18.404/6.840 Lecture 5

Last time:

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

Today: (Sipser §2.3, §3.1)

- Proving languages not Context Free
- Turing machines
- T-recognizable and T-decidable languages

Equivalence of CFGs and PDAs

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

$\rightarrow$ Done.

$\leftarrow$ Need to know the fact, not the proof

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).
Therefore the class of CFLs is not closed under $\cap$.

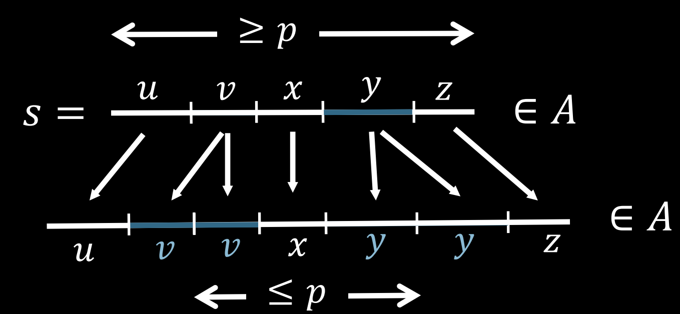
Note 2: The class of CFLs is closed under $\cup, \circ, *$ (see Pset 2).

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) $|vxy| \leq p$



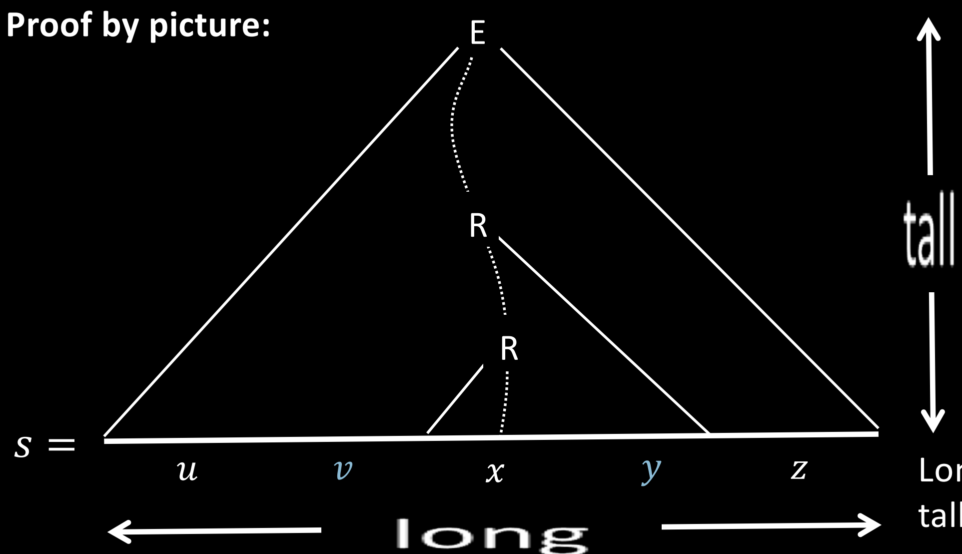
Informally: All long strings in A are pumpable and stay in A.

Pumping Lemma – Proof

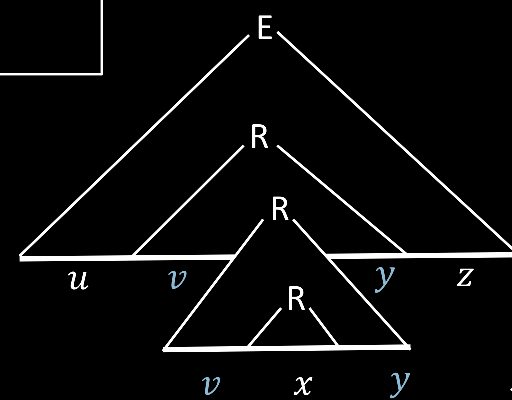
Pumping Lemma for CFLs: For every CFL A, there is a p such that if $s \in A$ and $|s| \geq p$ then $s = uvxyz$ where

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

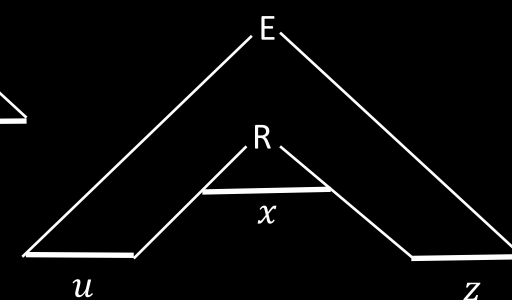
Proof by picture:



Long $s \rightarrow$ tall parse tree



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



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

"cutting and pasting" argument

Pumping Lemma – Proof details

For $s \in A$ where $|s| \geq p$, we have $s = uvxyz$ where:
1) $uv^ixy^iz \in A$ for all $i \geq 0$
2) $vy \neq \varepsilon$
3) $|vxy| \leq p$

Let $b =$ the length of the longest right hand side of a rule (E $\rightarrow$ E+T)
$=$ the max branching of the parse tree

E
E + T

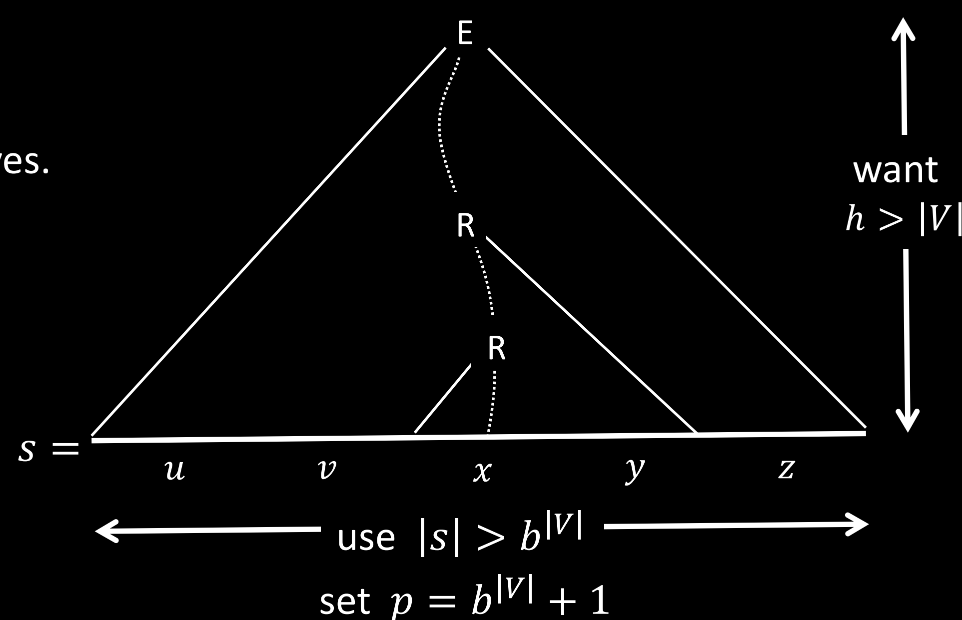
Let $h =$ the height of the parse tree for s.

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 $|s| > b^{|V|}$ and so $h > |V|$.

Thus at least $|V| + 1$ variables occur in the longest path.
So some variable R must repeat on a path.



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) $|vxy| \leq p$

Let $B = \{0^k1^k2^k \mid k \geq 0\}$

Show: B is not a CFL

Check-in 5.1

Let $A_1 = \{0^k1^k2^l \mid k, l \geq 0\}$ (equal #s of 0s and 1s)

Let $A_2 = \{0^l1^k2^k \mid k, l \geq 0\}$ (equal #s 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.

$s = 00 \cdots 0011 \cdots 1122 \cdots 22$

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

$\leftarrow \leq p \rightarrow$

Check-in 5.1

Example 2 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) $|vxy| \leq p$

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

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

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

Show s_2 cannot be pumped $s_2 = uvxyz$ satisfying the 3 conditions.

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.

$$s_1 = 000\cdots001000\cdots001$$

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

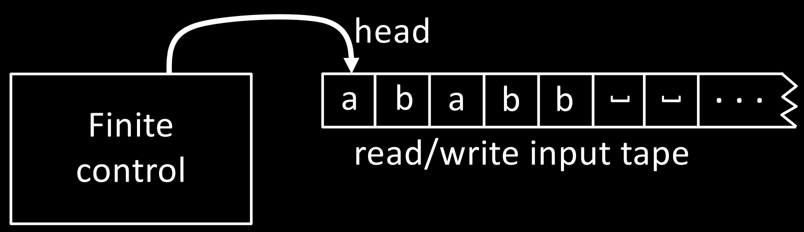
$\leftarrow \leq p \rightarrow$

$$s_2 = 0\cdots01\cdots10\cdots01\cdots1$$

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

$\leftarrow \leq p \rightarrow$

Turing Machines (TMs)

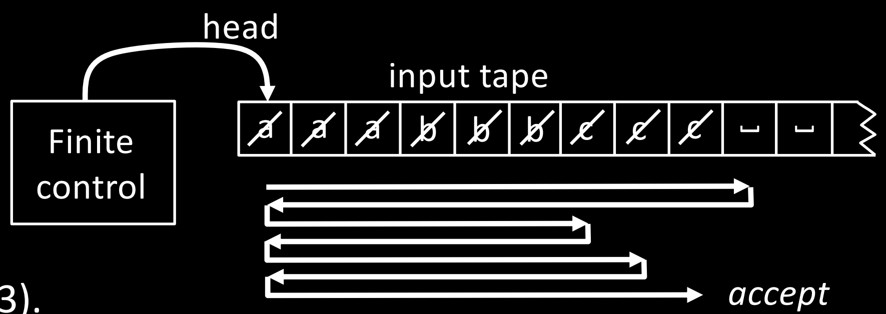


1) Head can read and write
2) Head is two way (can move left or right)
3) Tape is infinite (to the right)
4) Infinitely many blanks “␣“ follow input
5) Can accept or reject any time (not only at end of input)

TM – example

TM recognizing $B = \{a^k b^k c^k \mid k \geq 0\}$

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



Check-in 5.2

How do we get the effect of "crossing off" with a Turing machine?

a) We add that feature to the model.

b) We use a tape alphabet $\Gamma = \{a, b, c, \cancel{a}, \cancel{b}, \cancel{c}, ␣\}$.

c) All Turing machines come with an eraser.

Check-in 5.2

TM – Formal Definition

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

Σ input alphabet

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

δ: $\mathrm{Q}\times\Gamma \to Q\times\Gamma\times$ {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 M may run forever ("loop").

So M has 3 possible outcomes for each input w:

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

Check-in 5.3

This Turing machine model is deterministic.
How would we change it to be nondeterministic?

a) Add a second transition function.

b) Change δ to be δ: $\mathrm{Q}\times\Gamma \to \mathcal{P}(Q\times\Gamma\times$ {L, R} $)$

c) Change the tape alphabet Γ to be infinite.

Check-in 5.3

TM Recognizers and Deciders

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

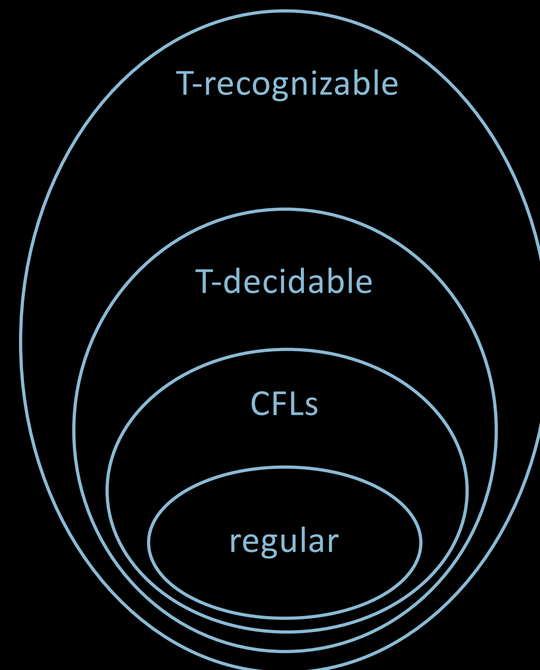
Say that M recognizes A if $A = L(M)$.

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. Proved the CFL Pumping Lemma as a tool for showing 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.