Spring, 2004 CIS 511

Introduction to the Theory of Computation
Jean Gallier

Homework 2

February 3, 2004; Due February 19, 2004

"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

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

Given any language $L' \subseteq \Delta^*$, we define $h^{-1}(L')$ as

$$h^{-1}(L') = \{w \in \Sigma^* \mid h(w) \in L'\}.$$

Prove that if $L \subseteq \Sigma^*$ and $L' \subseteq \Delta^*$ are regular languages, then so are $h(L)$ and $h^{-1}(L')$.

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 (50 pts). An *a-transducer* (or *nondeterministic sequential transducer with accepting states*) is a sextuple $M = (K, \Sigma, \Delta, \lambda, q_0, F)$, where K is a finite set of states, Σ is a finite input alphabet, Δ is a finite output alphabet, $q_0 \in K$ is the start (or initial) state, $F \subseteq K$ is the set of accepting (of final) states, and

$$\lambda \subseteq K \times \Sigma^* \times \Delta^* \times K$$

is a finite set of quadruples called the *transition function* of M.

An a-transducer defines a binary relation between Σ^* and Δ^*, or equivalently, a function $M\colon \Sigma^* \to 2^{\Delta^*}$. We can explain what this function is by describing how an a-transducer makes a sequence of moves from configurations to configurations. The current configuration of an a-transducer is described by a triple $(p, u, v) \in K \times \Sigma^* \times \Delta^*$, where p is the current state, u

is the remaining input, and v is some ouput produced so far. We define the binary relation $\vdash_M$ on $K \times \Sigma^* \times \Delta^*$ as follows: For all $p, q \in K$, $u, \alpha \in \Sigma^*$, $\beta, v \in \Delta^*$, if $(p, u, v, q) \in \lambda$, then

$$(p,\, u\alpha,\, \beta) \vdash_M (q,\, \alpha,\, \beta v).$$

Let $\vdash_M^*$ be the transitive and reflexive closure of $\vdash_M$.

The function $M\colon \Sigma^* \to 2^{\Delta^*}$ is defined such that for every $w \in \Sigma^*$,

$$M(w) = \{y \in \Delta^* \mid (q_0,\, w,\, \epsilon) \vdash_M^* (f,\, \epsilon,\, y),\ f \in F\}.$$

For every language $L \subseteq \Sigma^*$, let

$$M(L) = \bigcup_{w \in L} M(w).$$

(a) Let $\Sigma = \Delta = \{a, b\}$. Construct an a-transducer swapping a's and b's (for instance, if $w = abbaa$, then $y = baabb$).

(b) Given an a-transducer $M = (K, \Sigma, \Delta, \lambda, q_0, F)$, define the new alphabet T as follows:

$$T = \{[p, u, v, q] \mid (p, u, v, q) \in \lambda\}.$$

Let $f\colon T^* \to \Sigma^*$ and $g\colon T^* \to \Delta^*$ be the homomorphisms defined such that

$$f([p, u, v, q]) = u, \quad \text{and} \quad g([p, u, v, q]) = v.$$

Prove that the language

$$\begin{aligned} R = \{&[q_0, u_1, v_1, q_1][q_1, u_2, v_2, q_2] \cdots [q_{n-2}, u_{n-1}, v_{n-1}, q_{n-1}][q_{n-1}, u_n, v_n, q_n] \\ &\mid [q_{i-1}, u_i, v_i, q_i] \in T,\ 1 \le i \le n,\ q_n \in F,\ n \ge 1\} \cup \{\epsilon \mid q_0 \in F\} \end{aligned}$$

is a regular language.

(c) Prove that

$$\begin{aligned} f^{-1}(L) \cap R = \{&[q_0, u_1, v_1, q_1][q_1, u_2, v_2, q_2] \cdots [q_{n-2}, u_{n-1}, v_{n-1}, q_{n-1}][q_{n-1}, u_n, v_n, q_n] \\ &\mid [q_{i-1}, u_i, v_i, q_i] \in T,\ u_1 u_2 \cdots u_n \in L,\ q_n \in F,\ n \ge 1\} \cup \{\epsilon \mid q_0 \in F,\ \epsilon \in L\}. \end{aligned}$$

(d) Prove that

$$M(L) = g(f^{-1}(L) \cap R).$$

If $\mathcal{L}$ is a family of languages closed under intersection with regular languages, homomorphic images, and inverse homomorphic images, is $\mathcal{L}$ closed under a-transductions? (Justify your answer).

If L is a regular language, is $M(L)$ regular? (Justify your answer).

(e) If M is an a-transducer from Σ^* to Δ^* prove that for any regular language, $L' \subseteq \Delta^*$, the language $M^{-1}(L')$ is also regular (see the definition of $M^{-1}(L')$ in the class notes).

Problem B6 (60 pts). (*Free generation of regular expressions*) The definition of the set $\mathcal{R}(\Sigma)$ of regular expressions over an alphabet Σ can be formalized in the following way: First, define the new alphabet

$$\Delta = \Sigma \cup \{(,), +, \cdot, *, \epsilon, \emptyset\}.$$

Let $C_+ \colon \Delta^* \times \Delta^* \to \Delta^*$, $C_\cdot \colon \Delta^* \times \Delta^* \to \Delta^*$, and $C_* \colon \Delta^* \to \Delta^*$ be the functions defined so that

$$\begin{aligned} C_+(u,v) &= (u+v) \\ C_\cdot(u,v) &= (u \cdot v) \\ C_*(u) &= u*, \end{aligned}$$

for all $u, v \in \Delta^*$. Let

$$\begin{aligned} \mathcal{R}(\Sigma)_0 &= \Sigma \cup \{\epsilon, \emptyset\} \\ \mathcal{R}(\Sigma)_{n+1} &= \mathcal{R}(\Sigma)_n \cup \{C_+(u,v),\ C_\cdot(u,v),\ C_*(u) \mid u, v \in \mathcal{R}(\Sigma)_n\}, \end{aligned}$$

and finally, let

$$\mathcal{R}(\Sigma) = \bigcup_{n \geq 0} \mathcal{R}(\Sigma)_n.$$

We wish to prove that the functions C_+, $C_\cdot$, C_* are injective when restricted to $\mathcal{R}(\Sigma)$, which means that if

$$C_+(u,v) = C_+(u',v')$$

for any $u, v, u', v' \in \mathcal{R}(\Sigma)$, then $u = u'$ and $v = v'$, similarly for $C_\cdot$, and if

$$C_*(u) = C_*(u')$$

for any $u, u' \in \mathcal{R}(\Sigma)$, then $u = u'$. We also wish to prove that the sets $C_+(\mathcal{R}(\Sigma), \mathcal{R}(\Sigma))$, $C_\cdot(\mathcal{R}(\Sigma), \mathcal{R}(\Sigma))$, and $C_*(\mathcal{R}(\Sigma))$, are pairwise disjoint.

For this, we introduce the "head deficiency function", K, defined as follows:

$$\begin{aligned} K(+) &= -1 \\ K(\cdot) &= -1 \\ K(*) &= 0 \\ K(a) &= 1 \quad (a \in \Sigma) \\ K(\emptyset) &= 1 \\ K(\epsilon) &= 1 \\ K(\text{``(''}) &= 1 \\ K(\text{``)''}) &= -1. \end{aligned}$$

This function is extended to Δ^+ in the obvious way, i.e.,

$$K(w_1 \cdots w_k) = K(w_1) + \cdots + K(w_k),$$

for all $w_i \in \Delta$ and all $k \geq 1$.

(i) Prove the following properties:

(a) For any regular expression $R \in \mathcal{R}(\Sigma)$, we have $K(R) = 1$.

(b) For any proper suffix S of a regular expression, we have $K(S) \leq 0$.

(c) No proper suffix S of a regular expression is a regular expression.

(ii) Using the above, prove that the restrictions of the functions C_+, $C_\cdot$, C_* to $\mathcal{R}(\Sigma)$ are injective and that the sets $C_+(\mathcal{R}(\Sigma), \mathcal{R}(\Sigma))$, $C_\cdot(\mathcal{R}(\Sigma), \mathcal{R}(\Sigma))$, and $C_*(\mathcal{R}(\Sigma))$, are pairwise disjoint.

(iii) Prove that $\mathcal{R}(\Sigma)_{n+1} \neq \mathcal{R}(\Sigma)_n$ for all $n \geq 0$, and that $C_+(u, v) \notin \mathcal{R}(\Sigma)_n$, $C_\cdot(u, v) \notin \mathcal{R}(\Sigma)_n$, and $C_*(u) \notin \mathcal{R}(\Sigma)_n$, for all $u, v \in \mathcal{R}(\Sigma)_n - \mathcal{R}(\Sigma)_{n-1}$ and for all $n \geq 0$ (setting $\mathcal{R}(\Sigma)_{-1} = \emptyset$).

(iv) Recall that the set $R(\Sigma)$ of regular languages over Σ is defined inductively as follows:

$$R(\Sigma)_0 = \{\{a_1\}, \ldots, \{a_m\}, \{\epsilon\}, \emptyset\},$$

where $\Sigma = \{a_1, \ldots, a_m\}$,

$$R(\Sigma)_{n+1} = R(\Sigma)_n \cup \{L_1 \cup L_2,\ L_1 \cdot L_2,\ L^* \mid L_1, L_2, L \in R(\Sigma)_n\},$$

and

$$R(\Sigma) = \bigcup_{n \geq 0} R(\Sigma)_n.$$

The interpretation of regular expressions as regular languages is given by the function, $\mathcal{L}\colon \mathcal{R}(\Sigma) \to R(\Sigma)$, defined recursively as follows:

$$\begin{aligned}
\mathcal{L}[a_i] &= \{a_i\} \\
\mathcal{L}[\epsilon] &= \{\epsilon\} \\
\mathcal{L}[\emptyset] &= \emptyset \\
\mathcal{L}[(R_1 + R_2)] &= \mathcal{L}[R_1] \cup \mathcal{L}[R_2] \\
\mathcal{L}[(R_1 \cdot R_2)] &= \mathcal{L}[R_1] \cdot \mathcal{L}[R_2] \\
\mathcal{L}[R^*] &= (\mathcal{L}[R])^*.
\end{aligned}$$

Prove that the function $\mathcal{L}$ is indeed well-defined.

Hint. Define a sequence of functions, $\mathcal{L}_n\colon \mathcal{R}(\Sigma)_n \rightarrow R(\Sigma)$, by induction using (ii) and (iii), and let $\mathcal{L} = \bigcup_{n\geq 0} \mathcal{L}_n$. You will have to make sense of all of this.

(v) (Regular expressions in prefix notation) Define the new alphabet

$$\Delta = \Sigma \cup \{+, \cdot, *, \epsilon, \emptyset\}.$$

Let $C_+\colon \Delta^* \times \Delta^* \rightarrow \Delta^*$, $C_{\cdot}\colon \Delta^* \times \Delta^* \rightarrow \Delta^*$, and $C_*\colon \Delta^* \rightarrow \Delta^*$ be the functions defined so that

$$\begin{aligned} C_+(u, v) &= +uv \\ C_{\cdot}(u, v) &= \cdot\, uv \\ C_*(u) &= *u, \end{aligned}$$

for all $u, v \in \Delta^*$. Let

$$\begin{aligned} \mathcal{R}(\Sigma)_0 &= \Sigma \cup \{\epsilon, \emptyset\} \\ \mathcal{R}(\Sigma)_{n+1} &= \mathcal{R}(\Sigma)_n \cup \{C_+(u, v),\ C_{\cdot}(u, v),\ C_*(u) \mid u, v \in \mathcal{R}(\Sigma)_n\}, \end{aligned}$$

and finally, let

$$\mathcal{R}(\Sigma) = \bigcup_{n\geq 0} \mathcal{R}(\Sigma)_n.$$

Define the “tail deficiency function”, K, as before:

$$\begin{aligned} K(+) &= -1 \\ K(\cdot) &= -1 \\ K(*) &= 0 \\ K(a) &= 1 \quad (a \in \Sigma) \\ K(\emptyset) &= 1 \\ K(\epsilon) &= 1, \end{aligned}$$

and extend it to Δ^+ in the obvious way. Redo questions (i)–(iv) for regular expressions in prefix notation.

(vi) This time, consider the alphabet

$$\Delta = \Sigma \cup \{+, \cdot, *, \epsilon, \emptyset\}$$

and the functions $C_+\colon \Delta^* \times \Delta^* \rightarrow \Delta^*$, $C_{\cdot}\colon \Delta^* \times \Delta^* \rightarrow \Delta^*$, and $C_*\colon \Delta^* \rightarrow \Delta^*$ defined so that

$$\begin{aligned} C_+(u, v) &= u + v \\ C_{\cdot}(u, v) &= u \cdot v \\ C_*(u) &= u*, \end{aligned}$$

for all $u, v \in \Delta^*$.

Show that properties (b) and (c) of (i) fail, that (ii) also fails, and that $\mathcal{L}$ cannot be defined properly.

(vii) **Extra credit (20 pts).** Consider the alphabet

$$\Delta = \Sigma \cup \{), +, \cdot, *, \epsilon, \emptyset\}$$

and the functions $C_+ \colon \Delta^* \times \Delta^* \to \Delta^*$, $C_\cdot \colon \Delta^* \times \Delta^* \to \Delta^*$, and $C_* \colon \Delta^* \to \Delta^*$ defined so that

$$\begin{aligned} C_+(u, v) &= u + v) \\ C_\cdot(u, v) &= u \cdot v) \\ C_*(u) &= u*, \end{aligned}$$

for all $u, v \in \Delta^*$.

Redo questions (i)–(iv) for these strange regular expressions!

Problem B7 (40 pts). Let $D = (Q, \Sigma, \delta, q_0, F)$ be a deterministic finite automaton. Define the relations $\approx$ and $\sim$ on Σ^* as follows:

$$\begin{gathered} x \approx y \quad \text{if and only if,} \quad \text{for all} \quad p \in Q, \\ \delta^*(p, x) \in F \quad \text{iff} \quad \delta^*(p, y) \in F, \end{gathered}$$

and

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

(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 Σ^* is *left invariant* iff uRv implies that $wuRwv$ for all $w \in \Sigma^*$, and R is *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 λ_L and μ_L on Σ^* as follows:

$$u \, \lambda_L \, v \quad \text{iff,} \quad \text{for all} \quad z \in \Sigma^*, \quad zu \in L \quad \text{iff} \quad zv \in L,$$

and

$$u \, \mu_L \, v \quad \text{iff,} \quad \text{for all} \quad x, y \in \Sigma^*, \quad xuy \in L \quad \text{iff} \quad xvy \in L.$$

Prove that λ_L is left-invariant, and that μ_L is left and right-invariant. Prove that if L is regular, then both λ_L and μ_L have a finite number of equivalence classes.

Hint: Show that the number of classes of λ_L is at most the number of classes of $\approx$, and that the number of classes of μ_L is at most the number of classes of $\sim$.

TOTAL: 270 + 40 points.