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

Homework #6

Jimmy Ye A12405090

Solution to Problem 1

1. $L_{NFA} \leq_m L_{NFA1}$

Yes, f is a valid mapping reduction.

2. $L_{NFA} \leq_m ONE_{NFA}$

No, f is not a valid mapping reduction. Consider any NFA accepting more than one string (for example, the trivial NFA accepting every string formed from the alphabet $\{a\}$). Then f maps this NFA, M, to an equivalent NFA, M'. $\langle M\rangle$ is in L_{NFA} by definition, but M' accepts more than one string, so $\langle M'\rangle$ is not in ONE_{NFA}.

3. $EPS_{NFA} \leq_m EPS_{NFA}$

Yes, f is a valid mapping reduction.

4. $ONE_{NFA} \leq_m ONE_{NFA1}$

Yes, f is a valid mapping reduction.

5. $EPS_{NFA} \leq_m ONE_{NFA1}$

No, f is not a valid mapping reduction. Consider any NFA which accepts only one non-empty string (for example, the trivial NFA over the alphabet $\{a\}$ accepting only a). Then f maps this NFA, M, to an equivalent NFA, M'. However, $\langle M\rangle$ is not in EPS_{NFA} since M accepts a non-empty string, but $\langle M'\rangle$ is in ONE_{NFA1} since M' accepts only one string and has only one final state.

6. $ONE_{NFA} \leq_m L_{NFA}$

No, f is not a valid mapping reduction. Consider any NFA M accepting more than one string. M is not in ONE_{NFA}, but f maps M to an equivalent NFA M' and M' is in L_{NFA}.

Jimmy Ye — A12405090
Solution to Problem 2

a. Prove that ONE_{TM} is not recognizable by mapping-reduction from the diagonal language $D = \{\langle M \rangle | M$ is a Turing machine such that $\langle M \rangle \notin L(M)\}$.

We construct such a mapping-reduction, f, which outputs $\langle M' \rangle$:

Given any word w, if it is not a valid encoding of a Turing machine, M' always rejects.

Otherwise, $w = \langle M \rangle$ for some Turing machine M. We treat Turing machines as the languages which they accept, taking union and intersection as operations which construct Turing machines, which simulate both operands and accept the union or intersection of their languages, respectively.

Construct M_1 as a Turing machine accepting only $\langle M \rangle$, and M_2 as a Turing machine accepting only ϵ, the empty string. Then we construct M' as $M_2 \cup (M \cap M_1)$, which is possible in finite time using only the encoding of M.

If $\langle M \rangle \in D$: M_2 accepts ϵ. M does not accept $\langle M \rangle$ and by construction M_1 accepts no other strings, so $M \cap M_1$ accepts no strings. So M' accepts only one string, ϵ, so $\langle M' \rangle \in ONE_{TM}$.

If $\langle M \rangle \notin D$: M_2 accepts ϵ. M and M_1 accept $\langle M \rangle$, so $M \cap M_1$ accepts $\langle M \rangle$. So M' accepts ϵ and $\langle M \rangle$, so $\langle M' \rangle \notin ONE_{TM}$.

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

Note that $\overline{D} = \{\langle M \rangle | \langle M \rangle \in L(M)\}$

We construct such a mapping-reduction, f, which outputs $\langle M' \rangle$:

Given any word w, if it is not a valid encoding of a Turing machine, M' always rejects. Otherwise, $w = \langle M \rangle$ for some Turing machine M. Construct M_1 as a Turing machine accepting only $\langle M \rangle$. Then we construct M' as $M \cap M_1$, which is possible in finite time using only the encoding of M.

If $\langle M \rangle \in \overline{D}$, M and M_1 both accept $\langle M \rangle$, and by construction M_1 accepts no other strings. So M' accepts only one string, $\langle M \rangle$, so $\langle M' \rangle \in ONE_{TM}$.

If $\langle M \rangle \notin \overline{D}$, M does not accept $\langle M \rangle$, and by construction M_1 accepts no other strings. So M' accepts no strings, so $\langle M' \rangle \notin ONE_{TM}$.

Jimmy Ye — A12405090
Solution to Problem 3

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

We construct such a mapping-reduction, f, which outputs $\langle M' \rangle$:

Again, given w, if w is not a valid Turing machine encoding, M' always rejects.

Otherwise, $w = \langle M \rangle$ for some Turing machine M. We construct M' from M, adding an additional symbol x to the alphabet of M' which is not in the alphabet of M. We construct M' to accept the string x, but otherwise reject strings containing an x symbol. For strings that do not contain an x, M' behaves identically to M.

Then M' accepts exactly one more string than M does. So if $\langle M \rangle \in ONE_{TM}$, then $\langle M' \rangle \in TWO_{TM}$, and likewise if $\langle M \rangle \notin ONE_{TM}$, then $\langle M' \rangle \notin TWO_{TM}$.

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

We construct such a mapping-reduction, f, which outputs $\langle M' \rangle$:

Again, given w, if w is not a valid Turing machine encoding, M' always rejects.

Otherwise, $w = \langle M \rangle$ for some Turing machine M. Since M has a finite alphabet, we assign some arbitrary ordering of this alphabet (and subsequently this generates a lexicographical ordering for every string we can generate from this alphabet).

Then on any input w, we construct M' to simultaneously run M on every possible input w' which comes before w in the lexicographical ordering, as well as w itself.

If M accepts w before any other w' (i.e. at the step at which $M(w)$ reaches q_{accept}, no other simulated instance of M has reached q_{accept}), then M' rejects. If M accepts w at the same step as a nonempty set of other w', then M' rejects w only if w is the lexicographically smallest word, and accepts otherwise. If M accepts w after any other w', then M' accepts w.

Since, for any given w, there are finitely many w' which are lexicographically lesser, M' simulates finitely many instances of M, so M' is well-defined. Lexicographical comparison can be computed by a Turing machine in a finite number of steps, and there are finitely many w', so checking if w is the smallest can be done in a finite number of steps.

So by construction, M' differs from M only when the input w is the unique word (1) accepted in the least number of steps by M and (2) the lexicographically smallest such word if multiple words are accepted in the same number of steps. In this case, M' rejects w whereas M accepts w.

So M' always accepts one less string than M (unless M accepts no strings, in which case M' also accepts no strings), so if $\langle M \rangle \in TWO_{TM}$, then $\langle M' \rangle \in ONE_{TM}$, and likewise if $\langle M \rangle \notin TWO_{TM}$, then $\langle M' \rangle \notin ONE_{TM}$

Jimmy Ye — A12405090
Solution to 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.

From (3), we know that ONE_{TM} and TWO_{TM} are Turing equivalent. From (2), we know that ONE_{TM} is neither recognizable nor co-recognizable. This implies that TWO_{TM} is neither recognizable nor co-recognizable.

Consider $\overline{D}$. It is recognizable, since given $w = \langle M \rangle$, we construct M' to compute M with input $\langle M \rangle$, and if it is in $\overline{D}$, it will accept, and if it is not in $\overline{D}$, it will not accept. (On input w not representing a Turing machine, M' rejects)

Assume that TWO_{TM} is Turing equivalent to D. Then since TWO_{TM} is neither recognizable nor co-recognizable, D is also neither recognizable nor co-recognizable. But if D is not co-recognizable, then $\overline{D}$ is not recognizable, which contradicts what we have shown above.

So TWO_{TM} is not Turing equivalent to D.