Computer Science 236 Fall | Nov. 11, 2010
St. George Campus | University of Toronto

Assignment 3
Due Date: 2nd December, 2010

1. (10 marks) Assume that you are given a file of arbitrary length that contains student records for CSC236. Each line of the file has the following format:

   LastName,FirstName,Average

   There are no spaces and commas are the separators on each line. Assume that the only characters in the file are upper case letters, numerals (0-9) and commas and that the file is sorted by decreasing average. You need to find some information out about the students in the class. The easiest way to do this is to write a regular expression that matches the information you need (much like grep in Unix). If possible, construct a regular expression to extract the information. If no such regular expression exists, prove why not. [Note: you may use the notation $\backslash a$ to represent any alphabetic symbol and $\backslash n$ to represent any numeric symbol]

   (a) (2 marks) All those people whose last name begins with A, B, C or D.
   Soln:

   $$(A + B + C + D)\backslash a^*, \backslash a^*, \backslash n^*$$

   If you were more specific about the number of digits, i.e., ensuring that the name is non-empty or that the average is 2 or 3 digits that is fine as well.

   (b) (2 marks) All those people who are passing the course.
   Soln:

   $$\backslash a^*, \backslash a^*, ((5 + 6 + 7 + 8 + 9)\backslash n + 100)$$

   (c) (2 marks) All those people who are failing the course.
   Soln:

   $$\backslash a^*, \backslash a^*, ((1 + 2 + 3 + 4)\backslash n + \backslash n)$$

   (d) (4 marks) Determine whether there are more people passing the course than failing the course. [Hint: notice that the entries are sorted by from largest to smallest average]
   Soln:
   We can first use two variables to represent the regular expressions that denote all those people passing the course and all those failing the course, respectively (we know what these expressions are from (b) and (c)).
   Let F represent failing and P represent passing. Then what we want is an expression to represent all those files that satisfy L where $L = \{P^n F^m \ : \ n > m\}$ Can this be done?
   No. We can prove this using the Pumping Lemma.
   Proof by contradiction. Assume that there exists some regular expression that denotes L, then there must also exist some FSA to accept L. Assume that such an FSA has j

states. Now consider some file f that has i people passing, where $i > j$ and $i-1$ people failing. This file will be accepted by the DFA since it satisfies the requirements of more people passing than failing. We split f into three pieces as required by the pumping lemma: $f = uvw$.

We don't know the exact decomposition of f, but by the lemma $uv \leq j$ and v is nonempty. By the pumping lemma, files of the form uv^kw must also be in L. Now take $k = 0$, the string uw must also be in L and since v is non-empty this implies that the DFA accepts a string with either an equal number of people passing or fewer people passing than failing contradicting the definition of L. Therefore, L is not regular.

2. (16 marks) For each of the following, give a regular expression that denotes the set of strings as well as a DFA that accepts it.

(a) $S_1 = \{x \in \{0,1\}^* \mid \text{neither 00 nor 11 appear as substrings of } x\}$
Soln:

$$(10)^*(1+\epsilon) + (01)^*(0+\epsilon)$$

OR

$$(10)^* + (10)^*1 + (01)^* + (01)^*0$$

(b) $S_2 = \{x \in \{0,1\}^* \mid \text{both 00 and 11 appear as substrings of } x\}$
Soln:

$$(1+0)^*[(00(1+0)^*11) + (11(0+1)^*00)](1+0)^*$$

(c) $S_3 = \{0^n1^m \mid n, m \geq 0 \text{ and } n+m \text{ is odd }\}$
Soln:

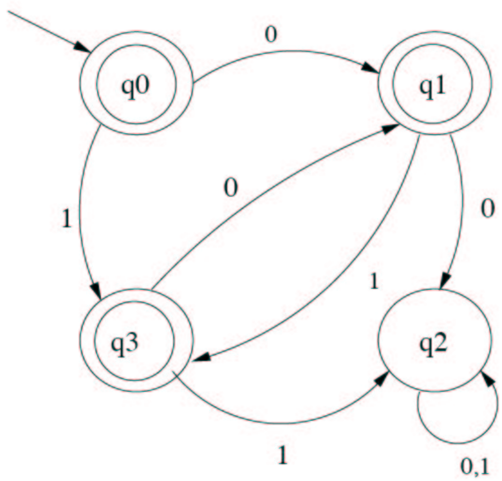
$$(00)^*(11)^*1 + 0(00)^*(11)^*$$

(d) $S_4 = \{x \in \{a,b,c\}^* \mid x \text{ starts and ends with the same letter}\}$.
Soln:

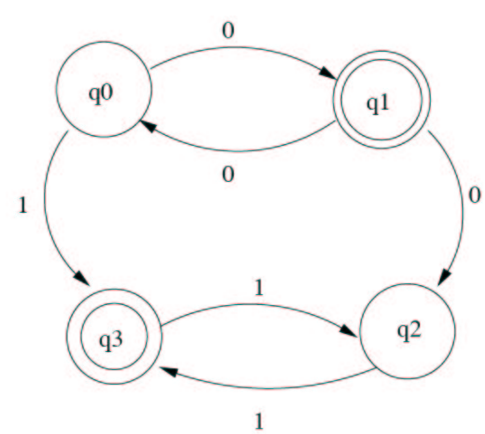
$$a(a+b+c)^*a + b(a+b+c)^*b + c(a+b+c)^*c$$

DFAs

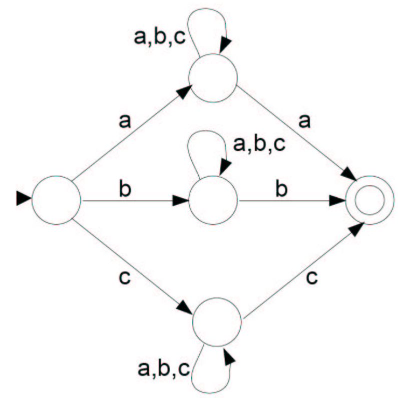
a)



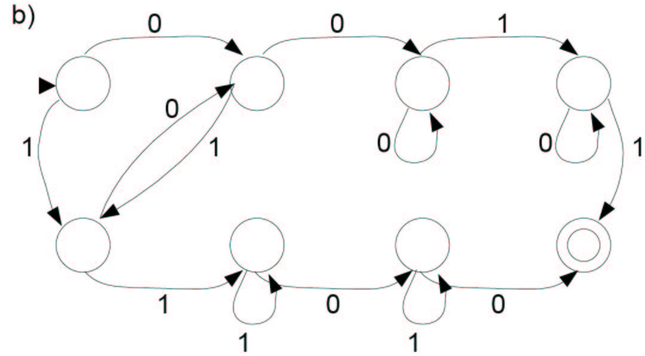
c)



d)



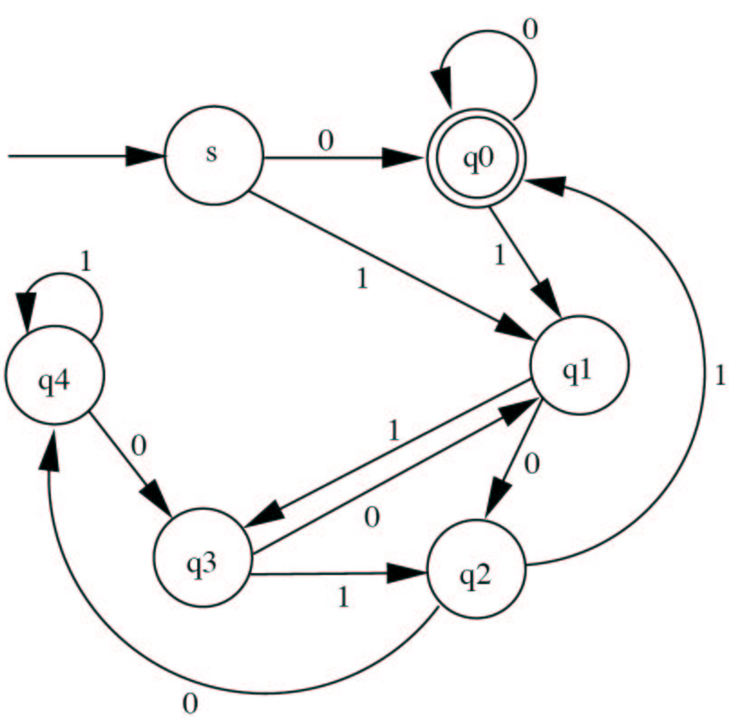
b)



3. (12 marks) Consider a set S of strings over the alphabet $\Sigma = \{0, 1\}$.

   (a) Construct a DFA that accepts $s \in S$ iff the decimal value of s is divisible by 5. For example, your DFA should accept 0, 101, 1010 and 1111 but it should not accept 10, 1001, 1011 etc.

   Soln:



(b) Use induction to prove that your DFA is correct.

Soln:

We will do this by structural induction (induction on the length of the string would be fine, too). Let $S(x)$ be:

$$S(x) : \delta^*(s, x) = \begin{cases} s & \text{iff } x \text{ is empty} \\ q_0 & \text{iff } x \equiv 0 \mod 5 \\ q_1 & \text{iff } x \equiv 1 \mod 5 \\ q_2 & \text{iff } x \equiv 2 \mod 5 \\ q_3 & \text{iff } x \equiv 3 \mod 5 \\ q_4 & \text{iff } x \equiv 4 \mod 5 \end{cases}$$

We must prove $S(x)$ for all strings in $\{0, 1\}^*$.

Base case: $x = \epsilon$. Then x ends at s and is not accepted as required.

Inductive step: Assume that $S(y)$ holds, i.e., that:

$$\delta^*(s, y) = \begin{cases} s & \text{iff } y \text{ is empty} \\ q_0 & \text{iff } y \equiv 0 \mod 5 \\ q_1 & \text{iff } y \equiv 1 \mod 5 \\ q_2 & \text{iff } y \equiv 2 \mod 5 \\ q_3 & \text{iff } y \equiv 3 \mod 5 \\ q_4 & \text{iff } y \equiv 4 \mod 5 \end{cases}$$

Let $x = ya$, where $a \in \{0, 1\}$. We will consider each case for $a = 0$ and $a = 1$, separately.

Case 1: $a = 0$. Then:

$$\delta^*(s, x) = \begin{cases} \delta(s, 0) & \text{iff } y \text{ is empty, i.e., } x \equiv 0 \mod 5 \\ \delta(q_0, 0) & \text{iff } y \equiv 0 \mod 5\text{, i.e., } x \equiv 0 \mod 5 \\ \delta(q_1, 0) & \text{iff } y \equiv 1 \mod 5\text{, i.e., } x \equiv 2 \mod 5 \\ \delta(q_2, 0) & \text{iff } y \equiv 2 \mod 5\text{, i.e., } x \equiv 4 \mod 5 \\ \delta(q_3, 0) & \text{iff } y \equiv 3 \mod 5\text{, i.e., } x \equiv 1 \mod 5 \\ \delta(q_4, 0) & \text{iff } y \equiv 4 \mod 5\text{, i.e., } x \equiv 3 \mod 5 \end{cases}$$

These hold because adding a low 0 bit multiplies the value of y by 2. By inspection of the DFA, we can see that:

$$\begin{aligned} \delta(s, 0) &= q_0 \\ \delta(q_0, 0) &= q_0 \\ \delta(q_1, 0) &= q_2 \\ \delta(q_2, 0) &= q_4 \\ \delta(q_3, 0) &= q_1 \\ \delta(q_4, 0) &= q_3 \end{aligned}$$

and thus the claim holds.

Case 2: $a = 1$. Then:

$$\delta^*(s,x) = \begin{cases} \delta(s,1) & \text{iff } y \text{ is empty, i.e., } x \equiv 1 \mod 5 \\ \delta(q_0,1) & \text{iff } y \equiv 0 \mod 5\text{, i.e., } x \equiv 1 \mod 5 \\ \delta(q_1,1) & \text{iff } y \equiv 1 \mod 5\text{, i.e., } x \equiv 3 \mod 5 \\ \delta(q_2,1) & \text{iff } y \equiv 2 \mod 5\text{, i.e., } x \equiv 0 \mod 5 \\ \delta(q_3,1) & \text{iff } y \equiv 3 \mod 5\text{, i.e., } x \equiv 2 \mod 5 \\ \delta(q_4,1) & \text{iff } y \equiv 4 \mod 5\text{, i.e., } x \equiv 4 \mod 5 \end{cases}$$

These hold because adding a low 1 bit multiplies the value of y by 2 and adds 1. By inspection of the DFA, we can see that:

$$\begin{aligned} \delta(s,1) &= q_1 \\ \delta(q_0,1) &= q_1 \\ \delta(q_1,1) &= q_3 \\ \delta(q_2,1) &= q_0 \\ \delta(q_3,1) &= q_2 \\ \delta(q_4,1) &= q_4 \end{aligned}$$

and thus the claim holds.

4. (6 marks) Show that if A is a regular language then A^R is as well, i.e., that the class of regular languages is closed under reversal.

Soln:

There are two possible ways to do this. One way is by induction, as follows. Let M represent the FSA for A and M^R the FSA for A^R if it exists.

Let S(n) be "There exists an M^R that accepts the language $A_n^R = \{x^R \ : \ |x| \leq n \ and \ x \in A\}$."

WTS: S(n) is true $\forall n \in \mathbb{N}$.

Define $A_i = \{x \ : \ |x| \leq i, x \in A\}$.

Base Case: n=0. Since $x = \epsilon$, $x \in A$ iff $x \in A_0^R$ and the single state machine can either accept or reject x depending on whether $x \in A$.

IH: Assume for arbitrary n that $S(j)$ is true for all j, $0 \leq j < n$.

Inductive Step: WTS S(n).

Notice that if $x = yz$ then $x^R = z^R y^R$, therefore, $A_n^R = A_i^R A_{n-i}^R$ where $y^R \in A_i^R$ and $z^R \in A_{n-i}^R$. Since $i < n$ and $n - i < n$, we can apply the inductive hypothesis, and each of these languages has a FSA that accepts them and therefore by the closure of concatenation, there exists a FSA to accept A_n^R.

Alternatively, we can show how to construct an NFSA for A^R.

Since A is a regular language, there exists some DFA M, that accepts it. Now consider the following method to construct a NFSA M' to accept A^R.

We keep M the same however first we reverse the direction of all the transitions.

So $\delta'(q', a) = \{q | \delta(q,a) = q'\}$.

We then turn q_0 in M into the final accepting state $F' = \{q_0\}$ of M'.

Then, we construct a new initial state q_0' for M' and define $\delta'(q_0', \epsilon) = \{F\}$, i.e, for each final accepting state in M we add an ϵ transition from q_0'.

5. (6 marks) A *palindrome* is a word that can be read forwards and backwards. For example, “bob”, “anna”, “kayak”, “radar”. Let Σ be the alphabet of lower case English letters. Consider the set P of *palindromes* from Σ^*. Let $\mathcal{L}$ be the language consisting of the set of strings in P. Determine whether $\mathcal{L}$ is regular and prove your claim either by giving a description of how to build a DFA to accept $\mathcal{L}$ or by constructing a contradiction.

   Soln:

   $\mathcal{L}$ is not regular. Suppose that it were. Then, by the pumping lemma, there is an n such that for any string, $s \in \mathcal{L}$, of length at least n, $s = uvw$, where $|uv| \leq n$, $v \neq \epsilon$ and for all $k \geq 0$, $uv^k w \in \mathcal{L}$.

   Let $s = b^n a b^n$. Clearly, $s \in \mathcal{L}$, and $|s| \geq n$. So there exists such a u, v and w and $|uv| \leq n$, and so consists entirely of bs. Because $v \neq \epsilon$, any $k > 0$ yields a $uv^k w = b^m a b^n$ for which $m > n$, and so is not in $\mathcal{L}$, which is a contradiction.

6. (15 marks)

   (a) (5 marks) Let $f, g : \mathbb{N} \to \mathbb{R}^{\geq 0}$. Prove that $f \in O(g) \Leftrightarrow g \in \Omega(f)$.
   (That is, prove $f \in O(g) \Rightarrow g \in \Omega(f)$ and prove $g \in \Omega(f) \Rightarrow f \in O(g)$.)
   Soln:
   We will first prove that $f \in O(g) \Rightarrow g \in \Omega(f)$. So assume that $f \in O(g)$. So let c and n_0 be positive numbers such that $f(n) \leq c \cdot g(n)$ for all integers $n \geq n_0$. So $g(n) \geq (1/c) \cdot f(n)$ for all integers $n \geq n_0$. So $g \in \Omega(f)$.
   We will now prove that $g \in \Omega(f) \Rightarrow f \in O(g)$. So assume that $g \in \Omega(f)$. So let c and n_0 be positive numbers such that $g(n) \geq c \cdot f(n)$ for all integers $n \geq n_0$. So $f(n) \leq (1/c) \cdot g(n)$ for all integers $n \geq n_0$. So $f \in O(g)$.

   (b) (5 marks) Prove or disprove the following Conjecture.
   Conjecture: For every $f, g : \mathbb{N} \to \mathbb{R}^{\geq 0}$, if $f \in O(g)$, then $|f - g| \in O(g)$.
   (We define $|f - g|$ to be that function that maps n to $|f(n) - g(n)|$.)
   Soln:
   We will prove the conjecture. Say that $f \in O(g)$. Let c and n_0 be positive numbers such that $f(n) \leq c \cdot g(n)$ for all integers $n \geq n_0$. It is not hard to show, in fact, that $|f(n) - g(n)| \leq c' \cdot g(n)$ for all $n \geq n_0$, where $c' = \max\{c, 1\}$. Note that $|f(n) - g(n)| = \max\{f(n) - g(n), g(n) - f(n)\}$, so it suffices to show that $f(n) - g(n) \leq c' \cdot g(n)$ and $g(n) - f(n) \leq c' \cdot g(n)$ for all $n \geq n_0$. But this is true because $f(n) - g(n) \leq f(n) \leq c' \cdot g(n)$, and $g(n) - f(n) \leq g(n) \leq c' \cdot g(n)$ for all $n \geq n_0$.

   (c) (5 marks) Define $f, g : \mathbb{N} \to \mathbb{R}^{\geq 0}$ by $f(n) = 100n^2 + 5n + 10$ and $g(n) =| n^3 - 100n^2 |$.
   Prove that $f \in O(g)$ by exhibiting particular constants c and n_0, and proving that $f(n) \leq c \cdot g(n)$ for all $n \geq n_0$.
   Soln:
   Let $f(n) = 100n^2 + 5n + 10$ and $g(n) = |n^3 - 100n^2|$. We will show that $f(n) \leq 2 \cdot g(n)$ for every $n \geq 200$. So let $n \geq 200$ be an integer. We have $n^3 \geq 200n^2$,

so $g(n) \geq 100n^2$, so $200n^2 \leq 2g(n)$. So it suffices to show that $f(n) \leq 200n^2$, and thus that $5n + 10 \leq 100n^2$. Since $n \geq 5$, we have $5n \leq n^2$ and $10 \leq n^2$, so $5n + 10 \leq 2n^2 \leq 100n^2$.

7. (10 marks) Consider the following recurrence for a function T that takes on nonnegative values and is defined on integers ≥ 1:

$$T(n) \leq \begin{cases} 5 & \text{if } 1 \leq n \leq 4 \\ T(\lfloor \frac{7}{10} n \rfloor) + T(\lfloor \frac{1}{5} n \rfloor) + 3n & \text{if } n > 4 \end{cases}$$

Prove that $T(n)$ is $O(n)$. Note that it doesn't help to use the general theorem about divide-and-conquer recurrences. You should present a particular constant c and prove that $T(n) \leq c \cdot n$ for all $n \geq 1$.

Motivation: This recurrence actually comes up in the following situation. Say we wish to find the median of n distinct elements, using as few comparisons as possible. More generally, say we wish to find the k-th smallest of n elements. We could sort the n numbers but this takes about $n \log n$ comparisons. Instead we do the following which, by the theorem proven in this question, uses $O(n)$ comparisons:

Divide the elements up into groups of 5 (don't worry now about what to do if n isn't divisible by 5); find the median of each group of 5, using a *linear* in n number of comparisons in total. This gives us about $n/5$ median points. Recursively find the median of these median points using about $T(n/5)$ comparisons; call this element b. Using a *linear* in n number of comparisons, compare every element to b; let S_1 be the set of elements $\leq b$, and let S_2 be the set of elements $> b$. Because of the way b was chosen, each of S_1 and S_2 must have at most $\frac{7}{10}n$ elements. If $k \leq |S_1|$, then recursively find the k-th smallest element of S_1, otherwise recursively find the $(k - |S_1|)$-th smallest element of S_2; this takes at most $T(\frac{7}{10}n)$ comparisons.

Soln:

Let $c = 30$. (Note that we chose c by going through the proof and seeing what value of c would work.)
Let $P(n)$ be: "$T(n) \leq c \cdot n$." We will use complete induction to prove that $P(n)$ holds for every integer $n \geq 1$.

Let $i \geq 1$ be an integer such that $P(j)$ holds for every integer j, $1 \leq j < i$. We will show $P(i)$.

CASE 1: $1 \leq i \leq 4$. For each such i, $T(i) = 5$. Since $c \geq 5$, it is easy to check that $T(i) \leq c \cdot i$ for each such i.

CASE 2: $i \geq 5$. So we have $1 = \lfloor (1/5)5 \rfloor \leq \lfloor (1/5)i \rfloor \leq \lfloor (7/10)i \rfloor \leq (7/10)i < i$. So $P(\lfloor (7/10)i \rfloor)$ and $P(\lfloor (1/5)i \rfloor)$ hold, so $T(\lfloor (7/10)i \rfloor) \leq c\lfloor (7/10)i \rfloor \leq c(7/10)i$ and $T(\lfloor (1/5)i \rfloor) \leq c\lfloor (1/5)i \rfloor \leq c(1/5)i$. We have $T(i) \leq T(\lfloor (7/10)i \rfloor) + T(\lfloor (1/5)i \rfloor) + 3i \leq c(7/10)i + c(1/5)i + 3i = ((9/10)c + 3)i = ((9/10)30 + 3)i = 30i \leq c \cdot i$. So $P(i)$ holds.