Lecture 14

CSE 431
Intro to Theory of Computation

Midterm: Next Friday here (MGH 058)
Closed book, closed notes

Review Session on Zoom: Wed Feb. 9, 4:30 pm

Proving that languages are not context-free.

Pumping Lemma for Context-Free Languages

Note: related lemma for regular languages. is not as good as 311 methods

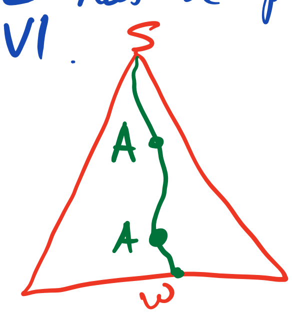
If L is a CFL then $\exists$ integer p (pumping length)
s.t.
$\forall w \in L$ with $|w| \geq p$
we can write $w = uvxyz$ s.t.

1. $|vy| \neq 0$ (can strengthen to $v \neq \varepsilon$)
2. $|vxy| \leq p$
3. $\forall i \geq 0.\ uv^ixy^iz \in L$

Proof Let G be CFG with $L = L(G)$ and assume that G is in Chomsky Normal Form.

Let V = # of variables in G

Suppose that $w \in L$ has a parse tree of height $> |V|$.

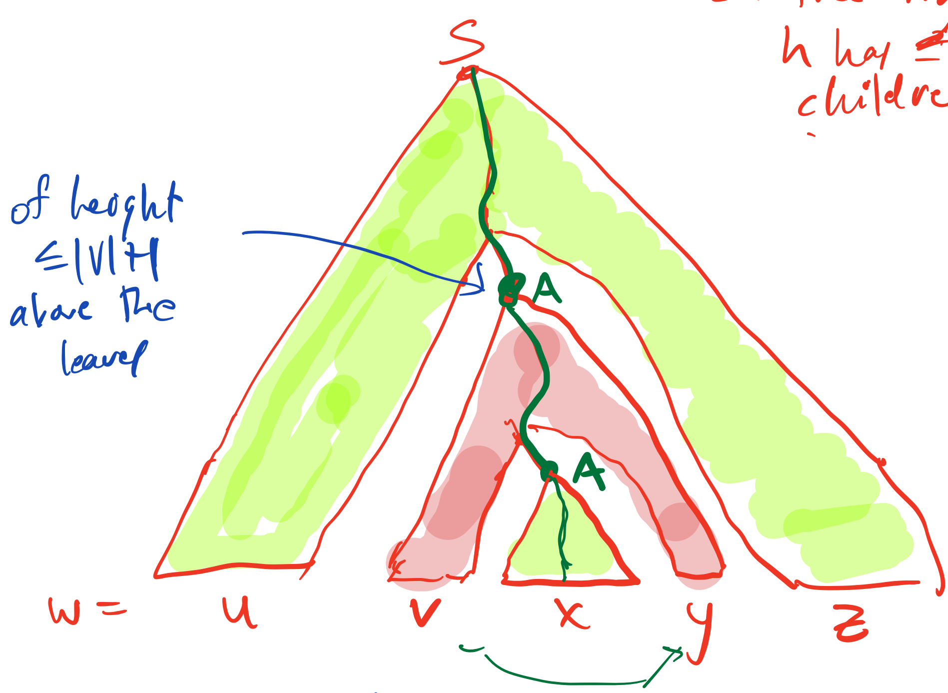


$\uparrow$ height $> |V|$ $\downarrow$

$\Rightarrow$ root-leaf path length $> |V|$ must contain repeated variable by the Pigeonhole principle. Choose lowest repeat.

Note: Since Chomsky Normal Form
- every leaf has a symbol in Σ
- parent of each leaf has 1 child
- every other internal node has 2 children

∴ tree with height h has $\leq 2^{h-1}$ children



Let $p = 2^{|V|}$.

- If $|w| \geq p$ then w requires parse tree height $\geq |V|+1$

$\Rightarrow$ parse tree has repeated variable on a path at height at most $|V|+1$ above leaves

Break up $w = uvxyz$ as in picture.

Since this is Chomsky normal form we must have either v or y (or both) non-empty since no unit rules

$\therefore$ ① is true

Since top A which generates vxy has height $\le |V|+1$

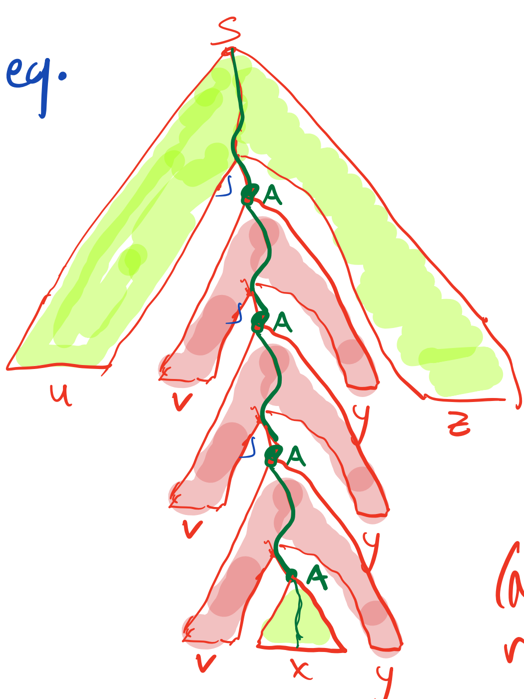
$\therefore |vxy| \le 2^{|V|} = p$

$\therefore$ ② is true

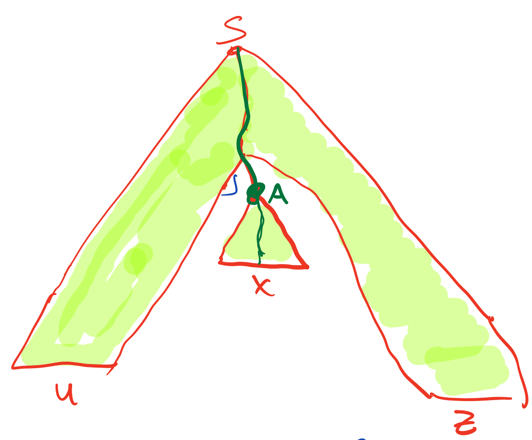
Finally, for ③, $w \in L$ is the case uv^1xy^1z, $i=1$.

We can repeat the red section that generates u and v any number of times

eg.



uv^3xy^3z



$uxz = uv^0xy^0z$

(Note if $v = \varepsilon$ we can rewrite $w = u_1 v_1 x_1 y_1 z_1$ where $u_1 = uvx$, $v_1 = y$, $x_1 = y_1 = \varepsilon$, $z_1 = z$) ∎

How to use this to show not context-free:

__Show__: $\forall p \exists w \in L$ $\forall$ ways of breaking up w into $uvxyz$, $\exists i$ s.t. $uv^ixy^iz \notin L$

---

e.g. $L = \{x\#x : x \in \{0,1\}^*\}$ is not a CFL

Let p be the pumping length for L

Consider $w = \underbrace{0^p1^p}_{1^{st}\text{ block}} \# \underbrace{0^p1^p}_{2^{nd}\text{ block}} \in L$

What are options for vxy as part of w?

only cases

- if vxy all in 1^{st} block:
  For all $i \neq 1$, $uv^ixy^iz \notin L$
  part before # won't match part after
- if vxy all in 2^{nd} block:
  Same as above since parts won't match
- if v or y contains #:
  For $i \neq 1$, have too many/few #
- if v in 1^{st} block and y in 2^{nd} block
  then since $|vxy| \leq p$,
  v has only 1's
  y has only 0's

and again # of 1's and 0's
won't match when
pumped $i \neq 1$.

$\therefore$ L is not a CFL

Another example: $L = \{0^n 1^n 2^n : n \geq 0\}$
is not a CFL

Similar idea:

Consider $0^p 1^p 2^p$:

Pumping either messes up the
order 0, 1, 2 or
only the number of one or two
of these symbols
will change
so don't get a string
in L.

We now move on to computational
complexity.

Time Complexity

Def^n The running time of a NTM M is the function $T: \mathbb{N} \to \mathbb{N}$ given by:

$T(n) = \max\{$ # steps M takes on any input $w \in \Sigma^*$ with $|w| = n$ & and any computation path that M may take on input w. $\}$

This gives the definition for both deterministic and nondeterministic TMs

---

Def^n For $T: \mathbb{N} \to \mathbb{N}$ define

NTIME$(T(\cdot)) = \{A$: there is a multitape NTM that decides A with running time that is $O(T(n))\}$

language

add these for the nondeterministic case

Note: text uses single tape, but multitape is used by researchers. more like other models.

Example: $A = \{x\#x : x \in \{0,1\}^*\}$

1-tape TM can't do this better

1-tape TM from 1st TM we produced running time $O(n^2)$ – need to shuttle back & forth

2-tape TM: copy part before # to tape 2
time $O(n)$ compare tapes 1 and 2

$\therefore A \in \mathrm{TIME}(n)$ linear time

Recall: Simulation of k-tape TM by 1-tape TM

k-tape TM $T(n)$ $\Rightarrow$ 1-tape TM $O(T^2(n))$

Best possible simulation even to go from 2-tapes to 1-tape because of above example

Can actually prove:

Hartmanis Lewis Stearns 1965

k-tape TM $T(n)$ $\Rightarrow$ 2-tape TM $O(T(n)\log T(n))$

Polynomial Time

Edmonds, Cobham 1965:
polynomial-time = good algorithm
more than polynomial-time = bad algorithm

<u>Def</u>

Def^n P polynomial time

$$P = \bigcup_k \mathrm{TIME}(n^k)$$

all languages that can be decided in time $O(n^k)$ for some constant k.

Def^n NP nondeterministic polynomial time

$$NP = \bigcup_k \mathrm{NTIME}(n^k)$$

Question: Does $P \overset{?}{=} NP$

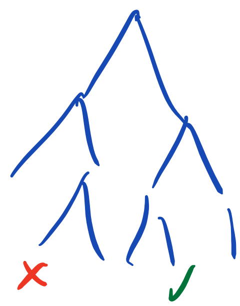
Cook 1971
Karp 1972

iron curtain

Levin 1973

Recall: NTM time $T(n)$ $\Rightarrow$ 1-tape TM time $2^{O(T(n))}$

$T(n)$



need to explore $2^{\Theta(T(n))}$ leaves

If $P=NP$ then one could get a vastly better simulation:

e.g. $(T(n))^k$ for some k for each language A (k might depend on A)