Problem Set 1

CS373 - Spring 2011

Due: Thursday Feb 10 at 2:00 PM in class (151 Everitt Lab)

Please follow the homework format guidelines posted on the class web page:

`http://www.cs.uiuc.edu/class/sp11/cs373/`

1. Encoding input and building a DFA
   [**Category**: Design, **Points**: 15] Let $\Sigma = \{0, 1\}$.

   i) Create an injective function g that maps the natural numbers to members of Σ^* such that $|g(x)| > |g(y)|$ only if $x > y$. (1 Point)

   ii) Design a DFA that recognizes the following language $L = \{w \in \Sigma^* \mid w = g(n)$, where $n = 5^k$ for a natural number $k\}$. You must use the injective function that you created in part i. To recieve full credit, the DFA can have at most six states. (5 Points)

   iii) Design a DFA that recognizes the following language $L = \{w \in \Sigma^* \mid w = g(n)$, where $n = 25^k$ for a natural number $k\}$. You must use the injective function that you created in part i. To recieve full credit, the DFA can have at most nine states. (3 Points)

   iv) Create a different injective function h that maps the natural numbers to members of Σ^*. There are no restrictions on h except that it must be injective. (1 Point)

   v) Design a DFA that recognizes the following language $L = \{w \in \Sigma^* \mid w = h(n)$, where $n = 5^k$ for a natural number $k\}$. You must use the injective function that you created in part iv. To recieve full credit, the DFA can have at most three states. (5 Points)

2. FAFA
   [**Category**: Proof, **Points**: 20] Recall that an NFA M can be represented as a directed graph $G(M)$ where the graph vertices represent the states of the NFA and the graph edges represent state transitions. Each graph edge is labelled with a member of $\Sigma \cup \{\varepsilon\}$ called the *transition symbol*. A *path* of a string w in an NFA M is a path through $G(M)$ that starts at a start state of M and where the concatenation of the state transition symbols of the edges traversed is equal to w. A string w could have a single path in M, multiple paths in M, or no paths in M.

   An NFA accepts a string $w \in \Sigma^*$ if any path of w in the NFA ends in an accept state. A for-all finite automaton (an FAFA) is defined in the same way as an NFA, except that a FAFA accepts a string w if there is at least one path of w that ends in an accept state and all paths of w end in an accept state.

   To use the informal terminology from the class, an FAFA accepts a string w if and only if at least one "universe" remains once w has been completely read, and all remaining universes are in an accept state.

   i) Prove that any language recognized by an FAFA is a regular language. (3 Points)

ii) Let $L_n = \{ww \mid w \in \{0,1\}^n\}$. This is the set of binary strings that consist of a length n word concatenated with itself. Show that for each n, there exists an FAFA M_n that recognizes L_n. The number of states in M_n must be $O(n^2)$. (8 Points)

iii) Show that an NFA recognizing L_n requires at least 2^n states. (9 Points)

3. Kleene Stars
[**Category**: Proof, **Points**: 15] Let $L_1 = \{a, b\}$. Let $L_2 = \{\varepsilon, a, b, ab\}$. Note that, even though L_1 and L_2 are different, $L_1^* = L_2^*$.

    i) If language L_3 is a non-empty subset of language L_4, then is it always true that $L_3^* = L_4^*$? If it is true, provide a proof. If it is false, then provide a counter-example. (3 Points)

    ii) Define $\emptyset^* = \{\varepsilon\}$. Find another language whose Kleene closure is also $\{\varepsilon\}$. (3 Points)

    iii) Are there two non-empty disjoint languages that have the same Kleene closure? Provide a proof for your answer. (9 Points)

4. Single character changes
[**Category**: Proof, **Points**: 15] Given a string w from the alphabet $\{0, 1\}$, define the following language: $D(w) = \{w' \in \Sigma^* \mid \ |w| = |w'|$ and exactly one character of w differs from $w'\}$. For example, $D(011) = \{111, 001, 010\}$. We can also extend the definition of D to languages: $D(L) = \{w' \in \Sigma^* \mid \exists w \in L, w' \in D(w)\}$.

    i) Given a regular language L, build an NFA that recognizes $D(L)$. (8 Points)

    ii) Given a regular language L, show that $D(L)$ is regular using only closure properties. Do not construct an automaton. (7 Points)

5. Graph walks
[**Category**: Proof, **Points**: 15] Consider a connected, undirected graph $G = (V, E)$ with vertices $v_i \in V$ and edges $\{v_i, v_j\} \in E$ (where both $|V| > 1$ and $|E| \geq 1$). We define a *walk* of length $n-1$ as a (possibly repeating) sequence of vertices $[v_1, v_2, v_3, \ldots, v_n]$ where adjacent vertices in the sequence are connected by an edge. We allow walks to be of length 0, meaning they have only one vertex. If we think of the vertices as symbols, every path (i.e. sequence of vertices) is equivalent to a string (i.e. sequence of symbols) over the alphabet $\Sigma = V$. Build the smallest DFA that recognizes the following language: $Walks(G) = \{v_1 v_2 \ldots v_n \in \Sigma^* \mid n \geq 1$ and $[v_1, v_2, \ldots, v_n]$ is a walk in $G\}$.

6. DFA vs. NFA size
[**Category**: Design, **Points**: 20] Let $|\Sigma| = k \geq 2$. Define the following language $L = \{w \in \Sigma^* \mid \exists a, b \in \Sigma$ where $a \neq b$ and w does not contain a or $b\}$.

    i) Build a DFA for L and give a formula for the number of states in it (as a function of k). (10 Points)

    ii) Build a NFA for L that contains at most $k^2 + 1$ states. (10 Points)