

Homework 10

7.1

- A. true.
- B. false.
- C. false.
- D. true.

7.5

No.

7.6

Let L_1 and L_2 be two languages in P , and M_1, M_2 such that $L(M_1) = L_1$ and $L(M_2) = L_2$.

Union.

Construct TM M as follows.

On input (w) :

1. Run M_1 on w . If M_1 accepts, M accepts.
2. Run M_2 on w . If M_2 accepts, M accepts.
3. reject.

Note that M accepts if either M_1 or M_2 does, thus deciding the union of the two languages. Also, Step 1 decides in polynomial time. Likewise Step 2 will be decided in polynomial time. Trivially, so will Step 3. Thus M decides in polynomial time.

Concatenation.

Construct TM M as follows:

On input (w) :

1. for $i = 0$ to $|w|$
 1. Run M_1 on $w[0,i]$
 2. Run M_2 on $w[i+1,|w|]$
 3. If M_1 and M_2 accept, then accept
2. reject

Note that M accepts only if both halves of w , as split at some point i , are in their respective languages. That is, $w[0,i]$, the substring of w from 0 to i is in L_1 and the remainder of the string, $w[i+1,|w|]$ is in L_2 . Only if there is no split of w in w_1, w_2 will TM M reject. Thus M decides the concatenation of L_1 and L_2 .

To show that M is in P , not that the inner loop steps each decide in polynomial time by construction. The summation of their execution times is thus in polynomial time, and the for loop will only increase this execution time by a factor. Thus M is in P .

Complement.

Construct TM M as follows:

On input(w):

1. Run M_1 on w .
2. If M_1 accepts w , reject. If M_1 rejects, accept.

Here, M outputs the opposite decider M_1 does. Thus M decides the complement of M_1 . Almost trivially, Step 1 and 2 are both executed in polynomial time, and thus M is in P .

7.9

To show that TRIANGLE is in P , consider the graph G represented as an adjacency matrix. The vertices become the column and row entries, and a 1 is marked if the two vertices are adjacent, a 0 if they are not.

For example, if $\langle G \rangle = \{V, E\} = \{ \{a, b, c\}, \{ab, bc\} \}$, then the adjacency matrix would look like:

	a	b	c
a	0	1	0
b	1	0	1
c	0	1	0

Now, given a matrix, representation, we can decide if the graph G has a triangle:

By construction, the matrix multiplication $A \times A = A^2$ amounts to the dot products of each row in A with each column in A . Specifically, $A^2[x, y] = A[x, *] \cdot A[*, y]$. Additionally, by construction, a dot product is the sum of pair-wise multiplications. Given that the values of A contain only 1 (for edge presence) or 0 (for a missing edge), the multiplication of two cells in A amounts to the binary AND operation indicating the presence of the two edges corresponding to the two cells. The dot product $A[x, *] \cdot A[*, y]$ amounts to the logical sum (OR) of all of the combinations of edges containing x and y as endpoints, and occupies cell $A^2[x, y]$. Therefore, A^2 gives the combinations of 2 edges that form angles, with vertices x and y at the unclosed ends; $A^2[x, y]$ is read as “an angle with unclosed ends x and y ”.

Similarly, the matrix multiplication $AxA^2 = A^3$ combines each edge each angle. The dot product of $A[x, *]$ with $A^2[* , y]$ gives the sum of all combinations of an edge containing vertex x with an angle with vertex y as an end point. Dot products including $A[x, y]$ and $A^2[x, y]$ indicate the pairing of an edge ending at vertexes x and y with an angle having endpoints at vertexes x and y , which by definition is a triangle. According to the definition of matrix multiplication, such combinations occur only at $A^3[x, x]$, which is along the diagonal. All other pairings involve $A[x, s]$ and $A^2[x, y]$, or $A[s, x]$ and $A^2[x, y]$, or or $A[t, u]$ and $A^2[x, y]$... all of which are not triangles.

Groups of three edges that begin with one vertex, and end with the same vertex are by definition a triangle and are found at each $A^3[x, x]$. A cell on the diagonal having a value of 1 or more indicates the presence of a triangle.

Thus, to create a TM S that decides TRIANGLE,

$S :=$ on input ($\langle G \rangle$)

1. create the adjacency matrix for the edges in G .
2. compute A^2 by multiplying A and A and write it after A on the tape.
3. compute A^3 by multiplying A and A^2 and write it after A^2 on the tape.
4. check the “diagonal” entries in A^3 , if any are 1, accept.
5. otherwise, reject.

Each step above can be performed in polynomial time. Step 1 in $O(n)$, Step 2 and 3 each in $O(n^3)$ and Step 4 $O(n)$ in linear time. Thus S decides TRIANGLE in Ptime.

7.11

To show that ISO is in NP, show that candidate isomorphic graphs G and H can be verified in polynomial time.

Thus, we are given some reordering of the nodes in G to G' so that they are identical to H . To verify that G' identical to H , verify each edge as follows:

Let L be a list of nodes that have been visited. Originally, this list is empty.

1. Let g be the first vertex on the tape in G
2. Let h in H be the candidate node equivalent to g .
3. If the degree of g is not equal to the degree of h , reject.
4. Add g and h to the list.
5. For each child g' of g not in the list.
 1. Let h' in H be the isomorphic vertex.
 2. If hh' is not in H , reject.
 3. Otherwise, execute Step 1 with the context of g' .
6. If each child of g is isomorphic to some child of h , then g and h are isomorphic.

Note that in the worst case, the recursion depth is the number of vertices in G . However, because only children not visited are considered at further depths, the algorithm halts and verifies that G is isomorphic to H in polynomial time.