Introduction to the Theory of Computation
Jean Gallier

Homework 2
July 12, 2017; Due July 19, 2017
Beginning of class

“A problems” are for practice only, and should not be turned in.

Problem A1. Recall that two regular expressions $R$ and $S$ are equivalent, denoted as $R \equiv S$, iff they denote the same regular language $L[R] = L[S]$. Show that the following identities hold for regular expressions:

- $R^{**} \equiv R^*$
- $(R + S)^* \equiv (R^* + S^*)^*$
- $(R + S)^* \equiv (R^*S)^*^*$
- $(R + S)^* \equiv (R^*S)^*R^*$

Problem A2. Recall that a homomorphism $h : \Sigma^* \rightarrow \Delta^*$ is a function such that $h(uv) = h(u)h(v)$ for all $u, v \in \Sigma^*$. Given any language, $L \subseteq \Sigma^*$, we define $h(L)$ as

$h(L) = \{h(w) \mid w \in L\}$.

Prove that if $L \subseteq \Sigma^*$ is a regular language, then so is $h(L)$.

Problem A3. Construct an NFA accepting the language $L = \{aa, aaa\}^*$. Apply the subset construction to get a DFA accepting $L$.

“B problems” must be turned in.

Problem B1 (30 pts). Let $\Sigma = \{a_1, \ldots, a_n\}$ be an alphabet of $n$ symbols.

- (a) Construct an NFA with $2n + 1$ states accepting the set $L_n$ of strings over $\Sigma$ such that, every string in $L_n$ has an odd number of $a_i$, for some $a_i \in \Sigma$. Equivalently, if $L_i^n$ is the set of all strings over $\Sigma$ with an odd number of $a_i$, then $L_n = L_1^n \cup \cdots \cup L_n^n$.
- (b) Prove that there is a DFA with $2^n$ states accepting the language $L_n$.
- (c) Prove that every DFA accepting $L_n$ has at least $2^n$ states.

Hint: If a DFA $D$ with $k < 2^n$ states accepts $L_n$, show that there are two strings $u, v$ with the property that, for some $a_i \in \Sigma$, $u$ contains an odd number of $a_i$’s, $v$ contains an even
number of \( a_i \)'s, and \( D \) ends in the same state after processing \( u \) and \( v \). From this, conclude that \( D \) accepts incorrect strings.

**Problem B2 (30 pts).** (a) Let \( T = \{0, 1, 2\} \), let \( C \) be the set of 20 strings of length three over the alphabet \( T \),
\[
C = \{ u \in T^3 \mid u \notin \{110, 111, 112, 101, 121, 011, 211\} \},
\]
let \( \Sigma = \{0, 1, 2, c\} \) and consider the language
\[
L_M = \{ w \in \Sigma^* \mid w = u_1cu_2\cdots cu_n, \ n \geq 1, u_i \in C \}.
\]
Prove that \( L \) is regular.

(b) The language \( L_M \) has a geometric interpretation as a certain subset of \( \mathbb{R}^3 \) (actually, \( \mathbb{Q}^3 \)), as follows: Given any string, \( w = u_1cu_2\cdots cu_n \in L_M \), denoting the \( j \)th character in \( u_i \) by \( u^j_i \), where \( j \in \{1, 2, 3\} \), we obtain three strings
\[
\begin{align*}
w^1 &= u^1_1u^1_2\cdots u^1_n \\
w^2 &= u^2_1u^2_2\cdots u^2_n \\
w^3 &= u^3_1u^3_2\cdots u^3_n.
\end{align*}
\]
For example, if \( w = 012c001c222c122 \) we have \( w^1 = 0021 \), \( w^2 = 1022 \), and \( w^3 = 2122 \). Now, a string \( v \in T^+ \) can be interpreted as a decimal real number written in base three! Indeed, if
\[
v = b_1b_2\cdots b_k, \quad \text{where} \quad b_i \in \{0, 1, 2\} = T \ (1 \leq i \leq k),
\]
we interpret \( v \) as \( n(v) = 0.b_1b_2\cdots b_k \), i.e.,
\[
n(v) = b_13^{-1} + b_23^{-2} + \cdots + b_k3^{-k}.
\]
Finally, a string, \( w = u_1cu_2\cdots cu_n \in L_M \), is interpreted as the point, \( (x_w, y_w, z_w) \in \mathbb{R}^3 \), where
\[
x_w = n(w^1), \ y_w = n(w^2), \ z_w = n(w^3).
\]
Therefore, the language, \( L_M \), is the encoding of a set of rational points in \( \mathbb{R}^3 \), call it \( M \). This turns out to be the rational part of a fractal known as the Menger sponge.

**Extra Credit (20 points).** Write a computer program to draw the Menger sponge (based on the ideas above).

**Problem B3 (40 pts).** Let \( D = (Q, \Sigma, \delta, q_0, F) \) be a deterministic finite automaton. Define the relations \( \approx \) and \( \sim \) on \( \Sigma^* \) as follows:
\[
\begin{align*}
x &\approx y \quad \text{if and only if} \quad \text{for all} \quad p \in Q, \\
\delta^*(p, x) &\in F \quad \text{iff} \quad \delta^*(p, y) \in F,
\end{align*}
\]

and
\[ x \sim y \text{ if and only if, for all } p \in Q, \delta^*(p, x) = \delta^*(p, y). \]

(a) Show that \( \approx \) is a left-invariant equivalence relation and that \( \sim \) is an equivalence relation that is both left and right invariant. (A relation \( R \) on \( \Sigma^* \) is \textit{left invariant} iff \( uRv \) implies that \( wuRwv \) for all \( w \in \Sigma^* \), and \( R \) is \textit{right invariant} iff \( uRv \) implies that \( uwRvw \) for all \( w \in \Sigma^* \).)

(b) Let \( n \) be the number of states in \( Q \) (the set of states of \( D \)). Show that \( \approx \) has at most \( 2^n \) equivalence classes and that \( \sim \) has at most \( n^n \) equivalence classes.

(c) Given any language \( L \subseteq \Sigma^* \), define the relations \( \lambda_L \) and \( \mu_L \) on \( \Sigma^* \) as follows:
\[ u \lambda_L v \text{ iff, for all } z \in \Sigma^*, zu \in L \text{ iff } zv \in L, \]
and
\[ u \mu_L v \text{ iff, for all } x, y \in \Sigma^*, xuy \in L \text{ iff } xvy \in L. \]

Prove that \( \lambda_L \) is left-invariant, and that \( \mu_L \) is left and right-invariant. Prove that if \( L \) is regular, then both \( \lambda_L \) and \( \mu_L \) have a finite number of equivalence classes.

\textit{Hint:} Show that the number of classes of \( \lambda_L \) is at most the number of classes of \( \approx \), and that the number of classes of \( \mu_L \) is at most the number of classes of \( \sim \).

\textbf{Problem B4 (10 pts).} Is the following language regular? Justify your answer.
\[ L_3 = \{ a^n \mid n \text{ is a prime number} \} \]

\textbf{TOTAL: 110 + 20 points.}