Yet more Dyn Prog

Thursday, October 15, 2020 1:34 PM

Parsing Strings w/ DP (the CYK)

- Take a string, check if it can be generated by a given grammar (CFG)

$$\underline{S} := \begin{array}{|l} AB \\ eF \end{array} \qquad \underline{S} \rightsquigarrow^{*} \text{"helloworld"}$$

Sentence Parsing: "the dog | is | a great dane"
(parse diagrams)

the dog: Noun phrase (article | noun)
is: v
a great dane: noun phrase (adjective phrase)

Code parsing: for i in 1...n do {...}

for loop
i: iterator variable
1...n: range expression (1: begin, n: end)

Simplifying assumptions for today:

Grammars in Chomsky Normal Form (CNF)

i) $A \to \underline{\alpha}$ single nonterminal $\to$ single terminal

ii) $A \to \underline{B}C$ single nonterminal $\to$ two nonterminals

This is WLOG. (any CFG grammar can be converted to CNF)

ex. $(3+(5\times4)+2)$

$$\underset{\text{expression}}{E} := \begin{array}{|l} D \\ E+E \\ E\times E \\ (E) \end{array} \implies$$

$L := ($ $\quad R :=)$ $\quad P := +$ $\quad M := \times$

$D = 1, 2, 3 \ldots 9$

$E := L\cancel{ER}$ // (E)

$L E_2$

$E_2 := E R$

$E ::= E\,E_4$ // $E \times E$

$E_4 ::= M\,E$

$E ::= d$

$E ::= E\,\cancel{PE}$ // $E+E$

$E\,E_3$

$E_3 ::= P\,E$

Given $X[1...n]$, grammar $G = (\Sigma, N, P, S)$ — P: rules, N: nonterminals, S: starting nonterminal

$Parse(i, j, M)$ = denotes whether we can parse $X[i...j]$ as M — $M \rightsquigarrow^* X[i...j]$

if $i = j$:

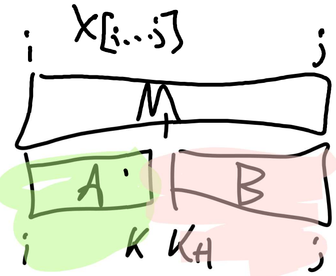
True if $\exists$ rule $M \to \alpha \in P$, $\alpha = X[i]$

False otherwise.

(base case — rule form i)

otherwise:

OR symbol: $\bigvee_{\substack{i \le k < j,\\ M \to AB \in P}} \left\{ Parse(i, k, A) \text{ and } Parse(k+1, j, B) \right.$



$M \rightsquigarrow^* X[i...j]$

ii) $M \to AB$

# of subproblems? $O(n^2 |N|)$

Time to compute each subproblem? $O(n|P|)$

total est: $O(n^3 |P||N|)$ ← By looping in a smart way, only takes $O(n^3|P|)$

Size of table: $n \times n \times |N|$

Example:

$E ::= L\,\cancel{ER}$ // (E)

$L\,E_2$

$E_2 ::= E\,R$

$E ::= E\,E_4$ // $E \times E$

$E_4 ::= M\,E$

$E ::= d$

$E ::= E\,\cancel{PE}$ // $E+E$

$E\,E_3$

$E_3 ::= P\,E$