Spring, 2010 CIS 511

Introduction to the Theory of Computation
Jean Gallier

Homework 4

March 4, 2010; Due March 23, 2010

Do either Problem B1 or Problem B2.

Do Problems B3, B4, B5 and B6.

"B problems" must be turned in.

Problem B1 (60 pts). Let $f(X)$ be a polynomial of degree $d \geq 1$ with integer coefficients,

$$f(X) = c_0X^d + c_1X^{d-1} + \cdots + c_{d-1}X + c_d,$$

and let $\alpha \in \mathbb{R}$ be some (real) root of $f(X)$, which means that $f(\alpha) = 0$. For example, $\sqrt[3]{2}$ is a root of $f(X) = X^3 - 2$. Let D be any real number such that $D > d$ and consider the rational numbers (fractions), m/n, with $m \in \mathbb{Z}$ and $n \geq 1$, that satisfy the inequality

$$\left|\frac{m}{n} - \alpha\right| \leq \frac{1}{n^D}. \tag{$*$}$$

(a) Prove that if $f(m/n) \neq 0$, then

$$\left|f\left(\frac{m}{n}\right)\right| \geq \frac{1}{n^d}.$$

Hint. Both the numerator and the denominator of $f(m/n)$ are nonzero integers.

(b) Since $f(\alpha) = 0$, we know from algebra that there is a unique polynomial, $g(X)$, of degree $d-1$, such that

$$f(X) = (X - \alpha)g(X).$$

Prove that there is a constant, $K > 0$, independent of m/n, such that if m/n satisfies $(*)$, then

$$\left|g\left(\frac{m}{n}\right)\right| \leq K.$$

(c) Deduce from (a) and (b) that if m/n satisfies $(*)$ and $f(m/n) \neq 0$, then

$$\frac{1}{n^d} \leq \left|f\left(\frac{m}{n}\right)\right| = \left|\frac{m}{n} - \alpha\right| \cdot \left|g\left(\frac{m}{n}\right)\right| \leq \frac{K}{n^D}$$

and thus, that

$$n < K^{\frac{1}{D-d}}.$$

(d) Prove that the language

$$L_D = \{a^m b^n \mid m/n \,\text{satisfies}\, (*),\ m \geq 0,\ n \geq 1,\ \gcd(m,n) = 1\}$$

is a regular language.

(e) Consider the following number, β, given by its decimal expansion

$$\begin{array}{rcl} \text{digit}: & & \begin{array}{l} \quad\ \ {\scriptstyle 1\,2}\quad\ {\scriptstyle 6}\qquad\qquad\qquad\qquad\qquad\ \ {\scriptstyle 24}\qquad\qquad\ {\scriptstyle 120} \\ \quad\ \downarrow\downarrow\ \ \downarrow\qquad\qquad\qquad\qquad\qquad\ \ \downarrow\qquad\qquad\ \ \downarrow \end{array} \\ \beta & = & 0.11000100000000000000000100\cdots 00100\cdots \end{array}$$

where the n^{th} "one" appears as the $n!^{\text{th}}$ decimal digit and all the other digits are zeros. Equivalently,

$$\begin{aligned} \beta &= \frac{1}{10} + \frac{1}{10^2} + \frac{1}{10^6} + \frac{1}{10^{24}} + \frac{1}{10^{120}} + \frac{1}{10^{720}} + \cdots \\ &= \sum_{n=1}^{\infty} \frac{1}{10^{n!}}. \end{aligned}$$

For every $n \geq 1$, consider the rational approximation, r_n, of β given by

$$r_n = \sum_{k=1}^{n} \frac{1}{10^{k!}}.$$

Prove that

$$\beta - r_n < \frac{2}{10^{(n+1)!}} = \frac{2}{(10^{n!})^{n+1}},$$

and thus, that

$$\beta - r_n < \frac{1}{(10^{n!})^n}.$$

Deduce from this that for every $D \geq 1$, there are infinitely many fractions, m/n, with $m \geq 0$ and $n \geq 1$, $\gcd(m,n) = 1$, such that

$$\left|\frac{m}{n} - \beta\right| \leq \frac{1}{n^D} \qquad (**)$$

and consequently, that every language

$$L'_D = \{a^m b^n \mid m/n \,\text{satisfies}\, (**),\ m \geq 0,\ n \geq 1,\ \gcd(m,n) = 1\}$$

is infinite.

Remark: I do not know whether L'_D is regular but I suspect it is not!

(f) Use (d) and (e) to prove that the number β is not the root of any polynomial with integer coefficients. We say that β is a *transcendental number*. The number β was discovered by Joseph Liouville in 1840. The numbers e and π are also transcendental but that's a lot harder to prove!

Problem B2 (60 pts). This problem is based on the method proved correct in Problem B4 of Homework 4.

Given a DFA $D = (Q, \Sigma, \delta, q_0, F)$, for any two states $p, q \in Q$, a fast algorithm for computing the forward closure of the relation $R = \{(p, q)\}$, or detecting a bad pair of states, can be obtained as follows: An equivalence relation on Q is represented by a partition Π. Each equivalence class C in the partition is represented by a tree structure consisting of nodes and (parent) pointers, with the pointers from the sons of a node to the node itself. The root has a null pointer. Each node also maintains a counter keeping track of the number of nodes in the subtree rooted at that node.

Two functions *union* and *find* are defined as follows. Given a state p, $find(p, \Pi)$ finds the root of the tree containing p as a node (not necessarily a leaf). Given two root nodes p, q, $union(p, q, \Pi)$ forms a new partition by merging the two trees with roots p and q as follows: if the counter of p is smaller than that of q, then let the root of p point to q, else let the root of q point to p.

In order to speed up the algorithm, we can modify *find* as follows: during a call $find(p, \Pi)$, as we follow the path from p to the root r of the tree containing p, we redirect the parent pointer of every node q on the path from p (including p itself) to r.

Say that a pair $\langle p, q\rangle$ is *bad* iff either both $p \in F$ and $q \notin F$, or both $p \notin F$ and $q \in F$. The function *bad* is such that $bad(\langle p, q\rangle) = true$ if $\langle p, q\rangle$ is bad, and $bad(\langle p, q\rangle) = false$ otherwise.

For details of this implementation of partitions, see *Fundamentals of data structures*, by Horowitz and Sahni, Computer Science press, pp. 248-256.

Then, the algorithm is as follows:

```
function unif[p, q, Π, dd]: flag;
  begin
    trans := left(dd); ff := right(dd); pq := (p, q); st := (pq); flag := 1;
    k := Length(first(trans));
    while st ≠ () ∧ flag ≠ 0 do
      uv := top(st); uu := left(uv); vv := right(uv);
      pop(st);
      if bad(ff, uv) = 1 then flag := 0
      else
        u := find(uu, Π); v := find(vv, Π);
        if u ≠ v then
          union(u, v, Π);
          for i = 1 to k do
            u1 := delta(trans, uu, k − i + 1); v1 := delta(trans, vv, k − i + 1);
            uv := (u1, v1); push(st, uv)
          endfor
        endif
      endif
    endwhile
  end
```

The initial partition Π is the identity relation on Q, i.e., it consists of blocks $\{q\}$ for all state $q \in Q$. The algorithm uses a stack st. We are assuming that the DFA dd is specified by a list of two sublists, the first list, denoted $left(dd)$ in the pseudo-code above, being a representation of the transition function, and the second one, denoted $right(dd)$, the set of final states. The transition function itself is a list of lists, where the i-th list represents the i-th row of the transition table for dd. The function $delta$ is such that $delta(trans, i, j)$ returns the j-th state in the i-th row of the transition table of dd. For example, we have a DFA

$$dd = (((2, 3), (2, 4), (2, 3), (2, 5), (2, 3), (7, 6), (7, 8), (7, 9), (7, 6)), (5, 9))$$

consisting of 9 states labeled $1, \ldots, 9$, and two final states 5 and 9. Also, the alphabet has two letters, since every row in the transition table consists of two entries. For example, the two transitions from state 3 are given by the pair $(2, 3)$, which indicates that $\delta(3, a) = 2$ and $\delta(3, b) = 3$.

Implement the above algorithm, and test it at least for the above DFA dd and the pairs of states $(1, 6)$ and $(1, 7)$. Pay particular attention to the input and output format. Explain your data structures.

Please, consult the instructions posted on the web page for CIS511, Homework section, for instructions on the format for the input and output for this computer program.

Extra Credit (up to **80 pts).** Implement your program in such a way that it displays the simultaneous parallel forward moves in the DFA and the updating of the trees representing the blocks of the partition.

Problem B3 (50 pts). Prove that the language

$$L = \{a^{4n+3} \mid 4n + 3 \text{ is prime}\}$$

is not regular.

Hint. First, you will have to prove that there are infinitely many primes of the form $4n+3$. The list of such primes begins with

$$3,\ 7,\ 11,\ 19,\ 23,\ 31,\ 43 \cdots$$

Say we already have $n+1$ of these primes, denoted by

$$3,\ p_1,\ p_2, \cdots, p_n,$$

where $p_i > 3$. Consider the number

$$m = 4p_1p_2 \cdots p_n + 3.$$

If $m = q_1 \cdots q_k$ is a prime factorization of m, prove that $q_j > 3$ for $j = 1, \ldots k$ and that no q_j is equal to any of the p_i's. Prove that one of the q_j's must be of the form $4n+3$, which shows that there is a prime of the form $4n+3$ greater than any of the previous primes of the same form.

Problem B4 (50 pts). The purpose of this problem is to get a fast algorithm for testing state equivalence in a DFA. Let $D = (Q, \Sigma, \delta, q_0, F)$ be a deterministic finite automaton. Recall that *state equivalence* is the equivalence relation $\equiv$ on Q, defined such that,

$$p \equiv q \quad \text{iff} \quad \forall z \in \Sigma^*(\delta^*(p, z) \in F \quad \text{iff} \quad \delta^*(q, z) \in F).$$

and that *i-equivalence* is the equivalence relation $\equiv_i$ on Q, defined such that,

$$p \equiv_i q \quad \text{iff} \quad \forall z \in \Sigma^*,\ |z| \leq i\ (\delta^*(p, z) \in F \quad \text{iff} \quad \delta^*(q, z) \in F).$$

A relation $S \subseteq Q \times Q$ is a *forward closure* iff it is an equivalence relation and whenever $(p, q) \in S$, then $(\delta(p, a), \delta(q, a)) \in S$, for all $a \in \Sigma$.

We say that a forward closure S is *good* iff whenever $(p, q) \in S$, then *good*(p, q), where *good*(p, q) holds iff either both $p, q \in F$, or both $p, q \notin F$.

Given any relation $R \subseteq Q \times Q$, recall that the smallest equivalence relation $R_{\approx}$ containing R is the relation $(R \cup R^{-1})^*$ (where $R^{-1} = \{(q, p) \mid (p, q) \in R\}$, and $(R \cup R^{-1})^*$ is the reflexive and transitive closure of $(R \cup R^{-1})$). We define the sequence of relations $R_i \subseteq Q \times Q$ as follows:

$$\begin{aligned} R_0 &= R_{\approx} \\ R_{i+1} &= (R_i \cup \{(\delta(p, a), \delta(q, a)) \mid (p, q) \in R_i,\ a \in \Sigma\})_{\approx}. \end{aligned}$$

(i) Prove that $R_{i_0+1} = R_{i_0}$ for some least i_0. Prove that R_{i_0} is the smallest forward closure containing R.

We denote the smallest forward closure R_{i_0} containing R as $R^{\dagger}$, and call it the *forward closure of* R.

(ii) Prove that $p \equiv q$ iff the forward closure $R^{\dagger}$ of the relation $R = \{(p, q)\}$ is good.

Problem B5 (70 pts). (i) Prove that the conclusion of the pumping lemma holds for the following language L over $\{a, b\}^*$, and yet, L is **not** regular!

$$L = \{w \mid \exists n \geq 1, \exists x_i \in a^+, \exists y_i \in b^+, 1 \leq i \leq n, n \text{ is not prime}, w = x_1 y_1 \cdots x_n y_n\}.$$

(ii) Consider the following version of the pumping lemma. For any regular language L, there is some $m \geq 1$ so that for every $y \in \Sigma^*$, if $|y| = m$, then there exist $u, x, v \in \Sigma^*$ so that

(1) $y = uxv$;

(2) $x \neq \epsilon$;

(3) For all $z \in \Sigma^*$,

$$yz \in L \quad \text{iff} \quad ux^i vz \in L$$

for all $i \geq 0$.

Prove that this pumping lemma holds.

(iii) Prove that the converse of the pumping lemma in (ii) also holds, i.e., if a language L satisfies the pumping lemma in (ii), then it is regular.

(iv) Consider yet another version of the pumping lemma. For any regular language L, there is some $m \geq 1$ so that for every $y \in \Sigma^*$, if $|y| \geq m$, then there exist $u, x, v \in \Sigma^*$ so that

(1) $y = uxv$;

(2) $x \neq \epsilon$;

(3) For all $\alpha, \beta \in \Sigma^*$,

$$\alpha u \beta \in L \quad \text{iff} \quad \alpha u x^i \beta \in L$$

for all $i \geq 0$.

Prove that this pumping lemma holds.

(v) Prove that the converse of the pumping lemma in (iv) also holds, i.e., if a language L satisfies the pumping lemma in (iv), then it is regular.

Problem B6 (60 pts). Let $D = (Q, \Sigma, \delta, q_0, F)$ be a *trim* DFA. Consider the following procedure:

(1) Form an NFA, N^R, by reversing all the transitions of D, i.e., there is a transition from p to q on input $a \in \Sigma$ in N iff $\delta(q, a) = p$ in D.

(2) Apply the subset construction to the NFA, N^R, obtained in (1), taking the start state to be the set F. The final states of the DFA obtained by applying the subset construction to N^R are all the subsets containing q_0. Then, trim the resulting DFA, to obtain the DFA D^R.

Observe that $L(D^R) = L(D)^R$.

Now, apply the above procedure to D, getting D^R, and apply this procedure again, to get D^{RR}. Prove that D^{RR} is a minimal DFA for $L = L(D)$.

Hint. First prove that if δ_R is the transition function of D^R, then for every $w \in \Sigma^*$ and for every state, $T \subseteq Q$, of D^R,

$$\delta_R^*(T, w) = \{q \in Q \mid \delta^*(q, w^R) \in T\}.$$

TOTAL: 290 points.