Problem Set 1 – Due Tuesday, April 6, 2010

Instructions: Write up your solutions as clearly and succinctly as you can. Typeset solutions, particularly in LaTeX, are always appreciated. Don't forget to acknowledge anyone with whom you discussed problems. Recall that homeworks are due at 4:40 pm sharp on Tuesdays, in the turn-in box in Kemper Hall, room #2131.

Problem 1. Show that at a party of 10 people, there are at least two people who have the same number of friends present at the party. Assume (however unrealistically) that friendship is symmetric and anti-reflexive. *Hint*: Carefully use the pigeonhole principle.

Problem 2. Let $G = (V, E)$ be a graph (the "usual" sort, being nonempty, finite, undirected, having no-self loops and no multiple edges). Prove (by giving a convincing argument) or disprove (by giving a smallest counter-example) that the following are equivalence relations for any graph G.

Part A. Let $x, y \in V$. Say that $x \sim y$ if there is a path in G from x to y (that is, a sequence of vertices $x_1, \ldots, x_n \in V$ ($n \geq 1$) where each $\{x_i, x_{i+1}\} \in E$ and $x = x_1$ and $y = x_n$).

Part B. Let $x, y \in V$. Say that $x \sim y$ if x is adjacent to y (that is, $\{x, y\} \in E$).

Part C. Let $x, y \in V$. Say that $x \sim y$ if $x = y$ or $\{x, y\} \in E$ or there are two vertex-disjoint paths from x to y (paths $x_1, \ldots, x_n$ and $x'_1, \ldots, x'_{n'}$ where $x_1 = x'_1 = x$ and $x_n = x'_{n'} = y$ and $\{x_2, \ldots, x_{n-1}\} \cap \{x'_2, \ldots, x'_{n'-1}\} = \emptyset$.

Part D. Let $x, y \in V$. Say that $x \sim y$ if there is a path from x to y and this remains so even if one removes any edge $e \in E$.

Problem 3. State whether the following propositions are true or false, explaining each answer.

Part A. $\emptyset$ is a language.

Part B. $\emptyset$ is a string.

Part C. ϵ is a language.

Part D. ϵ is a string.

Part E. Every language is infinite or has an infinite complement.

Part F. Some language is infinite and has an infinite complement.

Part G. The set of real numbers is a language.

Part H. There is a language that is a subset of every language.

Part I. The Kleene-star (Kleene closure) of a language is always infinite.

Part J. The concatenation of an infinite language and a finite language is always infinite.

Part K. There is an infinite language L containing the emptystring and such that L^i is a proper subset of L^* for all $i \geq 0$.

Problem 4. Give DFAs for the following languages. Assume an alphabet of $\Sigma = \{0, 1\}$.

(a) The set of all strings with 010 as a substring.

(b) The set of all strings which do not have 010 as a substring.

(c) The set of all strings which have an even number of 0's or an even number of 1's.

(d) The complement of $\{0, 01\}^*$.

(e) The binary encodings of numbers divisible by 3: $\{0\}^* \circ \{\varepsilon, 11, 110, 1001, 1100, 1111, \ldots\}$.

Problem 5 State whether the following propositions are true or false, proving each answer.

Part A. Every DFA-acceptable language can be accepted by a DFA with an even number of states.

Part B. Every DFA-acceptable language can be accepted by a DFA whose start state is never visited twice.

Part C. Every DFA-acceptable language can be accepted by a DFA no state of which is ever visited more than once.

Part D. Every infinite DFA-acceptable language can be accepted by a DFA that, for some string $x \in L$, visits the start state twice on input x.

Part E. Every DFA-acceptable language can be accepted by a DFA with only a single final state.

Problem 6. Recall the DIOPHANTINE EQUATION problem: given a multivariate polynomial P with integer coefficients (e.g., $P(x, y, z) = x^2 - 5xy + 3yz^2 + xyz$), decide whether or not P has an integer root. I claimed without proof that there is no algorithm to answer this question. But suppose I provide you with a "magic box" that answers the question. In a single computational step, it says *yes* or *no* according to whether or not P has a root. Given such a magic box, describe an algorithm that *finds* an integer root of any multivariate polynomial that has one (and the algorithm answers *No Root* if the polynomial provided doesn't have an integer root).