15-251 : Great Theoretical Ideas In Computer Science

Fall 2014

Assignment 9

Due: **Thursday**, November 13, 2014 11:59 PM

Name: _____

Andrew ID: _____

Question:	1	2	3	4	5	Total
Points:	20	20	15	20	25	100
Score:						

0. Warmup

(a) Build a DFA for each of the following languages ($\Sigma = \{0, 1\}$). *Try to keep the number of states to a minimum.*

1. $L = \{w \mid w \text{ contains at least three 1s}\}$
2. $L = \{w \mid w \text{ contains at exactly three 1's}\}$
3. $L = \{w \mid w \text{ ends in a 01}\}$
4. $L = (0^*101)^*$

(b) DFAs cannot count but they can verify addition. Consider an alphabet with rather interesting symbols:

$$\Sigma = \left\{ \begin{pmatrix}1\\1\\1\end{pmatrix}, \begin{pmatrix}1\\1\\0\end{pmatrix}, \begin{pmatrix}1\\0\\1\end{pmatrix}, \begin{pmatrix}1\\0\\0\end{pmatrix}, \begin{pmatrix}0\\1\\1\end{pmatrix}, \begin{pmatrix}0\\1\\0\end{pmatrix}, \begin{pmatrix}0\\0\\1\end{pmatrix}, \begin{pmatrix}0\\0\\0\end{pmatrix} \right\}$$

These symbols represent columns which could appear when writing the addition and result of two binary numbers. An input of length n represents the addition of two n-bit numbers where the input is given from least to most significant bit. The first number's binary representation is in the top row, the second number's representation is in the middle row, and the result is in the bottom row. Let L be the language of strings over this alphabet that represent correct addition statements as described above. As an example, correctly representing $3 + 4 = 7$ in this scheme would give the following string (remember that it is ordered least significant bit to most, so the first input symbol is on the left):

$$\begin{pmatrix}1\\0\\1\end{pmatrix} \begin{pmatrix}1\\0\\1\end{pmatrix} \begin{pmatrix}0\\1\\1\end{pmatrix}$$

The following represents $5 + 2 = 6$, and should therefore not be in L:

$$\begin{pmatrix}1\\0\\0\end{pmatrix} \begin{pmatrix}0\\1\\1\end{pmatrix} \begin{pmatrix}1\\0\\1\end{pmatrix}$$

Draw a DFA which accepts L. Try not to use more than 5 states.

1. Square roots

Let $n = pq$ where p and q are distinct odd primes.

(10) (a) Let $a \in Z_n^*$ be a square. Prove that there are 4 distinct square roots $r_1, r_2, r_2, r_4 \in Z_n^*$ of a, i.e., $r_i^2 = a$ for $i = 1, 2, 3, 4$.

Solution:

(10) (b) Suppose you have a black box that takes input $b \in Z_n^*$ and (correctly) returns whether b is a square or not, and if b is a square also arbitrarily returns one of its square roots.

Turns out this is a pretty neat black box to have. Give an efficient algorithm that succeeds in computing the prime factors p, q of n with 99% probability based on a few calls to the black box.

Solution:

2. Regular or Not

Determine whether the following languages are regular. If it's regular, show a DFA for it; if it's not regular, prove that fact.

(For the first two parts, the alphabet is $\{0, 1\}$, for the last one it is $\{a\}$.)

(5) (a) $L = \{w|$ every instance of "101" in w is immediately followed by "10"$\}$. Hence $\epsilon, 1, 10110, 10010010110110$ are all accepting strings, whereas $1010, 1010010, 101101101$ are not.

Solution:

(5) (b) $L = \{w|$ the third to last digit in w is a $0\}$

Solution:

(10) (c) $L = \{a^p | p$ is prime$\}$.

Solution:

3. Regular Expressions

Regular expressions are a way of representing a set of strings. We define a regular expressions inductively as follows:

- Any single string A is a regular expression representing the singleton set containing A.
- If A and B are regular expressions, then $(A+B)$ is a regular expression representing the union of A and B.
- If A and B are regular expressions, then (AB) is a regular expression representing the concatenation $\{xy \mid x \in A, y \in B\}$ of A and B.
- If A is a regular expression, then $A*$ is the regular expression that gives the Kleene Closure of the set described by A.

As you may have deduced from the name, a language is regular if and only if it can be represented by a regular expression. For this problem, draw DFAs that accept the regular languages represented by the following regular expressions, over the alphabet $\{0, 1\}$.

(5) (a) ((01) + (10))*

Solution:

(10) (b) ((01)* + (10)*) + (0(1)* + 1(0)*)

Solution:

4. Let's mix modular arithmetic and DFAs

Suppose we want to design a very fast algorithm to determine whether some long binary number is divisible by k. One way to do this is by building a class of DFAs (one for each k). We will read in the number starting with the *most significant* bit first, down to the least significant bit. The DFA should accept if the number is divisible by k and reject otherwise.

(10) (a) Give the class of DFAs that solves the problem and describe in a few sentences what the states in each DFA mean. What is the running time of this algorithm as a function of the number of bits n in the input?

Solution:

(10) (b) Now suppose the number is read with the *least significant* bit first, down to the most significant bit. Is there a DFA that solves the problem (accepts if the number is divisible by k and rejects otherwise)? Clearly justify your answer.

Solution:

5. Cardinality

(5) (a) Prove or disprove the following statement:
The union of countably many countable sets is countable.

Solution:

(5) (b) Prove or disprove the following statement:
The product of countably many countable sets is countable.

Solution:

(15) (c) Let us define $\mathbb{N}!$ to be the set of permutations of natural numbers $\mathbb{N}$. In particular, the elements of $\mathbb{N}!$ are the infinite sequences that contain every natural number exactly once. Prove that $\mathbb{N}!$ is uncountable. Note that a permutation requires two things: no numbers should repeat, and every number should be included.

Solution: