香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

SCHOOL OF
DATA SCIENCE
數據科學學院

CSC 6011 - Theory of Computation - Lecture 5:
The Context-Free Pumping Lemma, Turing Machines

Time: Mon & Wed, 4:00 - 5:20 PM

Location: Teaching A, 107

Instructor: Tao LIN (林涛)

TA: Yefan GAO (高叶繁)

Fall 2026

Announcement:
Homework 1 has been posted.
Due on Sep 30.

Lecture 5

Last time:
- Context free grammars (CFGs)
- Context free languages (CFLs)
- Pushdown automata (PDA)
- Converting CFGs to PDAs

Today:
- Proving languages not Context Free
- Turing machines
- Turing-recognizable and Turing-decidable languages

Equivalence of CFGs and PDAs

Recall Theorem: A is a CFL iff some PDA recognizes A

CFG → PDA: proved.

PDA → CFG: know the fact; the proof is not required.

Corollaries:

1) Every regular language is a CFL.
2) If A is a CFL and B is regular, then $A \cap B$ is a CFL.

Proof sketch of (2):

While reading the input, the finite control of the PDA for A simulates the DFA for B.

Note 1: If A and B are CFLs then $A \cap B$ may not be a CFL (will show today).

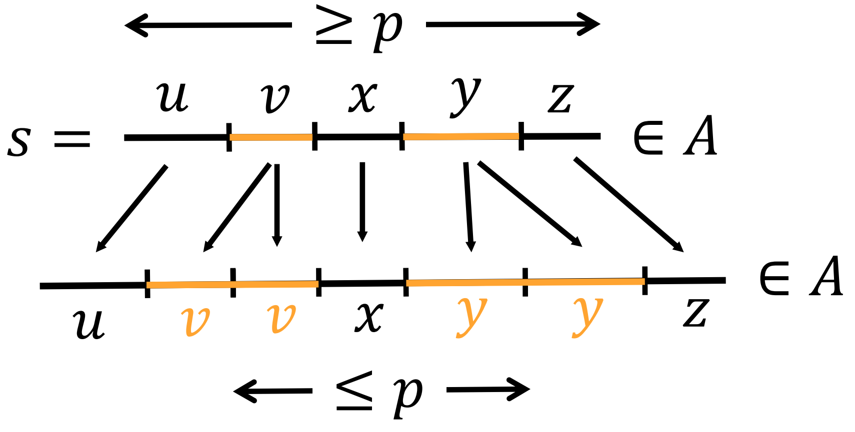
Note 2: The class of CFLs is closed under $\cup, \circ, *$ (Homework 2).

Proving languages not Context Free

Let $B = \{0^k 1^k 2^k \mid k \geq 0\}$. We will show that B isn't a CFL.

Pumping Lemma for CFLs: For every CFL A, there is a p such that: if $s \in A$ and $|s| \geq p$, then $s = uvxyz$ where

1) $uv^i xy^i z \in A$ for all $i \geq 0$
2) $vy \neq \varepsilon$
3) $|vxy| \leq p$



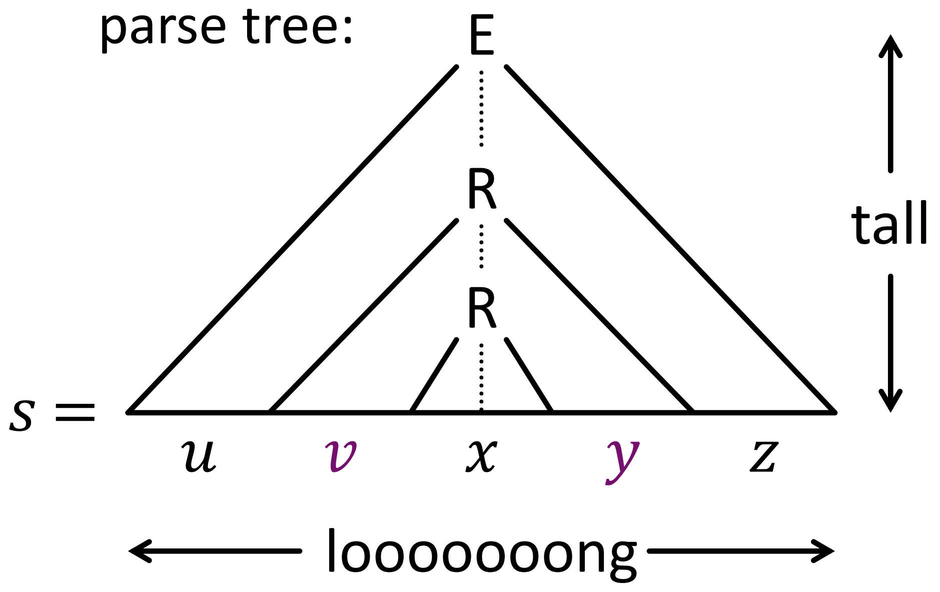
Informally: All long strings in A can be pumped and stay in A.

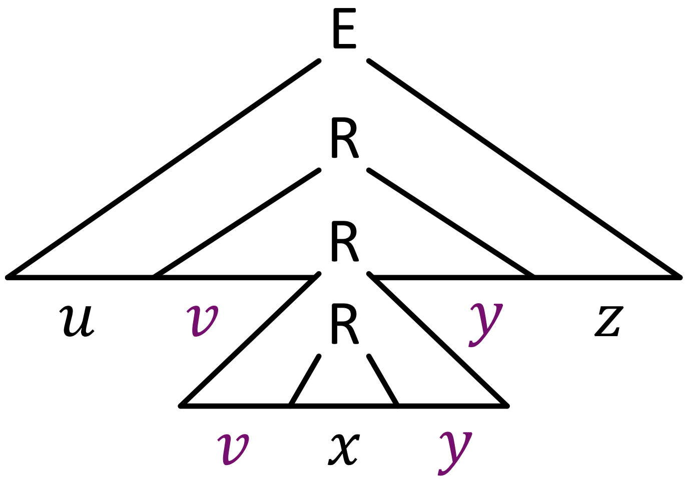
Pumping Lemma – Proof by picture

Pumping Lemma for CFLs: For every CFL A, there is a p such that: if $s \in A$ and $|s| \geq p$, then $s = uvxyz$ where

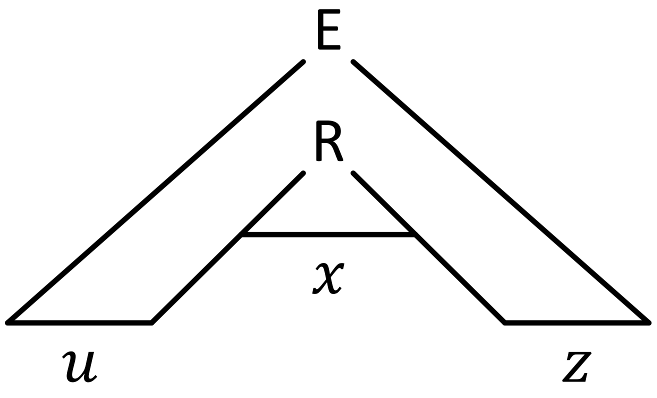
1) $uv^ixy^iz \in A$ for all $i \geq 0$
2) $vy \neq \varepsilon$
3) $|vxy| \leq p$

“cutting and pasting” argument





Generates $uvvxyyz$
$= uv^2xy^2z$



Generates uxz
$= uv^0xy^0z$

Pumping Lemma – Proof details

Pumping Lemma for CFLs: For every CFL A, there is a p such that: if $s \in A$ and $|s| \geq p$, then $s = uvxyz$ where

1) $uv^ixy^iz \in A$ for all $i \geq 0$ ...cutting and pasting
2) $vy \neq \varepsilon$ ...start with the smallest parse tree for s
3) $|vxy| \leq p$ ...pick the lowest repetition of a variable

Let $b =$ the length of the longest right-hand side of a rule (E → E+T)

$= $ the max branching of the parse tree

Let $h =$ the height of the parse tree for s.

A tree of height h and max branching b has at most b^h leaves.

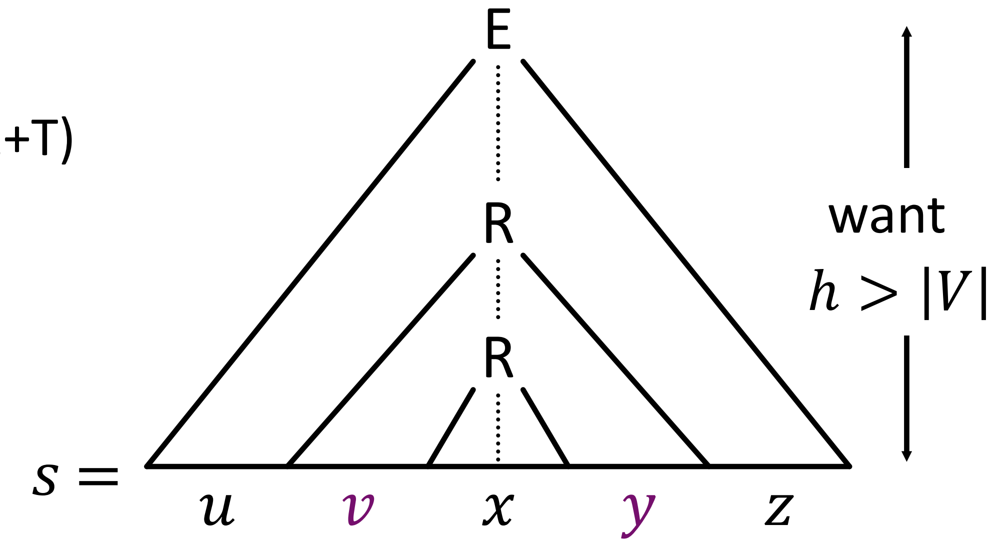
So $|s| \leq b^h$.

Let $p = b^{|V|} + 1$ where $|V| =$ # variables in the grammar.

So if $|s| \geq p > b^{|V|}$ then $b^h > b^{|V|}$ and so $h > |V|$.

Thus at least $|V| + 1$ variables occur in the longest path.

So some variable R must repeat on a path.



$b^h \geq |s| \geq p = ?\ b^{|V|} + 1$
implies $h > |V|$

Example 1 of Proving Non-CF

Pumping Lemma for CFLs: For every CFL A, there is a p such that: if $s \in A$ and $|s| \geq p$, then $s = uvxyz$ where

1) $uv^ixy^iz \in A$ for all $i \geq 0$
2) $vy \neq \varepsilon$
3) $|vxy| \leq p$

Prove that $B = \{0^k1^k2^k \mid k \geq 0\}$ is not a CFL

$$s = \underbrace{00\cdots0011\cdots1122\cdots22}$$

$u \mid v \mid x \mid y \mid z$

$\leftarrow \leq p \rightarrow$

Proof by Contradiction:

Assume (to get a contradiction) that B <u>is</u> a CFL .

The CFL pumping lemma gives p as above. Let $s = 0^p1^p2^p \in B$.

Pumping lemma says that we can divide $s = uvxyz$ satisfying the 3 conditions:

- Condition 3 $(|vxy| \leq p)$ implies that vxy cannot contain both 0s and 2s.
- So uv^2xy^2z has unequal numbers of 0s, 1s, and 2s.

Thus $uv^2xy^2z \notin B$, violating Condition 1. Contradiction!

Therefore our assumption (B is a CFL) is false. We conclude that B is not a CFL .

Mini-Quiz 5.1

We just showed that $B = \{0^k 1^k 2^k | \, k \geq 0\}$ is not a CFL

Let $A_1 = \{0^k 1^k 2^l | \, k, l \geq 0\}$ (equal numbers of 0s and 1s)

Let $A_2 = \{0^l 1^k 2^k | \, k, l \geq 0\}$ (equal numbers of 1s and 2s)

Observe that PDAs can recognize A_1 and A_2.

What can we now conclude?

a) The class of CFLs is not closed under intersection.

b) The Pumping Lemma shows that $A_1 \cup A_2$ is not a CFL.

c) The class of CFLs is closed under complement.

De Morgan's Laws:

- $(A \cup B)^c = A^c \cap B^c$
- $(A \cap B)^c = A^c \cup B^c \;\Rightarrow\; A \cap B = (A^c \cup B^c)^c$

Example 2

Pumping Lemma for CFLs: For every CFL A, there is a p such that: if $s \in A$ and $|s| \geq p$, then $s = uvxyz$ where
1) $uv^ixy^iz \in A$ for all $i \geq 0$
2) $vy \neq \varepsilon$
3) $|vxy| \leq p$

Question:
What string can we choose to apply the lemma to prove that F is not a CFL?

Let $F = \{ww |\ w \in \Sigma^*\}$. $\Sigma = \{0{,}1\}$.

Show: F is not a CFL.

Assume (for contradiction) that F is a CFL.

The CFL pumping lemma gives p as above.

Need to choose $s \in F$. Which s?

Try $s_1 = 0^p10^p1 \in F$.

But s_1 <u>can</u> be pumped and stay inside F. Bad choice of s.

$$s_1 = 000\cdots001000\cdots001$$

$u \quad v \; x \; y \quad z$

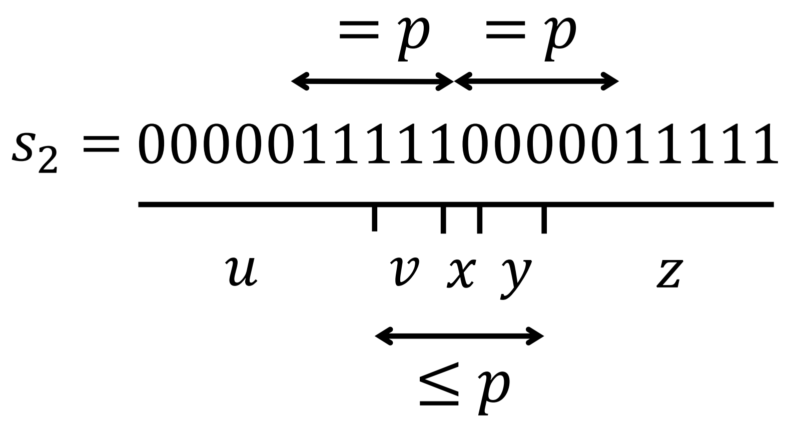
$\longleftrightarrow$

$\leq p$

Example 2 – A good choice of string

Pumping Lemma for CFLs: For every CFL A, there is a p such that: if $s \in A$ and $|s| \geq p$, then $s = uvxyz$ where

1) $uv^ixy^iz \in A$ for all $i \geq 0$
2) $vy \neq \varepsilon$
3) $|vxy| \leq p$



Let $F = \{ww | \, w \in \Sigma^*\}$. $\Sigma = \{0,1\}$.

Show: F is not a CFL.

Try $s_2 = 0^p1^p0^p1^p \in F$.

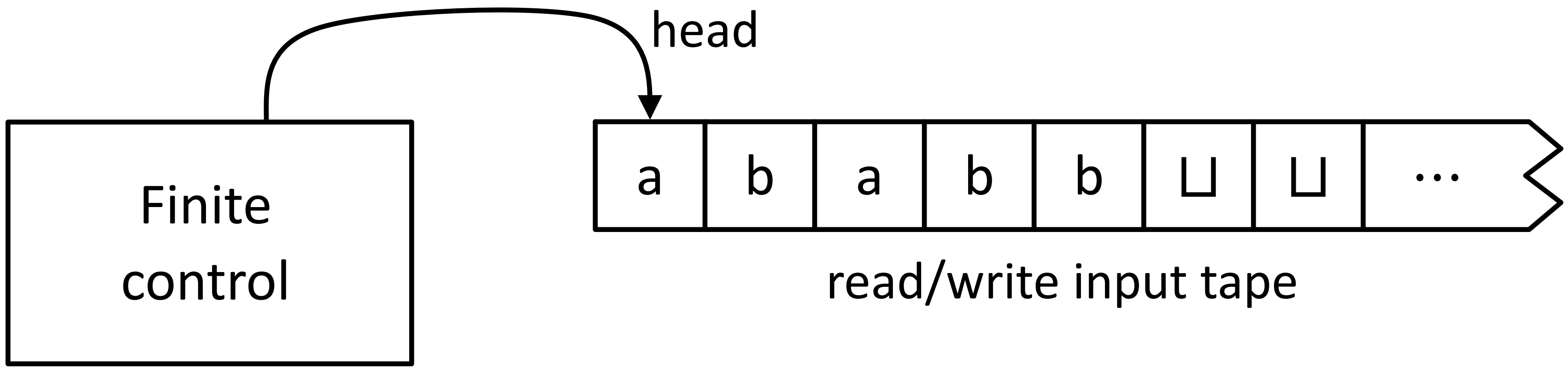
Consider any possible decomposition $s_2 = uvxyz$.

Condition 3 implies that vxy does not overlap two runs of 0s or two runs of 1s.

Therefore, in uv^2xy^2z, two runs of 0s or two runs of 1s have unequal length.

So $uv^2xy^2z \notin F$, violating Condition 1. Contradiction! Thus, F is not a CFL.

Turing Machines (TMs)

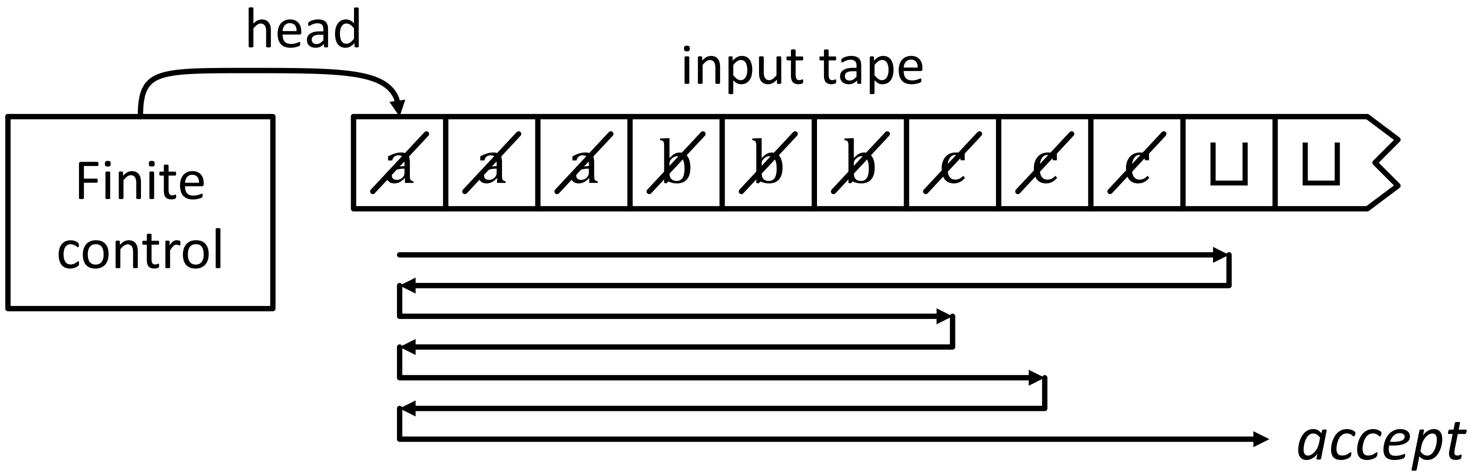


1) Head can read and write
2) Head is two way (can move left or right)
3) Tape is infinite (to the right)
4) Infinitely many blanks "⊔" follow input
5) Can accept or reject any time (not only at end of input)

TM – example

TM recognizing $B = \{\mathrm{a}^k\mathrm{b}^k\mathrm{c}^k \mid k \geq 0\}$ (recall: this is not a CFL)

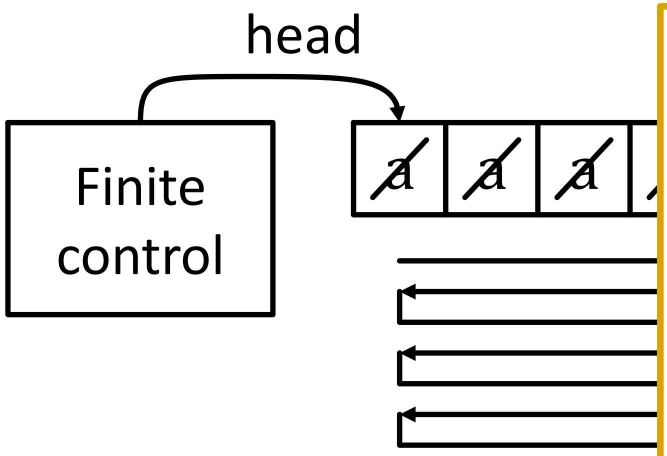
1) Scan right until $\sqcup$ while checking if input is in $\mathrm{a}^*\mathrm{b}^*\mathrm{c}^*$, *reject* if not.
2) Return head to left end.
3) Scan right, crossing off single a, b, and c.
4) If having crossed off the last one of every symbol, *accept*.
5) If having crossed off the last one of some symbol but not others, *reject*.
6) If some symbols remain, return to left end and repeat from (3).



TM – example

TM recognizing $B = \{\mathrm{a}^k\mathrm{b}^k\mathrm{c}^k \mid k \geq 0\}$ (recall: this is not a CFL)

1) Scan right until $\sqcup$ while checking if input is in $\mathrm{a}^*\mathrm{b}^*\mathrm{c}^*$, *reject* if not.
2) Return head to left end.
3) Scan right, crossing off single a, b, and c.
4) If having crossed off the last one of every symbol, *accept*.
5) If having crossed off the last one of some symbol but not others, *reject*.
6) If some symbols remain, return to left end and repeat from (3).



Mini-Quiz 5.2

How do we get the effect of “crossing off” in a Turing machine?

a) Add that feature to the model.

b) Use a tape alphabet $\Gamma = \{\mathrm{a}, \mathrm{b}, \mathrm{c}, \cancel{\mathrm{a}}, \cancel{\mathrm{b}}, \cancel{\mathrm{c}}, \sqcup\}$.

c) All Turing machines come with an eraser.

TM – Formal Definition

Defn: A <u>Turing Machine</u> (TM) is a 7-tuple $(Q, \Sigma, \Gamma, \delta, q_0, q_{\mathrm{acc}}, q_{\mathrm{rej}})$

Σ input alphabet

Γ tape alphabet ($\Sigma \subsetneq \Gamma$)

δ: $Q \times \Gamma \to Q \times \Gamma \times$ {L, R} (L = Left, R = Right)

$\delta(q, \mathrm{a}) = (r, \mathrm{b}, \mathrm{R})$

On input w, a TM M may halt (enter q_{acc} or q_{rej}) or run forever ("loop").

So M has 3 possible outcomes for each input w:

1. <u>*Accept*</u> w (enter q_{acc})
2. <u>*Reject*</u> w by halting (enter q_{rej})
3. <u>*Reject*</u> w by looping (running forever)

Remark:
This Turing machine model is deterministic.
We can change it to be nondeterministic by changing δ to δ: $Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{\mathrm{L}, \mathrm{R}\})$

TM Recognizers and Deciders

Let M be a TM. Then $L(M) = \{w \mid M \text{ accepts } w\}$.

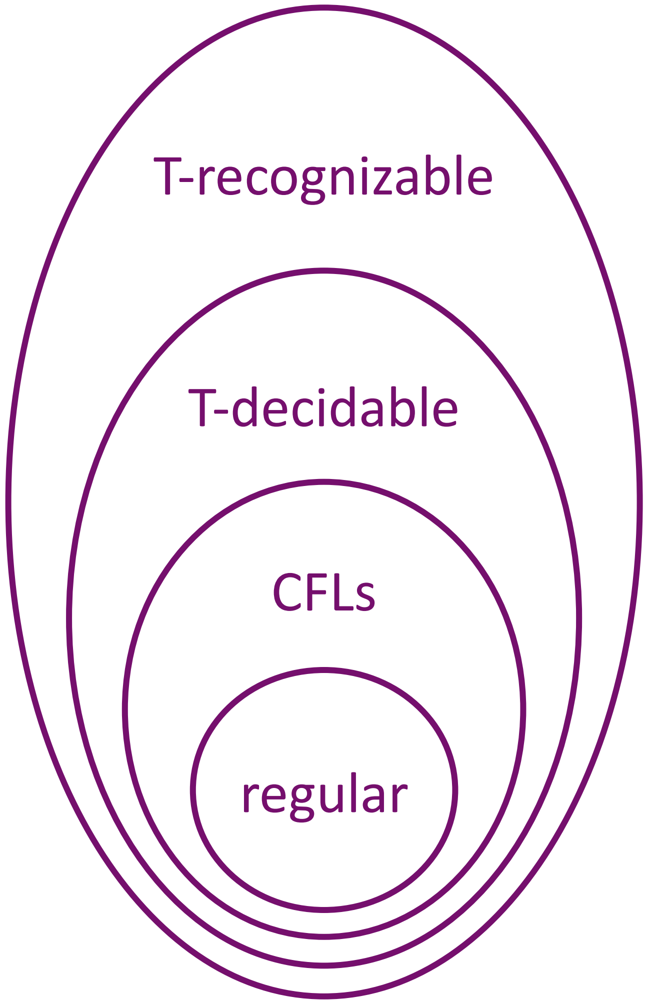
Say that M recognizes A if $A = L(M)$.

Defn: A is Turing-recognizable if $A = L(M)$ for some TM M.

Defn: TM M is a decider if M halts on all inputs.

Say that M decides A if $A = L(M)$ and M is a decider.

Defn: A is Turing-decidable if $A = L(M)$ for some TM decider M.



Quick review of today

1. CFL Pumping Lemma: a tool to show that languages are not context free.
2. Defined Turing machines (TMs).
3. Defined TM deciders (halt on all inputs).
4. T-recognizable and T-decidable languages.