CSCI 338: Assignment 5

Nathan Stouffer

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 $m[-,-]$.

We give the table $m[-,-]$:

	1	2	3	4	5
1	0	9000	9000	21000	13600
2	–	0	6000	16000	11200
3	–	–	0	15000	6400
4	–	–	–	0	4000
5	–	–	–	–	0

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

The optimal solution value is 136000 multiplications. This value can be found in the top right corner of the table at $m[1,5]$.

Problem 2

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

G: $S \longrightarrow AS \mid SB$
$A \longrightarrow AD \mid DA \mid a$
$B \longrightarrow BB \mid BD \mid b$
$D \longrightarrow DD \mid d$
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 $table[-,-]$.

We give now fill the entries of $table[-,-]$:

	1	2	3	4	5
1	B	B	B	B	B
2	–	D	$\emptyset$	$\emptyset$	$\emptyset$
3	–	–	B	B	B
4	–	–	–	D	D
5	–	–	–	–	D

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

We know whether G generates w by the following rule:

$$w \in L(G) \iff S \in table[1,5]$$

Since $S \notin table[1,5]$ we know that G does not generate w.

Problem 3

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

We first define $ALL_{DFA} := \{\langle D \rangle \mid D$ is a DFA and $L(D) = \Sigma^*\}$.

From Theorem 4.4, we know that $E_{DFA} := \{\langle A \rangle \mid A$ is a DFA and $L(A) = \emptyset\}$ is decidable by the following Turing Machine.

$T =$ "On input $\langle A \rangle$ a DFA:

1. Mark the start state of A

2. Repeat until no new states get marked:

3. Mark any state that has a transition coming into it from any state that is already marked

4. If no accept state is marked, *accept*; otherwise, *reject* "

Additionally, we can construct the complement of a DFA by reversing accept/nonaccept states in the DFA. We now give a Turing Machine A to decide ALL_{DFA}.

$A =$ "On input $\langle D \rangle$ a DFA:

1. Construct a DFA E where $L(E) = \overline{L(D)}$

2. Run TM T on input $\langle E \rangle$

3. If T accepts, *accept*

4. If T rejects, *reject* "

In words, A constructs the DFA that is the complement of D. It then tests if the complement's language is empty. If so, the original language must be Σ^*.

So A certainly decides ALL_{DFA}, but does it do so in polynomial time? Let s be the number of states for the input DFA D, then the number of transitions in D is $|\Sigma| * s$. Line 1 of A runs in $O(s)$ time. Lines 3 and 4 run in $O(1)$ time. The running time of line 2 is the same as the running time of T.

Lines 1 and 4 of T run in $O(1)$ time while line 3 runs in $O(|\Sigma| * s)$ time (the number of edges). But $|\Sigma|$ is a constant so $O(|\Sigma| * s) = O(s)$. In the worst case scenario, line 2 repeats s times so T is a $O(1) + s * O(s) + O(1) = O(s^2)$ machine.

This brings us back to machine A, which we now know to be a $O(1) + O(s^2) + O(1) + O(1) = O(s^2)$ machine. Since ALL_{DFA} has a polynomial time decider, it must be true that $ALL_{DFA} \in P$.

Problem 4

Show that Independent Set $\in$ NP.

We first define the problem:

$$IS := \{\langle G, k\rangle \mid G = (V, E) \text{ is a graph and } k \leq |V| \text{ where there exists an independent set } V' \subset V \text{ with } |V'| \geq k\}$$

We must also define an independent set of vertices. This is a set $V' \subset V$ such that for all $u, v \in V'$ it is true that $(u, v) \notin E$.

Note that a language A is a member of NP if A has a polynomial verifier. This is true if and only if A is decidable by some nondeterministic Turing Machine in polynomial time. We now give a nondeterministic Turing Machine S to decide IS.

$S =$ "On input $\langle G, k\rangle$ where $G = (V, E)$ a graph and $k \in \mathbb{N}$:

1. Nondeterministically select a subset of vertices V' such that $|V'| \geq k$
2. For each element $(u, v) \in V' \times V'$
3. If $(u, v) \in E$, *reject*
4. Test has been passed, *accept* "

Certainly S nondeterminstically decides IS, but does it run in polynomial time? Yes, in fact, it runs in $O(n^2)$ time, where $n = |V'|$. So we have a polynomial, nondeterminstic decider for IS, therefore $S \in NP$.