Indian Institute of Technology Kharagpur
Department of Computer Science and Engineering

Foundations of Computing Science (CS60005) **Autumn Semester, 2021-2022**

Test 3 [Marks = 30] **Date: 17-Nov-2021, 08:15 – 09:30** **Venue: Online**

There are ALL THREE questions. State all assumptions you make. Be brief and precise.

1. (a) Prove or disprove: $\{\mathcal{M} \mid \mathcal{M}$ is a DTM that runs in time $O(n^3)\}$ is undecidable. [6]

Solution: Let $L = \{\mathcal{M} \mid \mathcal{M}$ runs in time $O(n^3)\}$. We show that L is undecidable via a reduction from HP. Let $\mathcal{N}, x$ be an instance of HP. Construct $\mathcal{M}$ such that on input y, it does the following.

- Let $n = |y|$.
- Run $\mathcal{N}$ on x for n steps.
- If $\mathcal{N}$ halts within n steps, run n^3 arbitrary steps and halt.
- Otherwise, run n^4 arbitrary steps and halt.

Suppose that $\mathcal{N}$ halts on x in n_0 steps. Then $\forall y$ with $|y| = n \geq n_0$, $\mathcal{M}$ halts in n^3 steps. So, $\mathcal{M} \in L$. If $\mathcal{N}$, does not halt on x, then $\mathcal{M}$ runs for n^4 steps before halting. That is, $\mathcal{M} \notin L$. Since HP is undecidable, so is L.

(b) For a set $A \subseteq \Sigma^*$ (with $|\Sigma| \geq 2$), define $A^{\mathbf{R}} = \{w^{\mathbf{R}} \mid w \in A\}$ where $w^{\mathbf{R}}$ denotes w reversed. Is it decidable, for a given TM $\mathcal{M}$, whether $L(\mathcal{M}) = L(\mathcal{M})^{\mathbf{R}}$? Justify your answer. [4]

Solution: Let $\mathsf{REV} = \{\mathcal{M} \mid L(\mathcal{M}) = L(\mathcal{M})^{\mathbf{R}}\}$. We show that REV is undecidable via a reduction from $\neg$HP. Let $(\mathcal{M}, x)$ be an instance of $\neg$HP. Construct a TM $\mathcal{N}$ over Σ with $|\Sigma| > 1$ that on input y, does the following.

- Run $\mathcal{M}$ on x.
- If $\mathcal{M}$ halts and $y = a_1a_2$, then accept and halt. (Here $a_1, a_2 \in \Sigma$ and $a_1 \neq a_2$).
- Reject otherwise.

Now, if $(\mathcal{M}, x) \in \neg$HP i.e., $\mathcal{M}$ does not halt on x, then $L(\mathcal{N}) = \emptyset$ and trivially $L(\mathcal{N}) = L(\mathcal{N})^{\mathbf{R}}$. Otherwise, $L(\mathcal{M}) = \{a_1a_2\}$. Since $(a_1a_2)^{\mathbf{R}} = a_2a_1 \notin L(\mathcal{N})$, $L(\mathcal{N}) \neq L(\mathcal{N})^{\mathbf{R}}$. Hence REV is undecidable.

Alternate solution using Rice's theorem. Let P be a property on *r.e.* sets defined as

$$P(A) = \begin{cases} T & \text{if } A = A^{\mathbf{R}} \\ F & \text{otherwise} \end{cases}$$

Again, we consider languages over Σ of size > 1 for otherwise the problem is trivially decidable. Trivially, $P(\emptyset) = T$. Also $P(\{a_1a_2\}) = F$ for some set $a_1, a_2 \in \Sigma$ with $a_1 \neq a_2$ since $(a_1a_2)^{\mathbf{R}} = a_2a_1 \notin L(\mathcal{N})$. We have exhibited two sets, one for which the P holds and the other for which it does not. It follows that P is a non-trivial property and hence undecidable, by Rice's theorem.

2. Answer true or false. Justify your answer.

(a) All **NP**-Hard problems are decidable. [2]

Solution: False.

Halting problem is undecidable but is **NP**-hard.

(b) If L_1, L_2 are **NP**-Complete, then so is $L_1 \cup L_2$. [3]

Solution: False. We construct a counter example using the following two **NP**-Complete sets.

$$\mathsf{HAMPATH} = \{G \mid G \text{ is an undirected graph with a Hamiltonian path}\}$$

$$\mathsf{SAT} = \{\phi \mid \phi \text{ is a satisfiable CNF formula}\}$$

Let

$$L_1 = \{(G, \phi) \mid G \in \mathsf{HAMPATH}, \phi \text{ is any CNF formula}\},$$

and

$$L_2 = \{(G, \phi) \mid G \text{ is an undirected graph}, \phi \in \mathsf{SAT}\}.$$

Clearly L_1, L_2 are both **NP**-Complete, whereas

$$L_1 \cup L_2 = \{(G, \phi) \mid G \text{ is an undirected graph}, \phi \text{ is a CNF formula}\}$$

is not. It can be decided in poly-time.

(c) If every **NP**-hard language is **PSPACE**-hard, then **PSPACE** = **NP**. [2]

Solution: We know that $\mathbf{NP} \subseteq \mathbf{PSPACE}$. Let L be any **NP**-complete language. L is **NP**-hard implies that **PSPACE**-hard. For every $L' \in \mathbf{PSPACE}$, $L' \leq_p L$. Since $L \in \mathbf{NP}$, we can conclude that $L' \in \mathbf{NP}$ and hence $\mathbf{PSPACE} \subseteq \mathbf{NP}$. Therefore $\mathbf{NP} = \mathbf{PSPACE}$.

(d) $\mathbf{polyL} \neq \mathbf{polyNL}$,
where $\mathbf{polyL} = \cup_{c>0}\mathbf{DSPACE}(\log^c n)$ and $\mathbf{polyNL} = \cup_{c>0}\mathbf{NSPACE}(\log^c n)$. [3]

Solution: False.

The two classes are equal. Follows from Savitch's theorem.

3. (a) Consider a variant M-3SAT (*minimal* 3SAT) of 3SAT defined as follows: a 3 CNF Boolean formula ϕ is in M-3SAT if there exists atleast one satisfying assignment for ϕ which makes exactly one literal in every clause true. Show that M-3SAT is **NP**-Complete. [5]

Solution: Suppose that ψ is a "yes" instance of M-3SAT. Any satisfying assignment satisfying the constraint forms a poly-sized certificate for ψ that can be verified in detterministic polynomial time. Therefore, M-3SAT $\in$ **NP**.

Suppose that ϕ is an instance of 3SAT defined over n variables and containing m clauses. Construct a formula ψ over $n + 4m$ variables containing $3m$ clauses as follows. For every clause $x_i \vee x_j \vee x_k$ of ϕ, include clauses $(x_i \vee w_1 \vee w_2), (\neg x_j \vee w_1 \vee w_3), (\neg x_k \vee w_2 \vee w_4)$ in ψ, where w_1, w_2, w_3, w_4 are new variables.

$\phi \in$ 3SAT $\implies \psi \in$ M-3SAT**:** Consider any satisfying assignment for ϕ. Suppose that at least 2 literals are true in the clause $x_i \vee x_j \vee x_k$. Without loss of generality, assume $x_i = 1, x_j = 1, x_k = 0$. In that case setting $w_1 = 0, w_2 = 0, w_3 = 1, w_4 = 0$ results in a satisfying assignment for the group of clauses $(x_i \vee w_1 \vee w_2), (\neg x_j \vee w_1 \vee w_3), (\neg x_k \vee w_2 \vee w_4)$ with every clause containing exactly one true literal. Next, consider the case $x_i = 1, x_j = 1, x_k = 1$. We can set $w_1 = 0, w_2 = 0, w_3 = 1, w_4 = 1$ to ensure that every clause in the group has exactly one true literal.

$\psi \in$ M-3SAT $\implies \phi \in$ 3SAT**:** Consider the satisffying assigment of ψ which makes exactly one literal true in every clause and consider the group of clauses $(x_i \vee w_1 \vee w_2), (\neg x_j \vee w_1 \vee w_3), (\neg x_k \vee w_2 \vee w_4)$. It is not possible that $x_i = 0, \neg x_j = 1, \neg x_k = 1$ for otherwise either w_1 or w_2 must be true in order to make the first clause true. But making any of them true results in more than one literal in the one of the remaining clauses to be 1, contradicting our choice of assignment. So atleast one of x_i, x_j, x_k is 1 thus satisfying the clause $(x_i \vee x_j \vee x_k)$ of ϕ. Therefore $\phi \in$ 3SAT.

(b) Let A, B be two $n \times n$ Boolean matrices. Entries of the *Boolean product* $C = A \cdot B$ are defined as $C_{ij} = \vee_{k=1}^{n}(a_{ik} \wedge b_{kj})$ for $1 \leq i, j \leq n$. Describe an $O(\log n)$-space procedure to compute $A \cdot B$ given A, B. [5]

Solution:

Maintain 3 counters i, j, k, all initialised to 1.
Initialise two bits e, d (to 0).
for $i = 1, \ldots, n$
    for $j = 1, \ldots, n$
        $e = 0$
        for $k = 1, \ldots, n$
            Compute $d = a_{ik} \wedge b_{kj}$
            Set $e = e \vee d$
        set $c_{ij} = e$ and write it on the output tape

The TM needs space to store i, j, k plus 2 bits (for d, e). Since $i, j, k \leq n$, the space required to store them is $O(\log n)$.