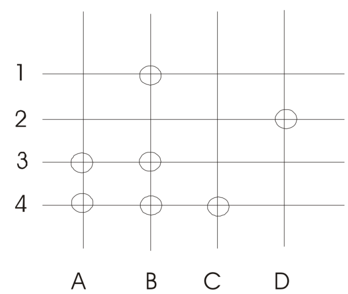
CS 341 Automata Theory
STUDENT NAME - EID
Homework 13
Due: Tuesday, April 17

This assignment covers Sections 21.1 - 21.3

1) In Appendix E.3, we describe a straightforward use of reduction that solves a grid coloring problem by reducing it to a graph problem. Given the grid G shown here:



a) Show the graph that corresponds to G.

Solution.

$$G' = \{V, E\}$$
$$V = \{A, B, C, D, 1, 2, 3, 4\}$$
$$E = \{(A, 3), (A, 4), (B, 1), (B, 3), (B, 4), (C, 4), (D, 2)\}$$

□

b) Use the graph algorithm we describe to find a coloring of G.

Solution. Start with B and color each edge alternately.

$$Red = \{(B, 1), (B, 4)\}$$
$$Blue = \{(B, 3)\}$$

Now do vertex 3:

$$Red = \{(B, 1), (B, 4), (A, 3)\}$$
$$Blue = \{(B, 3)\}$$

Now vertex A:

$$Red = \{(B, 1), (B, 4), (A, 3)\}$$
$$Blue = \{(B, 3), (A, 4)\}$$

Now vertex 4: (it has one of each already, so pick arbitrarily.)

$$Red = \{(B, 1), (B, 4), (A, 3)\}$$
$$Blue = \{(B, 3), (A, 4), (C, 4)\}$$

And now the last edge:

$$Red = \{(B,1),(B,4),(A,3),(D,2)\}$$
$$Blue = \{(B,3),(A,4),(C,4)\}$$

□

2) In this problem, we consider the relationship between H and a very simple language $\{a\}$.

a) Show that $\{\mathtt{a}\}$ is *mapping* reducible to H.

Solution. Construct a turing machine M which halts and accepts if the square to the right of the read head is an `a` and otherwise loops. The problem of recognizing $\{\mathtt{a}\}$ is now equivalent to the problem of deciding whether M halts. □

b) Is it possible to reduce H to $\{\mathtt{a}\}$? Prove your answer.

Answer. No. □

Proof. We know that $\{\mathtt{a}\}$ is a regular language, and is thus decidable. So if there were a way to reduce H to $\{\mathtt{a}\}$, then H would be decidable. However we know that H is not decidable, so such a reduction must not exist. □

3) Show that H_{ALL} is not in D by reduction from H.

Solution. Assume by way of contradiction that a Turing Machine *Oracle* that decided H_{ALL} existed. Now define a reduction $R(\langle M, w\rangle)$ which constructs a description of a new machine $M\#$ that does the following:

1. Erases its input tape.
2. Writes w onto the tape.
3. Passes control to M.

Now when $M(w)$ halts $Oracle(R(\langle M, w\rangle))$ accepts and when $M(w)$ does not halt $Oracle(R(\langle M, w\rangle))$ rejects. □

4) For each of the following languages L, state whether or not it is in D. Prove your answer. Assume that any input of the form $\langle M\rangle$ is a description of a Turing machine.

a) $\{\langle M\rangle \ : \ \mathtt{ab} \in L(M)\}$.

Answer. No. □

Proof. Define $R(\langle M, w\rangle)$, a mapping reduction from H to this language as follows:

1. Construct the description $\langle M\#\rangle$ of a new Turing machine $M\#$ that, on input x, operates as follows:
   1.1 Erase the tape.
   1.2 Write w on the tape.
   1.3 Run M on w.
   1.4 Accept.
2. Return $\langle M\#, w\rangle$.

If there were a machine to decide this language, call it $Oracle$ then $C = Oracle(R(\langle M, w\rangle))$ would decide H.
If $\langle M, w\rangle \in H$: M halts on w, so $M\#$ accepts everything. In particular, it accepts `ab`. $Oracle(\langle M\#\rangle)$ accepts.
If $\langle M, w\rangle \notin H$: M halts on w, so $M\#$ does not halt on any input. In particular, it does not halt on `ab`. $Oracle(\langle M\#\rangle)$ rejects. □

b) $\{\langle M, w\rangle$: TM M, on input w, begins by moving right one square onto w. Then it never moves off $w\}$.

Answer. No. □

Proof. □

c) $\{\langle M\rangle$: there exists a string w such that $|w| < |\langle M\rangle|$ and that M accepts $w\}$.

Answer. No. □

Proof. Let R be a mapping reduction from H to this language as follows:
$R(\langle M, w\rangle) =$

1. Construct the description $\langle\{\rangle\, M\#\}$ of a new Turine machine $M\#(x)$ that, on input x, operates as follows:
   1.1 Erase the tape.
   1.2 Write w to the tape.
   1.3 Run M on w
   1.4 Accept.
2. Return $\langle M\#\rangle$.

Assume by way of contradiction that there exists a machine to decide this language, call it $Oracle$. If $\langle M, w\rangle \in H$: then $M\#$ will accept on all input, including ϵ. So there exists at least one string with length less than $\langle M\rangle$ on which $M\#$ accepts. $Oracle$ accepts.
If $\langle M, w\rangle \notin H$: then $M\#$ will not halt, and thus not accept, on all input. So there does not exists at least one string with length less than $\langle M\rangle$ on which $M\#$ accepts. $Oracle$ rejects.
Therefore $Oracle(R(\langle M, w\rangle))$ decides H. However we know that no such machine exists, so $Oracle$ must not exist. □

5) In Appendix J.2, we proved Theorem J.1, which tells us that the safety of even a very simple security model is undecidable, by reduction from H_ϵ. Show an alternative proof that reduces $A = \{\langle M, w\rangle : M$ is a Turing machine and $w \in L(M)\}$ to the language Safety.

Proof. □