CSCI 338: Assignment 5 (6 points)

Elliott Pryor

April 3, 2020

This assignment is due on **Monday, April 20, 11:30pm**. It is strongly encouraged that you use Latex to generate a single pdf file and upload it under *Assignment 5* on D2L. But there will NOT be a penalty for not using Latex (to finish the assignment). This is **not** a group-assignment, so you must finish the assignment by yourself.

Problem 1

We are given 5 matrices $M_1, ..., M_5$, their dimensions (i.e., rows by columns) are as follows: M_1 is 15 $\times$ 20, M_2 is 20 $\times$ 30, M_3 is 30 $\times$ 10, M_4 is 10 $\times$ 50, and M_5 is 50 $\times$ 8.

(1.1) Run the dynamic programming algorithm for *matrix chain multiplication* that we covered in class to produce the table $m[-,-]$.

i \ j	1	2	3	4	5
1	0	9000	9000	165000	13600
2	X	0	6000	16000	11200
3	X	X	0	15000	6400
4	X	X	X	0	4000
5	X	X	X	X	0

Table 1: Solution to 1.1

(1.2) What is the optimal solution value? Where do you find it?

The optimal number of multiplications is 13600. It is located at m[1,5] in the top right corner of the table.

Problem 2

We are given a context-free grammar G as follows:

G: $S \to AS|SB$

$A \to AD|DA|a$

$B \to BB|BD|b$

$D \to DD|d$.

We are also given a string $w = bdbdd$.

(2.1) Run the dynamic programming algorithm for A_{CFG} that we covered in class to produce the table $table[-,-]$.

i \ j	1	2	3	4	5
1	B	B	B	B	B
2	X	D	$\emptyset$	$\emptyset$	$\emptyset$
3	X	X	B	B	B
4	X	X	X	D	D
5	X	X	X	X	D

Table 2: Solution to 1.1

(2.2) How do we know whether G generates w from the table?

G generates w if S is in the top right corner (table[1,5]).

So in this case, G cannot generate w since $S \notin table[1, 5]$

Problem 3

Show that $ALL_{DFA} \in$ P.

Proof.

We construct a decideer for ALL_{DFA} that runs in $O(n^k)$ time. We know that ALL_{DFA} accepts $\sum^*$ iff we arrive at an accept state in every possible configuration. So we construct the decider S as follows:

S: $< D >$ where D is some DFA:

1. Perform Breadth First Search on D starting at the start state q_0
2. Test if only accepting states were visited.
3. If the test returns true $\rightarrow$ accept
   Otherwise $\rightarrow$ reject

Clearly S decides ALL_{DFA} as S visits all possible states and if S accepts each one is an accepting state, so $L(D) = \sum^*$. If S rejects then there is some non-accepting state that can be reached from q_0 so $L(D) \neq \sum^*$ because it can arrive at some non-accepting state.

Also, S runs in polynomial time as BFS runs in polynomial time. Therefore, we have found a decider for ALL_{DFA} that runs in polynomial time. And by definition, $ALL_{DFA} \in P$

$\square$

Problem 4

Show that Independent Set $\in$ NP.

Proof.

We construct a non-deterministic polynomial time turing machine S to decide Independent Set.

S: $< G, k >$ where G is a graph $G = (V, E)$ and k is an integer st. $k \leq |V|$

1. Non-deterministically select some subset c of k verticies from G
2. Test whether G contains any of the $\binom{k}{2}$ edges connecting nodes in c
3. If test returns true, then G has an edge connecting some $u, v \in c \to$ Reject
   Otherwise, G has no edges connecting $u, v \in c \to$ Accept

S runs in $O(\binom{k}{2}) = O(n^2)$ time. So we have found a non-deterministic decider that runs in polynomial time. So Independent Set $\in NP$ by Theorem 7.20

$\square$