374 Lec 16 10/13/22

More dynamic programming

CYK = Cocke-Younger-Kasami.

$G = (V, T, P, S)$ = Context free grammar.

V: variables

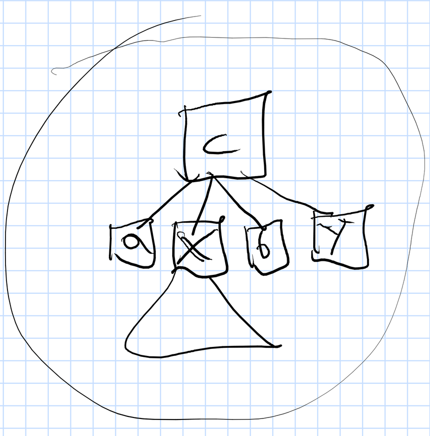
T: Terminals (Σ)

P: productions $V \to \text{string} \in (V \cup T)^*$

$\underbrace{C \to aXbY}$ | aXb

S: start variable

$L(S) = L(G)$: $S \to w_1 \to w_2 \to \dots \to w_k \in T^*$



$w = w_1 w_2 w_3 \dots w_n$ input string

G: CFG

Q: $w \in L(G)$?

$f(i, j, b)$ = true if the string $w_i w_{i+1} \dots w_j$ can be derived/produced from the variable R_b. $V = \{R_1, R_2, \dots, R_t\}$

G: Chomsky normal form

all productions have one of the following forms:

- $X \to YZ$
- $S \to \varepsilon$ ← only the start symbol.
- $X \to a$ $a \in T$

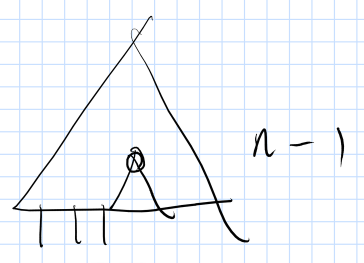
Bad grammar:

$S \to XY$

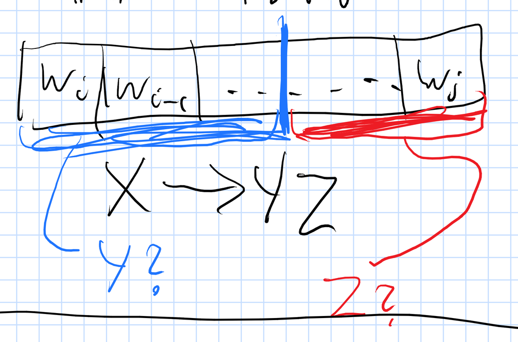
$X \to XXY \mid \varepsilon \mid a \mid \overline{XX}$

$Y \to aY \mid b$

$\overline{XX \dots Y}$



$R_\beta \to R_\gamma R_\delta$ β, γ, δ



$X \to YZ$

Y?

Z?

$f(i, j, b)$ $w_i \dots w_j$ be prod by R_b

Basis of recursion

If $i = j$ then return true $(R_b \to w_i) \in P$ otherwise return false.

$f(i, j, b)$:

for $k = i+1$ to j do

for any production $R_b \to R_c R_d$

if $f(i, k-1, c)$ and $f(k, j, d)$ then

return true

return false.

CFG has constant size

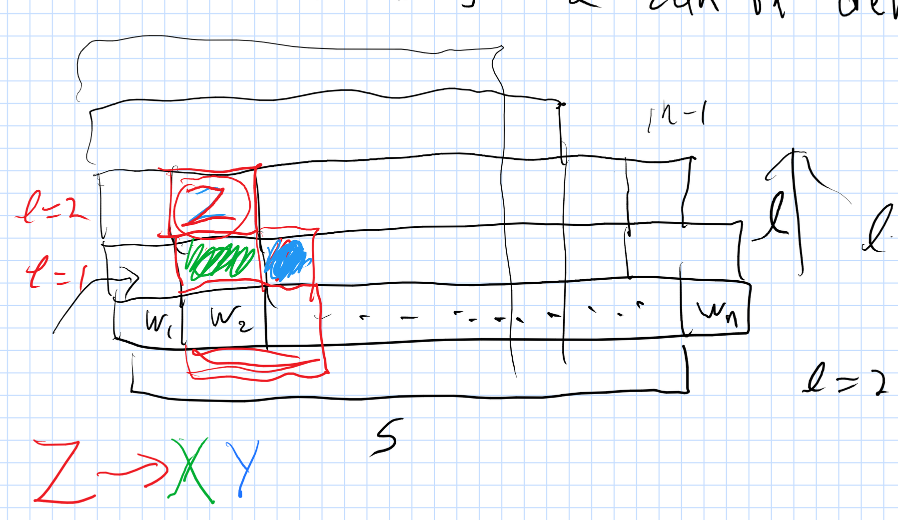
$g = |G|$ upper bound n # of variables

distinct recursive calls $n \cdot n \cdot g = O(n^2)$

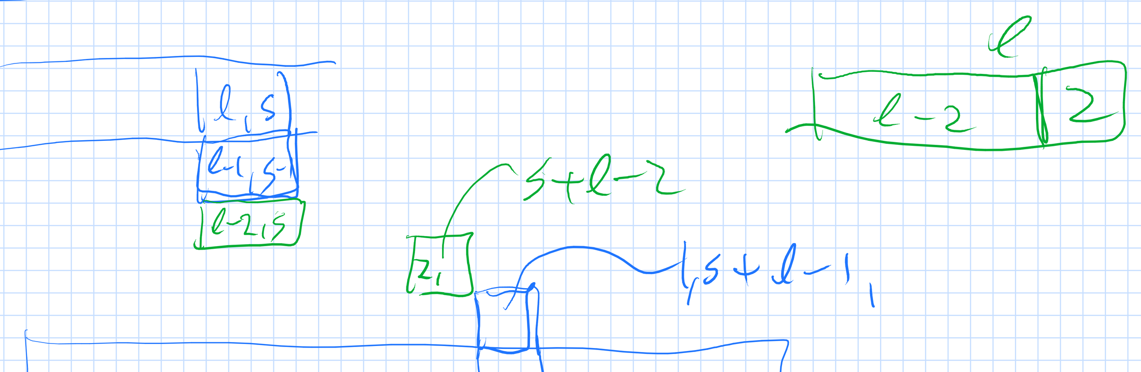
$O(n)$ work inside f (ignoring recursive calls)

$O(n^3)$: Running time using memoization

$g(\ell, s, b)$: string starting at s of length ℓ can be derived by R_b



$Z \to XY$



$R_1, R_2, \dots R_r$: variables in the grammar

$X = x_1 x_2 \dots x_n$

$P[n][n][r] = f(s, \ell, b) = 1 \iff w_s w_{s+1} \dots w_{s+\ell-1}$ can be produced by R_b

Init P to false.

for $i = 1$ to n do

for each $(R_j \to x_i) \in P$ do

$P[i, 1, j] \leftarrow$ True.

for $\ell = 2$ to n do

for $s = 1$ to $n - \ell + 1$ do

for $m = 1$ to $\ell - 1$ do (partition the string)

for $(R_\alpha \to R_\beta R_\gamma) \in P$ do

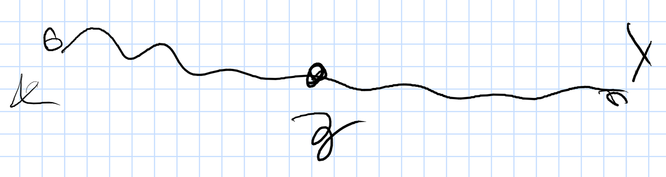
if $P[s, m, \beta]$ and $P[s+m, \ell-m, \gamma]$ then

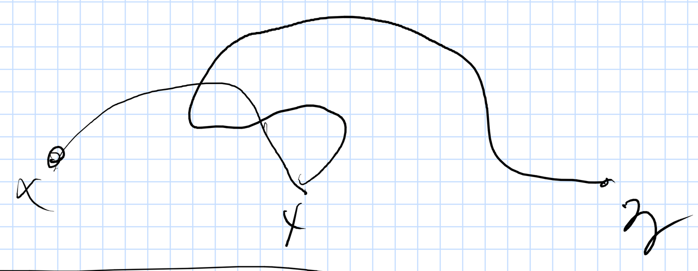
$P[s, \ell, \alpha] \leftarrow 1$

if $P[1][n][1]$ is True return "input string in L(G)"

else

return "no, nope, etc"





Longest common subsequence

$S[1 \dots m]$ $T[1 \dots n]$

~~united~~

~~disjoint~~

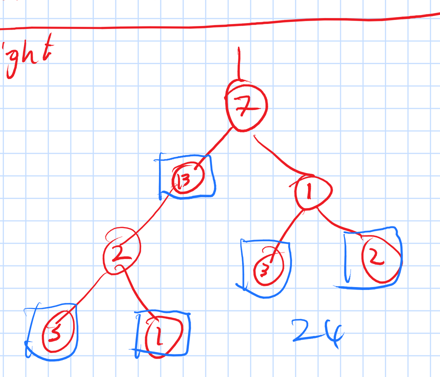
$S[1 \dots i]$ $T[1 \dots j]$

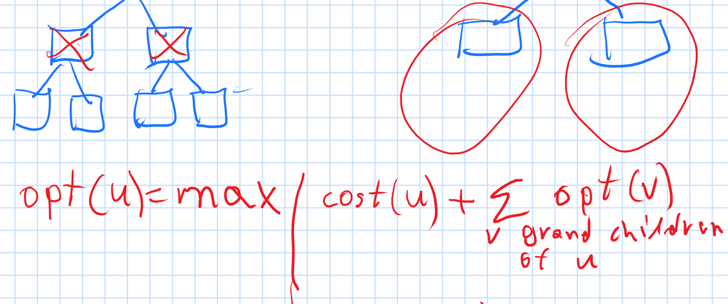
$$f(i,j) = \begin{cases} 0 & i=0 \text{ or } j=0 \\ \max(f(i, j-1), f(i-1, j)) & S[i] \neq T[j] \\ 1 + f(i-1, j-1) & S[i] = T[j] \end{cases}$$

$O(mn)$

Independent set in a tree

max weight indep set





$$opt(u) = \max \begin{cases} cost(u) + \sum_{v \text{ grand children of } u} opt(v) \\ \sum_{c \text{ child of } u} opt(c) \end{cases}$$

$O(n)$

post order traversal