Practice Problem Set 13

Released: Sunday November 3, 2019

1. Show that the following subsets of $\mathbb{R}$ are countably infinite:

   (a) $\{2^q : q \in \mathbb{Q}^+\}$

   Solution: Consider the one-to-one mapping $f : N^2 \to N$ as discussed in class. We give one-to-one mappings from both $S = \{2^q : q \in \mathbb{Q}^+\}$ to N and from N to S and then use the result stated in class to show this implies a bijection between S and N.

   Consider the mapping $f_1 : S \to N^2$ such that $f_1(2^{p/r}) = (p, r)$ where $p, r \in N$ and $\gcd(p, r) = 1$. It is easy to see that this is a one-to-one mapping from S to N^2 and $f \circ f_1$ is a one-to-one mapping from S to N. Consider $f_2 : N \to S$ as $f_2(n) = 2^n$ which is also a one-to-one mapping.

   (b) $\{x \in \mathbb{R} : x > 0 \wedge x^2 \in \mathbb{Q}\}$

   Solution: The solution approach is similar to (a).

2. Show that the following sets have the same cardinality as $\mathbb{R}$.

   (a) $(0, \infty)$

   Solution: Consider the one-to-one mapping $f_1 : (0, \infty) \to \mathbb{R}$ with $f_1(x) = x$. Also, $f_2 : \mathbb{R} \to (0, \infty)$ given by $f_2(x) = e^x$ is also a one-to-one mapping.

   (b) $[0, 1] \cup \mathbb{Z}$

   Solution: The identity map is a one-to-one map from the given set $S = [0, 1] \cup \mathbb{Z}$ to R. Also, consider the one-to-one mapping f_2 from R to $(0, \infty)$ as given in the last part. Now consider $f_3 : (0, \infty) \to (0, 1)$ as $f_3(x) = e^{-x}$ which is again a one-to-one mapping. Hence, $f_3 \circ f_2$ is a one-to-one mapping from R to $(0, 1)$.

   (c) $\mathbb{R} \times \mathbb{R}$ (*Hint: Flesh out the proof from the lecture.*)

   Solution: There are 3 parts to the proof (T is defined similarly as the lecture notes) -

   (i) $|T| = |\mathbb{R}|$. A one-to-one mapping from T to $\mathbb{R}$ was provided in the slides. Consider the one-to-one mapping f from $\mathbb{R}$ to $(0, 1)$ as proved in the last part. Define a mapping $g : (0, 1) \to T$ which on input $x \in (0, 1)$ computes the binary representation of x. This representation looks like $0.s$ where $s \in T$ and $g(x) = s$. It is easy to see that g is a one-to-one mapping and hence $g \circ f$ gives a one-to-one mapping from $\mathbb{R}$ to T.

   (ii) $|T^2| = |T|$. Consider the one-to-one mapping $f : T \to T^2$ with $f(s) = (s, s)$. Consider $g : T^2 \to T$ which takes 2 binary strings s_1 and s_2 as input and outputs $s_1[0]s_2[0]s_1[1]s_2[2]s_1[3]s_2[3]$.... (where $s[i]$ represents i^{th} bit of s) which is again a one-to-one mapping.

   (iii) $|T^2| = |\mathbb{R}^2|$. Consider the bijection $f : T \to R$ from (i) and let $g : T^2 \to R^2$ with $g(u, v) = (f(u), f(v))$ which is again a bijection.

3. Suppose for each $i \in \mathbb{N}$, S_i is a countable set. Using the result that $\mathbb{N} \times \mathbb{N}$ is countable, prove that the following sets are countable (but not necessarily infinite). Be sure to handle any special cases (e.g., some $S_i = \emptyset$). You may use the fact that S is countable iff $|S| \leq |\mathbb{N}|$.[1]

   (a) $\bigcup_{i \in \mathbb{N}} S_i$

   Solution: Let $S = \bigcup_{i \in \mathbb{N}} S_i$. Consider $g : S \to N$ which takes an element x in S and outputs the minimum $i \in N$ such $x \in S_i$. Let $f_i : S_i \to N$ be a one-to-one mapping. Such a mapping exists because S_i is countable. Consider $f : S \to N^2$ given by $f(x) = (g(x), f_{g(x)}(x))$. Check that this is a one-to-one mapping. Then use the one-to-one mapping from N^2 to N to get a mapping from S to N.

   (b) For any $k \in \mathbb{N}$, $S_0 \times \cdots \times S_k$

   Solution: First show that $A \times B$ is countable if both A and B are countable. Then use induction on k to show that $S_0 \times \cdots \times S_k$ is countable for any k.

[1] In class, we defined a set S as countable if S is finite or $|S| = |\mathbb{N}|$. By the "axiom of countable choice" this is equivalent to $|S| \leq |\mathbb{N}|$.

4. Let A, B and C be sets such that $|A| \leq |B|$ and $B \subseteq C$. Prove that $|A| \leq |C|$.

   Solution: Consider the injection $f : A \to B$. Since $B \subseteq C$, consider $g : A \to C$ such that $g(x) = f(x)$ which is again an injection.

5. $\mathbb{Z}[X]$ denotes the set of all polynomials in a variable X, with integer coefficients. Show that $\mathbb{Z}[X]$ is countable. *Hint: Prove first that for every integer $n \geq 1$ the set P_n of all polynomials of degree $\leq n$ with integer coefficients is countable. Then use the fact that the union of countably many countable sets is countable.*

   Solution: Consider the set P_n of all polynomials of degree $\leq n$ with integer coefficients. Consider the mapping $f : P_n \to \mathbb{Z}^{n+1}$ which takes a polynomial $a_0 + a_1X + a_2X^2 + ...a_nX^n$ with $a_0, a_1, ..., a_n \in \mathbb{Z}$ and outputs $(a_0, a_1, a_2, ..., a_n)$. Show that f is a bijection. Since $\mathbb{Z}^{n+1}$ is countable (cross product of finitely many countable sets is countable), there is a bijection from $\mathbb{Z}^{n+1}$ to N using which a bijection from P_n to N can be given. Since $\mathbb{Z}[X] = \bigcup_{n \in \mathbb{N}} P_n$ and using the result from 3.(a), we can show that $\mathbb{Z}[X]$ is countable.

6. Consider a machine which has n bits of memory (initialized to all 0s). Its state is determined by the contents of its memory. This machine takes inputs from the set $\Sigma = \{1, \ldots, n\}$: On input i, it toggles its i^{th} bit. (Toggling a bit changes its value from 0 to 1 or from 1 to 0.)

   (a) Write down the transition function for this machine for $n = 2$, as table, with columns "current state," "input" and "next state." (Note that each state is labeled with 2 bits.)
   Solution: The following is the transition function. (we take 1 as the least significant digit, although the other way is also fine).

current state	input	next state
00	1	01
00	2	10
01	1	00
01	2	11
10	1	11
10	2	00
11	1	10
11	2	01

   (b) What graph does the state-diagram for this machine (for a general value of n) resemble? You can treat a pair of directed edges between two states, but pointing in opposite directions, as a single undirected edge. Justify your answer. *Hint: For this part, it may be helpful to draw a diagram for the case $n = 2$ from the previous part.*

   Solution: An n-dimensional hyper cube. The state transition diagram is hyper cube with vertices coordinates representing memory state. It is known that every edges in a Hyper cube joints vertices which differ by exactly one bit. In the state-diagram this edge is a diagram with input of the position of the differing bit between the vertices. This is because, every state is connected to other states only if they differ by a single bit, which is the property of a Hyper cube.

7. Give deterministic finite-state acceptors for the following languages over the alphabet $\{0, 1\}$. In each case, write down its transition function (in the form of a table) and also draw the state diagram.

   (a) $\{x \mid x$ is non-empty and begins and ends with the same symbol$\}$

   (b) $\{x \mid x$ does not contain the substring 11$\}$

   (c) $\{x \mid x$ number of zeroes is even and number of ones is odd$\}$

   Solution: For each of the deterministic acceptor, it is easier to first construct it diagramatically and then write the acceptor table.

   One possible set of answers are attached in moodle.

8. Give non-deterministic finite-state acceptors for the following languages over the alphabet $\{0, 1\}$. In each case, write down its transition function (in the form of a table) and also draw the state diagram.
   Solution: Same as 7.

   (a) $\{x \mid x$ ends in 00 or 010 $\}$

(b) $\{x \mid x$ contains the substring 11$\}$

9. It was mentioned in class that the language consisting of *all* palindromes does not have a finite-state acceptor. In this problem we consider palindromes of a fixed length.

   (a) Describe, as explicitly as you can, a deterministic finite-state acceptor for the language consisting of all palindromes of length exactly 100. *Hint: The first 50 input bits need to be memorized. Arrange the states in a binary tree for this.*

   Solution: To memorize the first 50 bits, we use a complete binary tree of height 50. At each position $i, 1 \leq i \leq 50$, take left if the i^{th} bit is 0, else take a right. This way, each of the first 50 bits goes a unique leaf. Then from each leaf, construct a linear path which accepts the reverse of the bit string from root to that leaf. Finally there is a reject state. This is the desired acceptor. This binary tree is constructed for $n = 2$ and attached in moodle.

   (b) Using the pigeonhole principle, argue that any such acceptor should have at least 2^{50} states. Can you tighten this argument to show that your construction from above has the minimum possible number of states? *Hint: Consider the set of prefixes of strings in the language. Can two distinct prefixes take the machine to the same state?*

   Solution: Consider the set S of 50-length bitstrings. This set has size 2^{50} If any deterministic acceptor for palindrome is of size less than 2^{50}, then two bit strings from S will end up at same state by PHP. Let they be x, y. Then, as xx^R, yy^R are accepted, xy^R and yx^R should also be accepted by the machine, although they are not palindromes. This is a contradiction. Hence, any such machine must have atleast 2^{50} states.

   Note: The above constructed binary tree acceptor for palindromes has $2^{50} * 51 + 1$. To prove this indeed is the minimal number of states, you might need to use theorems on minimizing a finite state acceptor. (you will learn them in your Automata course. Challenge : Try to prove this indeed is minimal !)

10. Show that each of the following decision problems is in **NP**, by describing a "certificate" (for "yes" instances) that can be verified in time that is polynomial in the length of the instance.

    (a) Is x the binary representation of a composite number?

    (b) Is x the binary representation of an even number?

    (c) Are the pair of graphs (G_1, G_2) (represented using their adjacency matrices) isomorphic to each other?

    (d) Does the graph G have a Hamiltonian cycle?

    (e) Does a polynomial $p(X)$ with integer coefficients have an integer root? *Hint: You may use the fact that all the real roots of a polynomial $c_d X^d + \ldots + c_1 X + c_0$ lie in the range $[-s, s]$, where $s = \max\{1, |c_0| + \ldots + |c_n|\}$. Why do you need to use this fact?*

    Solution: We simply need to find a certificate and provide a verification algorithm which runs in polynomial time for all "yes" instances.

    (a) If x is a composite number, there is a proper divisor > 1. We can take the certificate to be binary representation of that divisor and verification is a simple division to check whether reminder is zero. Since, this isi a polynomial algorithm, it is is **NP**.

    (b) Certificate - x itself. Verification:- check whether least bit in x is 0.

    (c) Certificate - a mapping of vertices from $G_1, G2$. Verification :- check if every edge in G_1 is between the mapped vertices in G_2 and vice versa. This is again a polynomial in number of edges and vertices.

    (d) Certificate - the integral solution to the equation itself !! (remember certificate can be anything, only catch verification algorithm needs to be polynomial in complexity).

    Verification :- simply put the solution in the equation and check whether it equals 0. Only thing left, is to show the verification runs in polynomial complexity. Note that here, the size of instance is the number of bits used for storing the coefficients. If the solution is exponential in the size of coefficients, then evaluating the polynomial $p(X)$ will take exponential time in size of instance. (remember complexity of naively multiplying two numbers equals product of number of bits of each).

    But fortunately, since, a solution exists between the $[-s, s]$ with $s = \max\{1, |c_0| + \ldots + |c_n|\}$. This solution has number of bits proportional to size of instance. Thus, verification for this solution is polynomial in the size of instance. Thus, we take this solution as certificate, then verification takes polynomial time and hence, the problem is in **NP**.