24 Feb 2025 - Theory of Computation - Week 08

Recap: Pumping Lemma for CFLs

Let A be a CFL. Then $\exists\, k > 0$ s.t. $\forall\, s \in A$ $\exists\, s =$
$|s| \geq k$

$uvwxy$ s.t the following holds

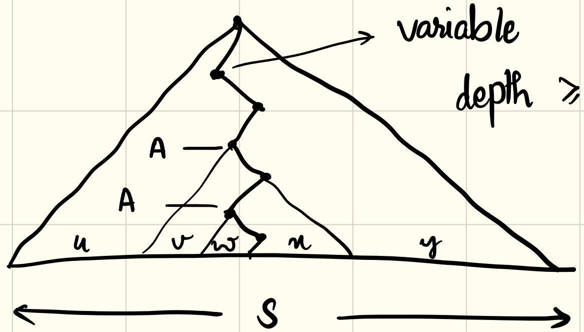
(i) $\forall\, i \geq 0$, $uv^i w x^i y \in A$

(ii) $vx \neq \varepsilon$

(iii) $|vwx| \leq k$

Proof idea: Assume CFL has a grammar in the Chomsky - Normal form. Let G be a CFG in CNF for A

$= (V, \Sigma, P, S)$



depth $\geq |V|+1$

if $|S|$ is large depth of parse tree is large

Pigeonhole

Contrapositive: If depth $\leq h$ $\Rightarrow$ length of derived string $\leq 2^h$

$vx \neq \varepsilon$ $\longrightarrow$ The grammar is in CNF

$|vwx| \leq k$ $\longrightarrow$ start from bottom of the parse tree.

Eg: $A = \{ 0^n 1^n 2^n \mid n \geq 0 \}$. Prove that A is not a CFL.

Proof: Proof by contradiction: Assume that A is a CFL.

This implies $\exists\ \ell > 0$ s.t. $\forall\ s \in A$ of $|s| \geq \ell$

$\exists\ s = uvwxy$ s.t

(i) $\forall\ i \geq 0$ $uv^i w x^i y \in A$ (ii) (iii)

Choose a string $\alpha \in A$ of sufficiently large length $(|\alpha| \geq \ell)$

We need to show that for all subdivisions $\alpha = uvwxy$

$\exists\ i \geq 0$ such that $uv^i w x^i y \notin A$

Choose $\alpha = 0^{\ell} 1^{\ell} 2^{\ell}$

Clearly $\alpha \in A$ and $|\alpha| = 3\ell \geq \ell$

$$\alpha = 0^{\ell} 1^{\ell} 2^{\ell}$$

$\because$ $|vwx| \leq \ell$: leftmost symbol 0 and rightmost 2 simultaneously not possible.

Case 1:

$$0^\ell$$

vwx

Choose $i = 2$ and uv^2wx^2y

only no. of zeroes $\uparrow \Rightarrow$ not in A

Similar argument for other cases.

Case 2:

$$\underbrace{0^\ell}_{vwx}\ 1^\ell\ 2^\ell$$

$\hookrightarrow$ only no. of zeroes and 1s change $\Longrightarrow$ not in A

e.g.: $B = \{ ww \mid w \in \{0, 1\}^* \}$

Assume B is a CFL

... same ...

Choose a good string to disprove

$\alpha = 0^l 1 0^l 1$

$\longrightarrow$ can be pumped

not a good choice.

$$\underbrace{0^l}_{u \mid v} \; \underbrace{1}_{w} \; \underbrace{0^l}_{x \mid y} \; 1$$

$\alpha = 0^l 1^l 0^l 1^l \rightsquigarrow$ similar cases.

Closure Properties of CFLs

Union

L_1 a CFL $\longrightarrow$ $G_1 = (V_1, \Sigma_1, P_1, S_1)$

L_2 another CFL $\longrightarrow$ $G_2 = (V_2, \Sigma_2, P_2, S_2)$

G_3

$$S_3 \rightarrow S_1 \mid S_2 \longrightarrow *$$

$$G_3 = (V_1 \cup V_2 \cup \{S_3\}, \ \Sigma_1 \cup \Sigma_1, \ P_1 \cup P_2 \cup *, \ S_3).$$

Concatenation

$$S_3 \rightarrow S_1 S_2$$

Star

L is a CFL.

$$S_3 \rightarrow S_1 S_3 \mid \varepsilon$$

Intersection

L_1, L_2 are CFLs. what about $L_1 \cap L_2$?

$$A = \{ 0^n 1^n 2^m \mid n, m \geq 0 \}$$

$$B = \{ 0^n 1^m 2^m \mid n, m \geq 0 \}$$

Intersection

$0^n 1^n 2^n \rightarrow$ not a CFL

Prove that A and B are CFL $\rightarrow$ use concatenation properties

Complement   Given L a CFL, what about $\bar{L}$?

$$\overline{\bar{A} \cup \bar{B}} = A \cap B$$

If $\bar{A}, \bar{B} \rightsquigarrow$ CFL $\rightarrow$ $A \cap B$ CFL $\Rightarrow\Leftarrow$

Given a CFG and a string $\rightarrow$ determine if the string can be generated by CFG $\Big\}$ Compilers $\rightarrow$ Wednesday

$\rightarrow$ Problems – Wednesday

Exam: focus more on Regular Languages

26 Feb 2025

Given a CFG $G = (V, \Sigma, P, S)$ decide

(i) empty ?

(ii) finiteness ?

Input size: encoding of G

empty $\Rightarrow$ never end in terminals

$$\left.\begin{array}{l} A \to a \\ B \to Aa \end{array}\right\} \to \text{Not empty}$$

Marked variables that has a rule $A \to \alpha$

$B \to \underset{\substack{\uparrow \\ \text{marked}}}{A}\,\underset{\substack{\nwarrow \\ \text{marked}}}{C}\,a \Rightarrow$ Mark B iteratively

Fix a CFG $G = (V, \Sigma, P, S)$ — Grammar is not an input

Problem: Given $x \in \Sigma^*$, decide if $x \in L(G)$?

$\boxed{\begin{array}{l} A \to BC \\ A \to a \end{array}}$

assume CNF

Input size: $|x|$

Length of the derivation $\leq 2|x| - 1$

??

$2^{O(|x|)}$

$S \rightarrow AB \mid BC$

$A \rightarrow BA \mid a$

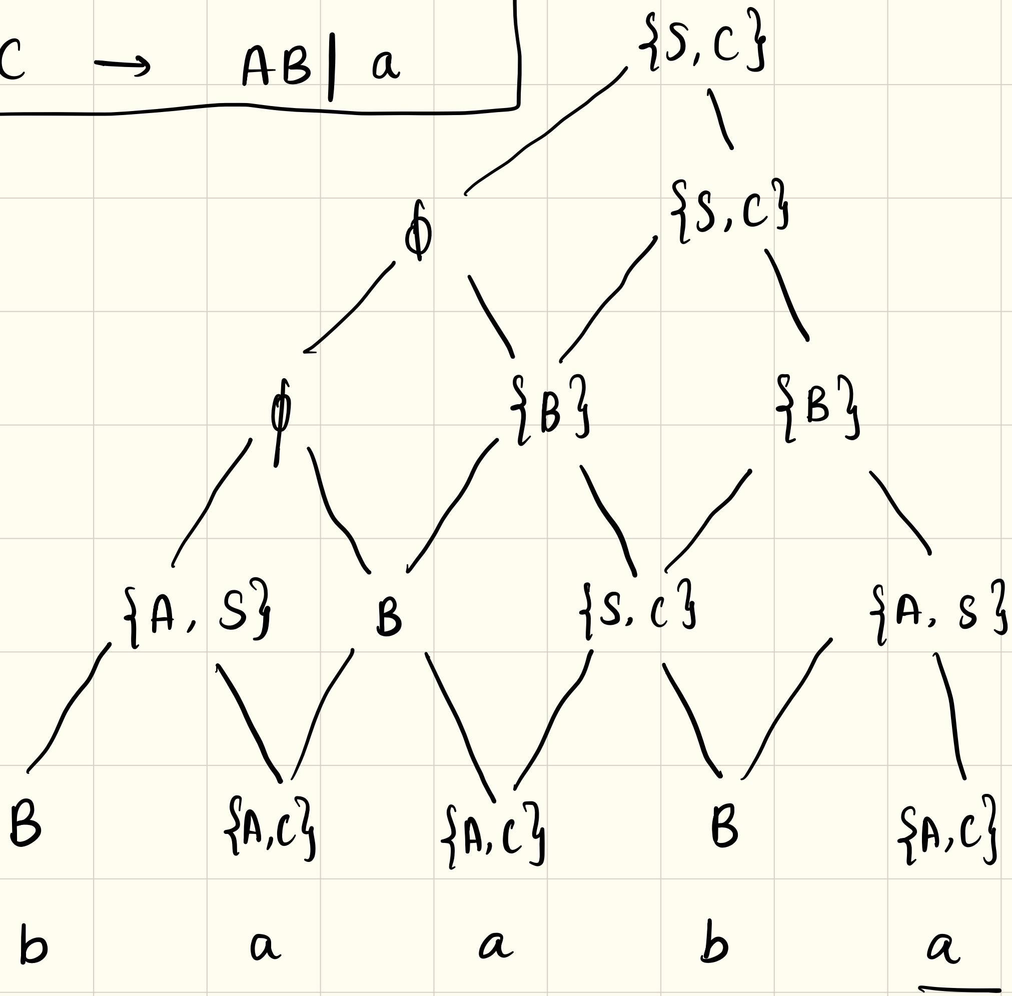
$B \rightarrow CC \mid b$

$C \rightarrow AB \mid a$

input is

b a a b a

Σi



ba

{BA|BC}

S A
S C
C A
C C

A A
A S
C A
C S

ϕ

Set of all vars that can produce a fixed string

V_{ij} := set of vars that can derive x_{ij}

$x_{ij} := x_i \, x_{i+1} \cdots x_{i+j-1}$

$1 \leq j, i \leq n$ (string of length j starting at i)

$V_{i1} \rightsquigarrow x_i$ (first step)

$V_{ij} = x_i \mid x_{i+1} \mid \cdots \; x_{i+j-1}$

$(j-1)$ parts

All possible ways of breaking it into two parts

$\underline{b}$ $\underline{a\ a}$

Not possible

$\underline{b\ a}$ a

Not possible

$\underline{a}$ $\underline{a\ b}$

$$1 \times n \quad + \quad 2 \times (n-1) \quad + \quad 3 \times (n-2)$$

$$+ \quad \cdots \quad + \quad n \times 1$$

---

27 Feb 2025

Quiz 1 discussion

3. $A = \Sigma^*$

Proof by induction on the length of strings.

$w = \underbrace{\_\_\_\_\_\_ x_{n-1}, x_n}_{n}$ $\qquad$ $w' = w_1 \ldots w_{n-1}$

$= xy$

$w = w' w_n$ $\qquad\qquad \dfrac{0\,0\,1}{y}$

$\quad = xy \; w_n$

If $w_n = 0 \longrightarrow$ same partition

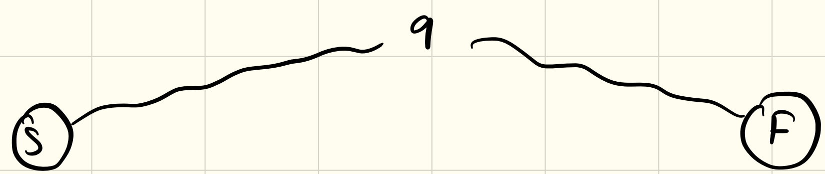
$\quad w_n = 1 \longrightarrow$ shift one symbol $(0$ or $1)$ to x

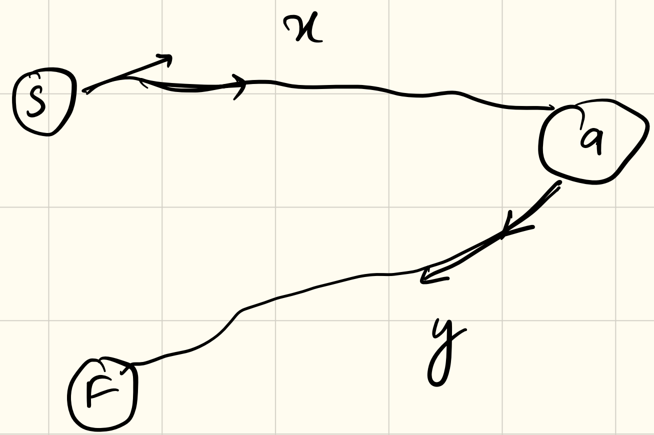
$0 \downarrow \; 0\uparrow$ for x $\qquad 1 \downarrow \; 1\downarrow$ for y

$\text{First half}(A) = \{ x \mid \exists y \;\; |x| = |y| \text{ and } xy \in A \}$

A is regular

DFA $\qquad M = (Q, \Sigma, \delta, s, F)$





y should reach F

x should reach Q

$$\begin{bmatrix} s \\ q \\ q \end{bmatrix} \xrightarrow{x_1} \begin{bmatrix} \delta(s, x_1) \\ q \\ \delta(q, a \end{bmatrix} \quad \forall a \in \Sigma$$