

University of Waterloo
CS 462 — Formal Languages and Parsing
Winter 2020
Problem Set 10

Distributed Monday, March 23 2020. Due Friday March 27 2020 at 5 PM. Submit via LEARN.

1. [10 marks] Using the CYK algorithm, find two different parse trees for the word *abaabb* in the following grammar:

$$\begin{aligned} S &\rightarrow AC \\ A &\rightarrow AA \mid AB \mid a \\ B &\rightarrow b \\ C &\rightarrow AC \mid BB \mid b \end{aligned}$$

Show the table in the algorithm.

2. [10 marks] Consider the following grammar:

$$\begin{aligned} S &\rightarrow T \\ T &\rightarrow aBT \mid bAT \mid c \\ A &\rightarrow a \mid bAA \\ B &\rightarrow b \mid aBB \end{aligned}$$

Is it LR(0)? Prove or disprove.

3. [10 marks] Let G be a fixed context-free grammar in Chomsky normal form. Show how to determine, given a string $w \in L(G)$, the *number* of distinct parse trees for w in G . Your algorithm should run in time polynomial in $|w|$ in the “log-cost” model, where you take into account the sizes of the numbers involved. In this (slightly unrealistic) model, adding and multiplying two n -bit numbers costs $O(n)$ time. (I don’t expect a really detailed analysis; just explain why everything runs in polynomial time.) Hint: modify the CYK algorithm.