Qualifying Examination
Theoretical Computer Science
Monday, October 7, 2019
Part II: Automata and Complexity

Instructions:

(a) This is a closed book exam.

(b) The exam has four problems worth 25 points each. Read all the problems carefully to see the order in which you want to tackle them. You have all day (9am–5pm) to solve the problems.

(c) Write clearly and concisely. You may appeal to some standard algorithms/facts from textbooks unless the problem explicitly asks for a proof of that fact or the details of that algorithm.

(d) If you cannot solve a problem, to get partial credit write down your main idea/approach in a clear and concise way. For example you can obtain a solution assuming a clearly stated lemma that you believe should be true but cannot prove during the exam. However, please do not write down a laundry list of half-baked ideas just to get partial credit.

(e) In problems with multiple parts, if you cannot solve a part, you can still solve subsequent parts (assuming that claims from earlier parts are true) and get partial credit.

May the force be with you.

Problem 1: Let L be an arbitrary regular language over the alphabet $\Sigma = \{0, 1\}$, and let X be an *arbitrary* set of non-negative integers. Prove that the language $L^X = \{w \in \Sigma^* \mid w^n \in L \text{ for some } n \in X\}$ is regular.

Problem 2: For this question, we define *perpetual* DFAs. Perpetual DFAs are like the usual deterministic finite automata, *but they work on infinitely long inputs.*

A perpetual DFA is defined by the tuple $(Q, S_0, S_1, \Sigma, \delta, q_0)$, where

- S_0 and S_1 partition the **finite** set of states Q: in other words, for each state q in the perpetual DFA, either $q \in S_0$ or $q \in S_1$. Thus, $Q = S_0 \cup S_1$.
- Σ is the alphabet.

- $\delta : Q \times \Sigma \to Q$ is the (deterministic) state transition function.
- $q_0 \in Q$ is the start state.

Observe that like the usual DFAs, perpetual DFAs have a finite number of states. But unlike the usual DFAs, perpetual DFAs have no accepting states or final state. Instead, looking ahead, the sets of states S_0 and S_1 will be used to convey meaning about the input read so far.

Denote by Σ^ω the set of all infinite words over the alphabet Σ: $\Sigma^\omega = \{a_1a_2a_3\cdots \mid \forall i,\ a_i \in \Sigma\}$. The perpetual DFA on input $\alpha = a_1a_2a_3\cdots \in \Sigma^\omega$ transitions across an infinite sequence of states $q_0q_1q_2\cdots$, where q_0 is the initial state, and $q_{i+1} = \delta(q_i, a_{i+1})$, $\forall i \geq 0$. We denote the current state of the perpetual DFA after reading the prefix $\alpha_i = a_1a_2a_3\cdots a_i$ (that is, the first i letters of input), by q_i.

(a) Construct a perpetual DFA such that:
For every finite prefix α_i of the input α, it is true that: $q_i \in S_0$ if the parity of the input bits read so far is 0 (that is, $a_1 \oplus a_2 \oplus \cdots \oplus a_i = 0$). Similarly, $q_i \in S_1$ if the parity of the input bits read so far is 1 (that is, $a_1 \oplus a_2 \oplus \cdots \oplus a_i = 1$). On the empty input, there is no requirement for the (start) state to belong to any specific set.

(b) Construct a perpetual DFA such that:
For every finite prefix α_i of the input α, it is true that: $q_i \in S_0$ if the parity of the previous two input bits is 0 (that is, $a_{i-1} \oplus a_i = 0$). Similarly, $q_i \in S_1$ if the parity of the previous two input bits is 1 (that is, $a_{i-1} \oplus a_i = 1$). On inputs of size 0 or 1, there is no requirement for the current state to belong to any specific set.

(c) Prove that you *cannot* construct a perpetual DFA which satisfies the following property:
For every finite prefix α_i of the input α, it is true that: $q_i \in S_0$ if the majority of the input bits read so far is 0 (that is, $\text{majority}(a_1, a_2, \ldots, a_i) = 0$). Similarly, $q_i \in S_1$ if the majority of the input read so far is 1 (that is, $\text{majority}(a_1, a_2, \ldots, a_i) = 1$). If the number of 0s equals the number of 1s, there is no requirement for the current state to belong to any specific set.

Problem 3: A language $L \subseteq \{0,1\}^\star$ is *unary* if L only contains strings with no zeroes, that is, $L \subseteq \{1\}^\star = \{\lambda, 1, 1^2, 1^3, \ldots, 1^n, \ldots\}$ (where λ denotes the empty string). Prove that if every unary language in NP is also in P, then $\mathsf{EXP} = \mathsf{NEXP}$.

Problem 4: Let $\mathsf{MULT} = \{a\#b\#c \mid a, b, c \text{ are binary natural numbers}, a \times b = c\}$. Show that $\mathsf{MULT} \in \mathsf{L}$ (where L denotes the class of all languages decidable in logarithmic space).