Spring, 2009 CIS 511

Introduction to the Theory of Computation
Jean Gallier

Homework 2

February 5, 2009; Due February 19, 2009

"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 $R \cong S$, iff they denote the same regular language $\mathcal{L}[R] = \mathcal{L}[S]$. Show that the following identities hold for regular expressions:

$$\begin{aligned} R^{**} &\cong R^* \\ (R+S)^* &\cong (R^* + S^*)^* \\ (R+S)^* &\cong (R^* S^*)^* \\ (R+S)^* &\cong (R^* S)^* R^* \end{aligned}$$

Problem A2. Recall that a homomorphism $h\colon \Sigma^* \to \Delta^*$ 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 (25 pts). Let $\Sigma = \{a_1, \ldots, a_n\}$ be an alphabet of n symbols.

(a) Construct an NFA with $2n+1$ (or $2n$) states accepting the set L_n of strings over Σ such that, every string in L_n has an odd number of a_i, for some $a_i \in \Sigma$. Equivalently, if L_n^i is the set of all strings over Σ with an odd number of a_i, then $L_n = L_n^1 \cup \cdots \cup L_n^n$.

(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 (25 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

$$L_M = \{w \in \Sigma^* \mid w = u_1 c u_2 c \cdots c u_n,\ n \geq 1, u_i \in C\}.$$

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, $w = u_1 c u_2 c \cdots c u_n \in L_M$, denoting the jth character in u_i by u_i^j, where $j \in \{1, 2, 3\}$, we obtain three strings

$$\begin{aligned} w^1 &= u_1^1 u_2^1 \cdots u_n^1 \\ w^2 &= u_1^2 u_2^2 \cdots u_n^2 \\ w^3 &= u_1^3 u_2^3 \cdots u_n^3. \end{aligned}$$

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

$$v = b_1 b_2 \cdots b_k, \quad \text{where} \quad b_i \in \{0, 1, 2\} = T\ \ (1 \leq i \leq k),$$

we interpret v as $n(v) = 0.b_1 b_2 \cdots b_k$, i.e.,

$$n(v) = b_1 3^{-1} + b_2 3^{-2} + \cdots + b_k 3^{-k}.$$

Finally, a string, $w = u_1 c u_2 c \cdots c u_n \in L_M$, 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*.

Explain the best you can what are the recursive rules to create the Menger sponge, starting from a unit cube in $\mathbb{R}^3$. Draw some pictures illustrating this process and showing approximations of the Menger sponge.

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

Problem B3 (50 pts). Recall that for two regular expressions R and S, $R \cong S$ means that $\mathcal{L}[R] = \mathcal{L}[S]$, i.e. R and S denote the same language.

(i) Show that if $R \cong S^*T$, then $R \cong SR + T$.

(ii) Assume that ϵ (the empty string) is not in the language $\mathcal{L}[S]$ denoted by the regular expression S.

Show that if $R \cong SR + T$, then $R \cong S^*T$.

Hint: Prove that $x \in \mathcal{L}[R]$ if and only if $x \in \mathcal{L}[S^*T]$, by observing that $R \cong SR + T$ implies that for every $k \geq 0$,

$$R \cong S^{k+1}R + (S^k + S^{k-1} + \ldots + S^2 + S + \epsilon)T,$$

and that since $\epsilon \notin \mathcal{L}[S]$, every string in $\mathcal{L}[S^{k+1}R]$ has length at least $k + 1$.

(iii) Consider the following system of equations where $X_1, \ldots, X_n$ are variables standing for regular expressions, and the $S_{i,j}$ and the T_i are regular expressions:

$$\begin{aligned} X_1 &\cong S_{1,1}X_1 + \cdots + S_{1,n}X_n + T_1, \\ \cdots &\cong \cdots \\ X_n &\cong S_{n,1}X_1 + \cdots + S_{n,n}X_n + T_n. \end{aligned}$$

If $\epsilon \notin \mathcal{L}[S_{i,j}]$ for all i, j, $1 \leq i, j \leq n$, prove that this system has a unique solution. Do you see any connection between this problem and the node elimination algorithm?

Hint: Eliminate the X_i one by one.

Problem B4 (20 pts). Let R be any regular language over some alphabet Σ. Prove that the language

$$L = \{u \mid \exists v \in \Sigma^*,\, uv \in R,\, |u| = |v|\}$$

is regular

Problem B5 (60 pts). let Σ be an alphabet. For any language L and any string $x \in \Sigma^*$, the *left derivative of* L *w.r.t.* x, denoted by $x\backslash L$, or $D_x L$, or $\dfrac{dL}{dx}$, is the language

$$D_x L = \{y \in \Sigma^* \mid xy \in L\}.$$

(1) Prove the following identities for all languages L, A, B over Σ:

$$\begin{aligned} D_{xy}L &= D_y(D_x L), \\ D_\epsilon L &= L, \\ D_x(A \cup B) &= D_x A \cup D_x B, \end{aligned}$$

and for every symbol $a \in \Sigma$,

$$\begin{aligned} D_a(AB) &= (D_a A)B \cup (A \cap \{\epsilon\})D_a B, \\ D_a(L^*) &= (D_a L)L^*. \end{aligned}$$

Given a regular expression R and a string $x \in \Sigma^*$, we define the (left) derivative $D_x R$ of R w.r.t. x so that

$$\mathcal{L}[D_x R] = D_x \mathcal{L}[R].$$

We let

$$D_\epsilon R = R \quad \text{and} \quad D_{xa} R = D_a(D_x R)$$

where $a \in \Sigma$ and $x \in \Sigma^*$,

$$D_a \emptyset = \emptyset, \quad D_a \epsilon = \emptyset, \quad D_a a = \epsilon, \quad D_a b = \emptyset,$$

for all $a, b \in \Sigma$, $a \neq b$,

$$\begin{aligned} D_a((R+S)) &= (D_a R + D_a S), \\ D_a(R^*) &= (D_a R\, R^*), \\ D_a(RS) &= \begin{cases} (D_a R\, S) & \text{if } \epsilon \notin \mathcal{L}[R], \\ ((D_a R\, S) + D_a S) & \text{if } \epsilon \in \mathcal{L}[R], \end{cases} \end{aligned}$$

where R, S are any regular expressions.

(2) Give a simple algorithm to decide whether $\epsilon \in \mathcal{L}[R]$, where R is any given regular expression.

Prove that every regular expression has finitely many distinct derivatives (by distinct derivatives, we mean inequivalent derivatives).

Hint. Use an induction on the number of occurrences of the symbols from $\Sigma \cup \{\epsilon, \emptyset, +, \cdot, *\}$. When $R = (S \cdot T)$, prove that $D_x R$ is equivalent to an expression of the form

$$(D_x S\, T + D_{v_1} T + \cdots + D_{v_k} T),$$

where for every i, $1 \leq i \leq k$, there is some $u_i \in \mathcal{L}[S]$ such that $x = u_i v_i$. When $R = S^*$, prove that $D_x R$ is equivalent to an expression of the form

$$(D_{v_1} S + \cdots + D_{v_k} S) S^*,$$

where for every i, $1 \leq i \leq k$, there is some $u_i \in \mathcal{L}[S^*]$ such that $x = u_i v_i$.

(3) Assuming that R has n distinct derivatives, prove that every derivative of R belongs to the finite set

$$\{D_x R \mid x \in \Sigma^*,\ 0 \leq |x| < n\}.$$

Show that the upper bound on the number of derivatives is a product of towers of exponentials (in terms of the length of R).

(4) Prove that if D is a DFA accepting $\mathcal{L}[R]$ and D has n states, then R has at most n distinct derivatives.

If $\nu(R)$ is the number of occurrences in R of the symbols from $\Sigma \cup \{\epsilon, \emptyset, +, \cdot, *\}$, prove that R has at most

$$2^{2\nu(R)} \leq 4^{|R|}$$

distinct derivatives (where $|R|$ denotes the length of R).

(5) If L is any regular language over Σ^*, prove that the number of states of every minimal DFA for L is equal to the number of distinct derivatives, $D_u(L)$, of L.

(6) Prove that the regular expression

$$R = (a+b)^* a \underbrace{(a+b)\cdots(a+b)}_{n}$$

has $\nu(R) = 3n + 5$ (if we do not count $\cdot$, otherwise, $\nu(R) = 4n + 6$) and that R has 2^{n+1} distinct derivatives.

Prove that there is a 2-state DFA accepting the language denoted by $(a+b)^* a$ and there is an $(n+2)$-state DFA accepting the language denoted by $\underbrace{(a+b)\cdots(a+b)}_{n}$.

Yet, prove that any minimal DFA for the language denoted by R above has 2^{n+1} states.

TOTAL: 180 + 20 points.