c^k, i \neq j \text{ or } j \neq k\}$. L is CF. If L is DCF then so is:

$L' = \neg L'$.

$= \{a^i b^j c^k, i, j, k \geq 0 \text{ and } i = j = k\} \cup$

$\{w \in \{a, b, c\}^* : \text{the letters are out of order}\}$.

But then so is:

$L'' = L' \cap a^*b^*c^*$.

$= \{a^n b^n c^n, n \geq 0\}$.

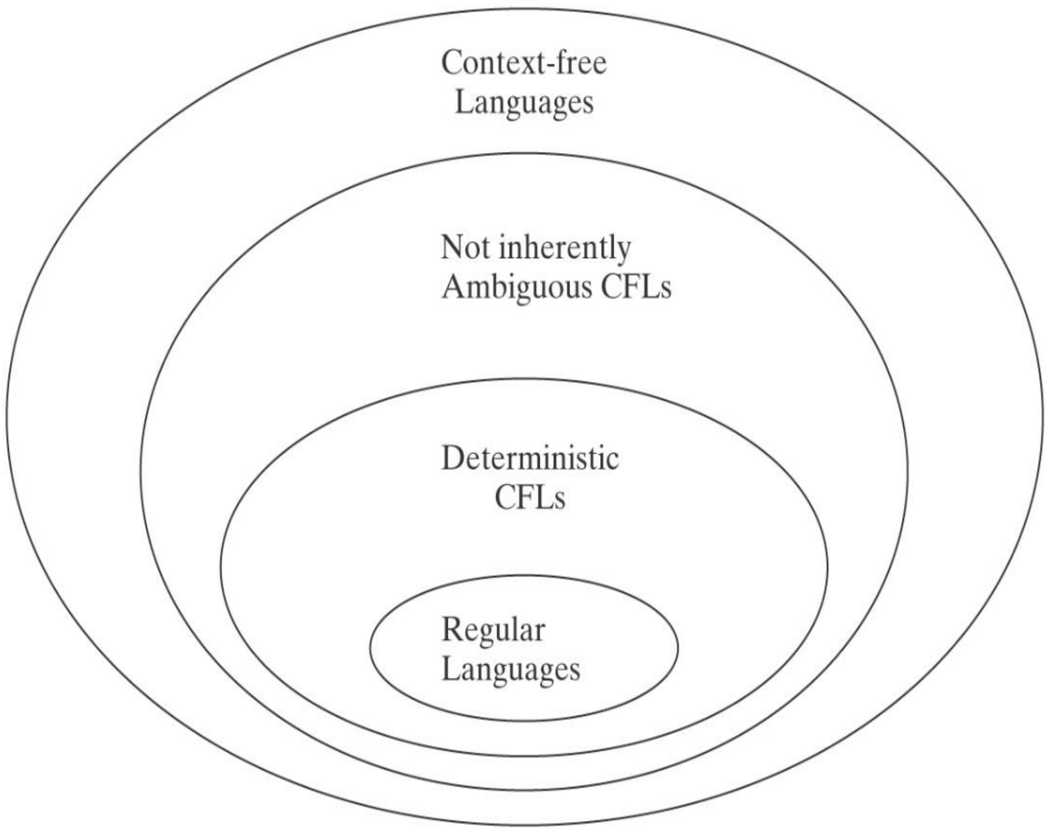
But it isn't. So L is context-free but not deterministic context-free.

DCFLs and Unambiguous CFGs

Theorem 13.14 Every RL is deterministic CFL.

Theorem 13.15 For every **deterministic** CFL, there exists an **unambiguous CFG**.

The CFL Hierarchy

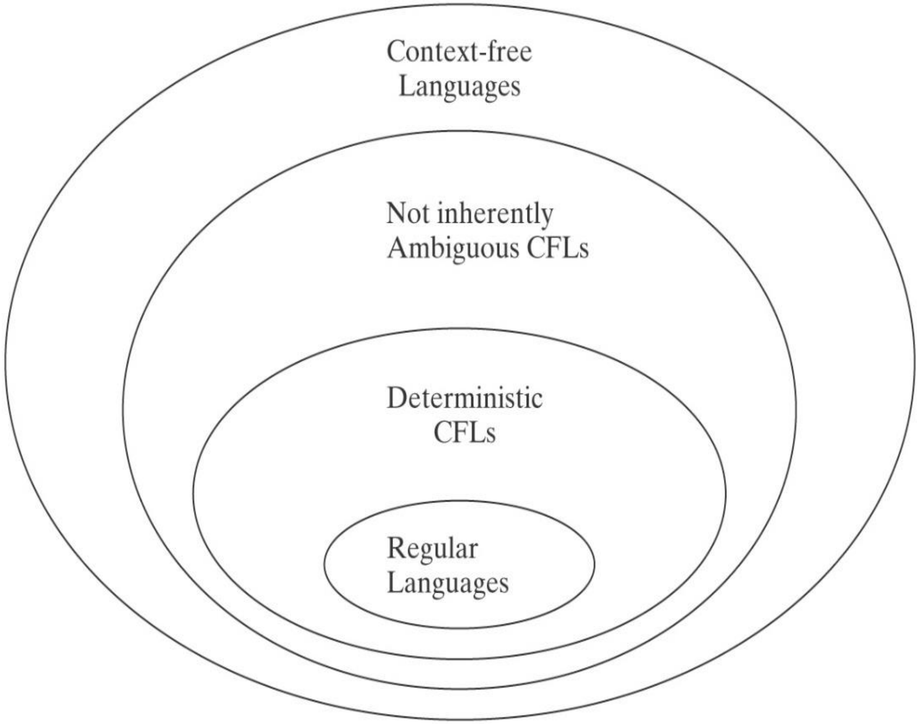


The CFL Hierarchy

- $L_1 = \{a^i b^j c^k : i, j, k \geq 0 \text{ and } (i = j) \text{ or } (j = k)\} = \{a^n b^n c^m : n, m \geq 0\} \cup \{a^n b^m c^m : n, m \geq 0\}$

- $L_2 = \{a^n b^n c^m d : n, m \geq 0\} \cup \{a^n b^m c^m e : n, m \geq 0\}$
- PalEven = $\{ww^R : w \in \{a, b\}^*\}$

- $A^nB^n = \{a^n b^n : n \geq 0\}$

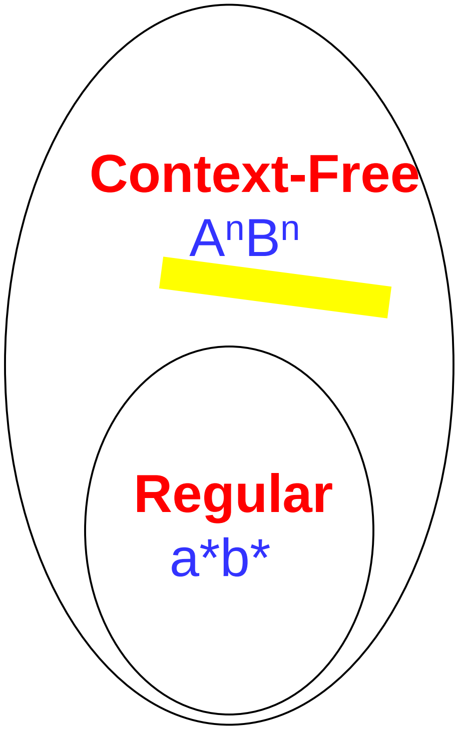


Language Summary

IN

CF grammar
PDA
Closure

R grammar
Regular Expression
FSM



OUT

Pumping
Closure

Pumping
Closure

Reading Assignment

Chapter 13:

Sections
13.1
13.2
13.3
13.4

In-Class Exercises

Chapter 13:

1 – a & j & q
3

Algorithms and Decision Procedures for Context-Free Languages

Membership

Theorem 14.1 Given a context-free language L and a string w, there exists a decision procedure that answers the question, is $w \in L$?

Emptiness & Finiteness

Theorem 14.4 Given a CFL L, there exists a decision procedure that answers the question, is L(M) = ∅? & is L finite?

Equivalence of DCFLs

Theorem 14.5 Given two ***deterministic*** context-free languages L_1 and L_2, there exists a decision procedure to determine whether $L_1 = L_2$?

Undecidable Questions about CFLs

- Is $L = \Sigma^*$? **(Totality)**
- Is the complement of L context-free?
- Is L regular?
- Is $L_1 = L_2$? **(Equivalence)**
- Is $L_1 \subseteq L_2$?
- Is $L_1 \cap L_2 = \emptyset$?
- Is L inherently ambiguous?
- Is G ambiguous?

Reading Assignment

Chapter 14:

Sections
14.1
14.2
14.3

In-Class Exercises

Chapter 14:

1 - a

Context Free Languages: Summary

Context-Free Languages

- context-free grammars
- = NDPDAs
- parse
- find unambiguous grammars
- reduce nondeterminism in PDAs
- find efficient parsers
- closed under:
  - ♦ concatenation
  - ♦ union
  - ♦ Kleene star
  - ♦ intersection w/ reg. langs
- pumping theorem
- D ≠ ND

RLs vs CFLs:Summary

Regular Languages

- regular exprs.
  or
- regular grammars
- = DFSMs
- recognize
- minimize FSMs
- closed under:
  - ♦ concatenation
  - ♦ union
  - ♦ Kleene star
  - ♦ complement
  - ♦ intersection
- pumping theorem
- D = ND

vs.

Context-Free Languages

- context-free grammars
- = NDPDAs
- parse
- find unambiguous grammars
- reduce nondeterminism in PDAs
- find efficient parsers
- closed under:
  - ♦ concatenation
  - ♦ union
  - ♦ Kleene star
  - ♦ intersection w/ reg. langs
- pumping theorem
- D ≠ ND