CS 383
Exam 1 Solution
March 7, 2007

Note: Problems #1 and #5 are worth 20 points each, the other four problems are worth 15 points each.

1. Which of the following are regular languages? You do not need to justify your answers. Be sure that I can tell what your answer is; either write "Yes" or "No" next to each, or say something like "a, b, and c are regular, the others are not."
   a) Strings of a's, b's, and c's that have at most 3 a's and at most 2 b's, but any number of c's (with letters in any order).
   b) Strings where the number of a's and the number of b's have the same parity: either both numbers are even or both numbers are odd.
   c) Strings that start and end with the same letter.
   d) Strings of odd length, whose middle letter is "o". Two such strings are "bob" and "stops".
   e) Strings of even length whose first half is all 0's and whose second half is any combination of 0's and 1's.

   a, b, and c are regular, d and e are not.

   a: To see that strings with at most 3 a's make a regular language, just have states that represent 0, 1, 2, 3, or 4 a's. The first 4 are terminal, the 4-a state is not. Do the same with the 2-b language, then take the intersection of these regular languages.

   b: For this you just need 2 states, representing the same parity or different parities; you toggle between them on an a or a b.

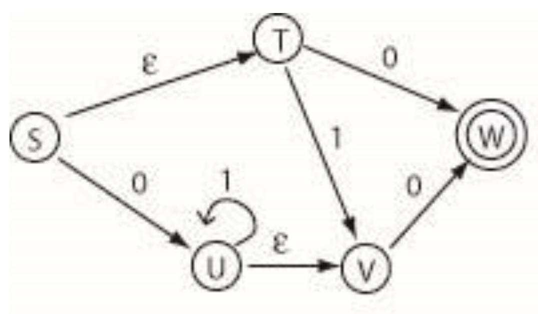
   c: For the {0,1} alphabet this is 1(1+0)*1 + 0(1+0)*0

   d: This is silly. How could a DFA ever find the middle of a string? A pumping lemma argument with string $w=a^p o a^p$ would work here – any way to pump changes the middle from an "o" to an "a".

   e: Take the string $w=0^p 1^p$. This has all 0's in the first half, but if we take a decomposition w=xyz and pump y 0 times, to get xz, the result will have some 1's in the first half.

2. Give an ε-NFA that accepts strings denoted by the regular expression $(0^*11)^*1$.

3. Convert the following ε-NFA to a DFA.



4. Use the pumping lemma to show that the set of strings of 0's and 1's with more 1's than 0's is not regular.

Suppose this language is regular; let p be its pumping constant. Let w = 0^p1^{p+1}. This string certainly is longer than the pumping constant. Consider any decomposition w=xyz where |xy|<=p. The y string consists of a positive number of 0's. If we pump once: xy^2z contains at least as many 0's as 1's and so is not in the language. Since w is sufficiently long and no decomposition of it can be pumped, the language cannot be regular.

5. Suppose a regular language is accepted by a DFA with p states.
   a) Show that if the language includes a string of length p or more then the language contains infinitely many strings.[Hint: This isn't very deep. Think of a FAMOUS LEMMA.]

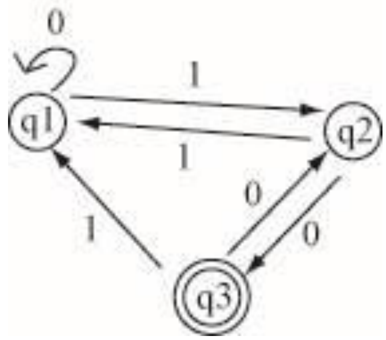
      The number of states is just the pumping constant. If string w is at least this long it can be pumped; each xy^kz produces a different string, so the language is infinite.

   b) Show that if the language has no string of length between p and 2p then the language is finite.

      If the language is infinite it must contain some string w whose length is more than p and this string can be pumped: there is a decomposition w=xyz, where xz must be in the language. We get xz from w by removing y, which has up to p symbols. If xz has length more than p it too can be shortened by up to p symbols and remain in the language. This can continue as long as necessary until the result has length less than p. Since each of these shortenings removes at most p symbols, there must be one such string whose length is between p and 2p (you can't jump over this whole range if no jump is longer than p). So if there is no string in the language whose length is between p and 2p, then there is no string in the language whole length is p or more, and the language must be finite.

Note that parts (a) and (b) together give an algorithm for determining if a regular language is finite or infinite.

6. Consider the following automaton:



Find the regular expressions r^k_{ij} for k = 0 and k=1.

Note that $r^1_{ij}=r^0_{ij}+r^0_{i1}(r^0_{11})^*r^0_{1j}$, and $r^0_{11}=0^*$.

	k = 0	k = 1
r^k_{11}	**0+ε**	(0+ε)+(0+ε)0*(0+ε)=0*
r^k_{12}	**1**	1+0*1 = 0*1
r^k_{13}	**ϕ**	ϕ
r^k_{21}	**1**	1+10*(0+ε)=10*
r^k_{22}	**ε**	ε+10*1
r^k_{23}	**0**	0+ ϕ=0
r^k_{31}	**1**	1+10*=10*
r^k_{32}	**0**	0+10*1
r^k_{33}	**ε**	ε+ϕ=ε