CSE 105: Automata and Computability Theory **Fall 2015**

Homework #6

Due: Wednesday, Dec 2nd, 2015, 11:59pm PM

Submit your solutions to gradescope.com. Sign up/Sign in using your ucsd.edu email address. You will have to add yourself to the course using the 6 digit entry code "9WB6WM". For each answer, include a line indicating your name , PID and names of all students you collaborated with on the solution. For issues regarding gradescope access drop in an email to one of the TA's.

In this problem set we make use of the following languages:

- $L_{NFA} = \{\langle M \rangle : M \text{ is an NFA}\}$, i.e., the set of all syntactically valid NFAs.
- $L_{NFA1} = \{\langle M \rangle : M \text{ is an NFA with } |F| = 1\}$, i.e., the set of all NFAs with a single final state.
- $ONE_{NFA} = \{\langle M \rangle : M \text{ is an NFA such that } |\mathcal{L}(M)| = 1\}$, i.e., the set of all NFAs that accept only one string.
- $EPS_{NFA} = \{\langle M \rangle : M \text{ is an NFA such that } \mathcal{L}(M) = \{\epsilon\}\}$, i.e., the set of all NFAs that accept only the empty string.
- $ONE_{NFA1} = ONE_{NFA} \cap L_{NFA1}$, like ONE_{NFA} but with only one final state.
- $EPS_{NFA1} = EPS_{NFA} \cap L_{NFA1}$, like EPS_{NFA} but with only one final state.
- $ONE_{TM} = \{\langle M \rangle : M \text{ is a Turing machine and } |\mathcal{L}(M)| = 1\}$ i.e., the set of all Turing machines that accept precisely one input string.
- $TWO_{TM} = \{\langle M \rangle : M \text{ is a Turing machine and } |\mathcal{L}(M)| = 2\}$ i.e., the set of Turing machines that accept precisely two inputs.

1 Problem 1

Let $f(\langle M \rangle) = \langle M' \rangle$ be a function that on input an NFA M outputs an equivalent NFA M' with only one final state. Specifically, if $M = \langle Q, \Sigma, \delta, s, F \rangle$, then $M' = \langle Q \cup \{f\}, \Sigma, \delta', s, \{f\} \rangle$ where $f \notin Q$ is a new state, and the transition function δ' is defined as

$$\delta'(q, x) = \begin{cases} \emptyset & \text{if } q = f \text{ and } x \in \Sigma \cup \{\epsilon\} \\ \delta(q, x) \cup \{f\} & \text{if } q \in F \text{ and } x = \epsilon \\ \delta(q, x) & \text{otherwise} \end{cases}$$

For each of the following pair of languages, state if f is a valid mapping reduction from the first to the second, and if not, provide an input demonstrating that f is not valid:

1. $L_{NFA} \leq_m L_{NFA1}$
2. $L_{NFA} \leq_m ONE_{NFA}$
3. $EPS_{NFA} \leq_m EPS_{NFA}$
4. $ONE_{NFA} \leq_m ONE_{NFA1}$
5. $EPS_{NFA} \leq_m ONE_{NFA1}$
6. $ONE_{NFA} \leq_m L_{NFA}$

Notice that in all cases the question is whether the specific funciton f defined above is a mapping reduction. (We are not asking you to find a reduction, or whether a reduction exists.)

Problem 2

In this problem you will show that ONE_{TM} is neither recognizable nor co-recognizable

a. Prove that ONE_{TM} is not recognizable by mapping-reduction from the diagonal language

$$D = \{\langle M \rangle \mid M \text{ is a Turing machine such that } \langle M \rangle \notin \mathcal{L}(M)\}$$

b. Prove that ONE_{TM} is not co-recognizable by reduction from the complement of the diagonal language $\overline{D}$.

Problem 3

We say that two languages A and B are Turing equivalent if there is both a mapping reduction from A to B, and a mapping reduction from B to A. In this problem you will prove that the languages ONE_{TM} and TWO_{TM} are Turing equivalent.

a. Give a mapping reduction from ONE_{TM} to TWO_{TM}

b. Give a mapping reduction from TWO_{TM} to ONE_{TM}.

Hint: The b. direction is the harder one. For this direction, think about how, given a machine M, you might design a machine M' that accepts all strings M does, except the first one.

Problem 4

Use the results from problem 2 and problem 3 to prove that the language TWO_{TM} is *not* Turing equivalent to the diagonal language D.