Take-Home Final, CS 500, Spring 2014
due by 4:00pm, Wednesday May 14.

Start each problem on a fresh sheet of paper. You may typeset or handwrite. Turn in a hardcopy of your solutions at my office, FEC 149, on Wednesday, between 2:00 and 4:00, or earlier by appointment.

Rules:

1. You may not consult any person other than Prof. Hayes during the exam.
2. You may not consult any textbooks or internet resources regarding the exam problems or subject material. Exception: the course textbook and Automata Theory supplement are allowed. Requests to allow other specific sources will be considered.
3. You may discuss any problem with Prof. Hayes by email. Please keep such correspondence short and to the point. This will be considered part of your solution set. Answers or hints given in response to questions you pose by email may reduce the total points available to you. I will try to answer all emails within 24 hours of receipt.
4. Your solution must contain the sentence, "I have read and followed the rules," followed by your signature.

There are 2 questions with 3 parts each. All answers must be proved, not just stated.

1. Invent a way to combine any two languages $L_1, L_2 \subset \{0,1\}^*$ to create a third language $L_3 = f(L_1, L_2)$, with the following properties.

1. Whenever $L \leq L_1$, then $L \leq L_3$. Also, whenever $L \leq L_2$, then $L \leq L_3$.
2. Whenever $L_1 \leq L'$ and $L_2 \leq L'$, then $L_3 \leq L'$.

(a) Prove that your invention satisfies both properties.
We say a complexity class $\mathcal{C}$ is "closed under f" if whenever $L_1, L_2 \in \mathcal{C}$, then $f(L_1, L_2) \in \mathcal{C}$.
(b) For your definition of f, are the regular languages closed?
(c) For your definition of f, is the class 1-DCA closed?

2. In Vertex-and-Edge Coloring, the input is a graph, and a "k-coloring" assigns one of k colors, $\{1, 2, \ldots, k\}$, to every edge and to every vertex. Simple graph

(a) Suppose, to be a "proper" coloring, we require only that every vertex be a different color from every edge touching it. For which values of k is k-Vertex-and-Edge Coloring NP-complete?

(b) Now suppose that, to be a proper coloring, we *additionally* require that every vertex be a different color from every adjacent vertex. For which values of k is k-Vertex-and-Edge Coloring NP-complete?

(c) Finally, suppose that, to be a proper coloring, we add yet another requirement, that every two edges that meet at a shared vertex must be given different colors. For which values of k is this version NP-complete?

undirected edges
no self-loops