

Week 10 CS-312 Homework

Cory Ness

Jack Engledow

James Sgrazzutti

December 10, 2020

1 Problem 1

1.1 Question

Show that it is undecidable if two TMs accept the same language.

1.2 Answer

Suppose by contradiction there is a Turing Machine T that always decides whether two TMs accept the same language, that is $S_{tm} = \{ \langle A, B \rangle : A \text{ is a Turing Machine that accepts the same language as } B \}$ where T describes S_{tm} . Take two new TMs A, B , where A will accept any language and B will always run forever.

Take T and pass A and B into it. Since T is decidable, it will always return an accept or a reject.

In order to decide anything, it must be determined if B accepts or rejects a string, which is impossible as it will run forever. If we were able to determine if B halts, we could solve this, we cannot due to the halting problem, hence there is a contradiction and TM T cannot exist.

2 Problem 2

2.1 Question

Show that $ALL_{DFA} \in \mathcal{P}$. That is, show that you can test in polynomial time whether a DFA accepts every string.

2.2 Answer

A DFA accepts every string if and only if every state inside it is a final state. Hence, we can perform a search of every reachable node in the DFA (using some graph traversing algorithm such as breadth first which is known to be complete in polynomial time), and if every one of them is final, the DFA accepts every string. Since every operation performed is done in polynomial time, the test is done in polynomial time.

3 Problem 3

3.1 Question

Show that the set \mathcal{P} is closed under the star operation. (Hint: Dynamic Programming)

3.2 Answer

To show that \mathcal{P} is closed under the star operation, we have to show that $x_1...x_n \in L$ in polynomial time. We can do this inductively, with the base case being $n = 0$. With $n = 0$, we accept immediately, which is clearly polynomial time. In the inductive step, we prove that $x_{n+1} \in \mathcal{P}$ if $x_n \in \mathcal{P}$. This is done by realizing that x_{n+1} is computed the same as $x_n + x_{n+1}$, which is clearly computed in polynomial time if x_n is calculated in polynomial time. Hence, \mathcal{P} is closed under the star operation.

4 Problem 4

4.1 Question A

Define SPATH as the set of $\langle G, a, b, k \rangle$ such that G is a graph with a path from a to b of length at most k . Show that SPATH is in \mathcal{P} .

4.2 Question B

Define LPATH as the set of $\langle G, a, b, k \rangle$ such that G is a graph with a path from a to b without repeated nodes of length at least k . Show that LPATH is \mathcal{NP} -Complete. (You may assume NP-completeness of HAMPATH).

4.3 Answer A

Perform a depth-first search (done in polynomial time) modified such that you don't end when you reach a previously reached node, instead you stop when you reach a node that's been reached in this specific branch of the search. Start at node a . During the depth-first search, count down from k , if you find b before your count reaches 0, then accept. If your count reaches 0 before finding b , end your branch search and move to the next depth. If after the search you never find b before your count reaches 0, then reject. Every operation after the depth-first search is also done in polynomial time (counting down, comparing), hence SPATH is in \mathcal{P} .

4.4 Answer B

5 Problem 5

5.1 Question

Show that if $\mathcal{P} = \mathcal{NP}$, then every language in \mathcal{P} except \emptyset and Σ^* is \mathcal{NP} -complete.

5.2 Answer

We know that $\mathcal{P} \subseteq \mathcal{NP}$. We also know that if $\mathcal{P} = \mathcal{NP}$, then every \mathcal{NP} language is polynomial-time reducible to \mathcal{P} . Since every \mathcal{P} language is also \mathcal{NP} . Since every \mathcal{P} language is polynomial-time reducible to another language in \mathcal{P} , and since $\mathcal{P} = \mathcal{NP}$, every language in \mathcal{P} is polynomial-time reducible to a \mathcal{NP} -complete language, hence \mathcal{P} is \mathcal{NP} -complete.

6 Problem 6

6.1 Question

Show that the CLIQUE problem is \mathcal{NP} -complete by reducing from VERTEX_COVER.

6.2 Answer

7 Problem 7

7.1 Question

Convince your instructor that you understood this course.

7.2 Answer