CSCI2243-Logic and Computation
Final Exam

December 20, 2016

It's a good idea to read all the questions before you start to work. Show all your work in the blue exam book. For Problems 3(b-e), and all parts of Problem 5, it is not enough to just give the answer, you also need to give some account of the reasoning by which you arrived at your answer.

Point values:
1: 15 (3 for each part)
2: 18 (3 for each part)
3: 15 (5 for each part)
4(a): 3, 4(b):3, 4(c): 9
5: 20 (5 for each part)
6: 12 (3 for part (a), and 3 for each description and regular expression in part (b))
7: 20 (5 for each part)
Total: 115

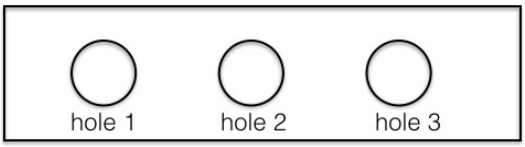

Figure 1: The pegboard for the puzzle in Problem 1

1 Propositional Logic

You have to place pegs in board with three holes arranged in a straight line, shown in Figure 1. The rules are that you must place exactly one red peg, and exactly one blue peg, in such a manner that the blue peg is not immediately to the right of the red peg (it is permissible to have a space between the blue and red pegs). We will model these rules by a formula of propositional logic in conjunctive normal form, such as we might create in preparing input to a SAT solver. As there is a fairly large number of variables and clauses, you will not have to write out the entire formula, but instead will answer questions about particular clauses in the specification.

There are exactly six variables: $r_1, r_2, r_3, b_1, b_2, b_3$. The variable r_1 is assigned the value true if and only if the red peg is placed in hole 1, and I am sure that you can now figure out what the other five variables mean.

(a) One of the clauses says that the red peg has to be placed somewhere. Write this clause. (Remember that a clause has to be an OR of literals.)

(b) One of the clauses is equivalent to

$$b_1 \rightarrow \neg b_2.$$

However, each clause in the CNF is required to be an OR of literals. Write the equivalent clause.

(c) What does the clause in *(b)* say in English?

(d) Write the clause or clauses expressing the condition that the blue peg is not immediately to the right of the red peg.

(e) The solution to the puzzle is a formula in disjunctive normal form, equivalent to the CNF specification, in which each disjunct represents a satisfying assignment. Write *one* of these disjuncts. How many satisfying assignments are there in all?

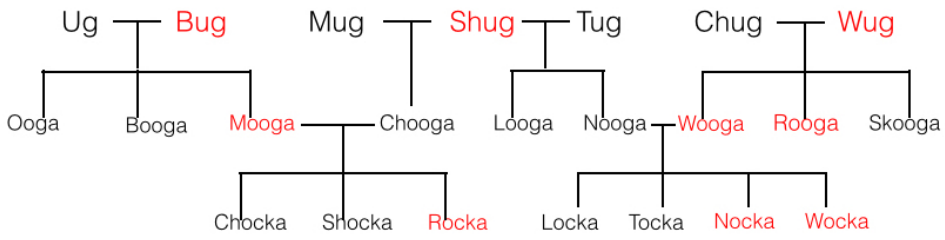

Figure 2: The Ug Family Tree for Problem 2.

2 Predicate Logic and Relations

Figure 2 shows the family tree of several generations of the Ug clan. Thus, for example, Chocka is the child of Mooga and Chooga.

Given such a family tree, containing a finite set X of individuals, we define a function

$$f : X \to \mathcal{P}(X),$$

where for each $x \in X$, $f(x)$ denotes the set of ancestors of x. We assume that each person is their own ancestor. Thus, for example,

$$f(\text{Chocka}) = \{\text{Chocka}, \text{Mooga.Chooga}, \text{Ug}, \text{Bug}, \text{Mug}, \text{Shug}\}.$$

(Note that Tug is *not* an ancestor of Chocka.)

We can describe a property of this family in three ways: by a sentence of English, by a sentence of predicate logic containing a relation symbol $A(x, y)$ which means that x is an ancestor of y, and by an equation or inequality involving sets and the function f.

In the table below, you are given a number of properties expressed in one or two of these languages, and your job is to fill in the blank spaces in the table, translating the condition into the other languages. The descriptions should work for any family tree with individuals having these names, not just the one pictured. (Observe that the property given in the third row of the table is not true in the pictured model.)

The expression in the bottom left cell may be a little tricky to figure out! The idea is that Chooga is a parent of Chocka if there is no one else in the chain of ancestors from Chocka to Chooga

Mug $\in f($Shocka$)$	Shocka is a descendant of Mug.	*(a)*
(b)	Wug has no ancestors other than himself	$\forall x(A(x,$ Wug$) \rightarrow x =$ Wug$)$
(c)	Chocka and Tocka have an ancestor in common.	*(d)*
$\{x \in X :$ Chooga $\in f(x)\} \cap f($Chocka$)$ $=$ {Chooga,Chocka}	Chooga is a parent of Chocka.	*(e)*

(f) What kind of relation is A? An equivalence relation? A partial order? A preorder that is not a partial order? None of these?

3 Sets, Functions, Cardinality

Let X be the set of nonempty subsets of $\{1, 2, 3, 4, 5, 6\}$ with exactly three elements. That is,

$$X = \{A \in \mathcal{P}(\{1, 2, 3, 4, 5, 6\}) : |A| = 3\}.$$

(a)Is $3 \in X$? Is $\{1, 2, 4\} \in X$? Is $\{1, 2, 3, 6\} \backslash \{2, 4, 5\} \in X$?

(b) What is $|X|$? Give the exact numerical answer, and explain how you got it.

(c) Consider the function $m : X \to \{1, 2, 3, 4, 5, 6\}$, where for each $A \in X$, $m(A)$ is the largest element of A. Is m one-to-one? Is m onto?

(d) Let m be the function defined in part *(c)*. What is $|m^{-1}(4)|$?

(e) Let

$$W = \{A \in \mathcal{P}(\mathbf{Z}^+) : |A| = 3\}.$$

Is there a one-to-one function $f : W \to \mathbf{Z}^+$?

4 Towers of Hanoi Revisited

This is a version of the Towers of Hanoi puzzle in which you have to move the tower of n disks from the first post to the third post. But in addition to the rule that you cannot put a larger disk on top of a smaller one, there is a new rule that says that one each move, you can only move a disk to an adjacent post: that is, we can move a disk between the first and the second posts, and between the second and the third posts, but we cannot move a disk from the first to the third post in a single move.

Here is a recursive solution to the puzzle with this new restriction: To move n disks from post 1 to post 3, use your strategy for $n-1$ disks to move the top $n-1$ disks from post 1 to post 3. Then move the largest disk from post 1 to post 2. Use your strategy for $n-1$ disks in reverse to move the $n-1$ disks from post 3 back to post 1. Then move the largest disk from post 2 to post 3. Use your strategy one last time to move $n-1$ disks from post 1 to post 3. (Python code implementing this solution is shown at the end of the exam.)

Let S_n denote the number of moves this strategy uses to move a tower of n disks.

(a) What are S_1, S_2 and S_3? (Use the description of the algorithm, not the formula in (c) below, to find these.)

(b) Write an equation giving the value of S_{n+1} in terms of S_n for $n \geq 1$. (Again, this should be based on the description above.)

(c) Use mathematical induction and the recurrence relation in (b) to prove that $S_n = 3^n - 1$ for all $n \geq 1$.

5 Some Questions about Eleven

All of the questions below can be answered with minimal computation. If you find yourself performing a lengthy calculation, you're probably missing the point.

Representing an integer at base eleven requires an additional digit to represent ten, so we will use A for this, just as we do in hexadecimal notation.

(a) What is the base eleven representation of the integer whose decimal representation is 111?

(b) Are there integers x and y such that $44x + 1111y = 11$? (You don't have to find them, just tell whether or not they exist, and why.)

(c) Find $10^{4975} \bmod 11$.

(d) For which values of n is the integer with decimal representation

$$1\underbrace{00\cdots0}_{n \text{ zeros}}1$$

divisible by eleven?

6 Regular Expressions and Automata

Consider the DFAs pictured in Figure 3.

(a) Find a string that is recognized by the DFA on the right, but not by the one on the left.

(b) The language recognized by the DFA on the right has a particularly simple description in English, and both languages are represented by fairly simple regular expressions. Give such a description for the right-hand language, and regular expressions for both languages.

7 Turing Machines

The Turing machine $\mathcal{M}$ in the accompanying diagram accepts all strings over the one-letter alphabet $\{a\}$ whose length is a power of two. The labeling follows our usual conventions for such diagrams: We only indicate a symbol to write on the tape if it is different from the symbol scanned, and any transition that is missing is assumed to lead to the reject state and move the reading head to the right. The initial state is state 0. (Observe that the blank symbol is denoted $\Box$ in the problem below, but by B in the diagram, which was automatically generated by software that requires the use of ordinary printable characters.)

The machine works by repeatedly scanning the input and keeping track of whether it has seen an even number or an odd number of a's, and whether the number of a's is greater than 1. On each scan, it crosses out every second a, thus dividing the number of a's by two. If, at the end of a scan, it has seen an odd number of a's greater than 1, it rejects; if it sees exactly one a, it accepts.

(a) What is the input alphabet of $\mathcal{M}$? What is the tape alphabet of $\mathcal{M}$?

(b) Let δ denote the next-state function of $\mathcal{M}$. Recall that δ is a function

$$\delta : (Q - \{q_{acc}, q_{rej}\}) \times \Gamma \to Q \times \Gamma \times \{L, R\},$$

where Q denotes the state set and Γ the tape alphabet. Give the values of $\delta(0, a)$, $\delta(0, x)$, and $\delta(0, \Box)$.

(c) Trace the execution of this machine on the input strings aa and aaa.

(d) Let L be the set of all strings of the form a^k, where k is a power of 2. Does $\mathcal{M}$ recognize L? Does it decide L? (You should assume that the description of the machine's operation given above is correct!)

HAPPY HOLIDAYS!

Here is Python code to print the solution of the modified Towers of Hanoi puzzle (Problem 4).

```
#The towers puzzle with the restriction that you can only
# move to an adjacent post.
#Setting direction =1 means move right from post A to post C,
#direction=-1 means move left from post C to post A.
def hanoi_variant(numtowers,direction):
    if direction== 1:
        src='A'
        dest='C'
    else:
        src ='C'
        dest='A'
    if numtowers>0:
        hanoi_variant(numtowers-1,direction)
        print 'Move disk from '+src+ ' to B.'
        hanoi_variant(numtowers-1,-direction)
        print 'Move disk from B to '+dest+'.'
        hanoi_variant(numtowers-1,direction)
```

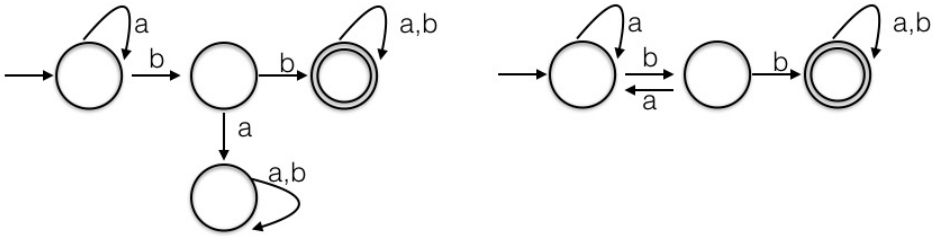

Figure 3: DFAs for Problem 7

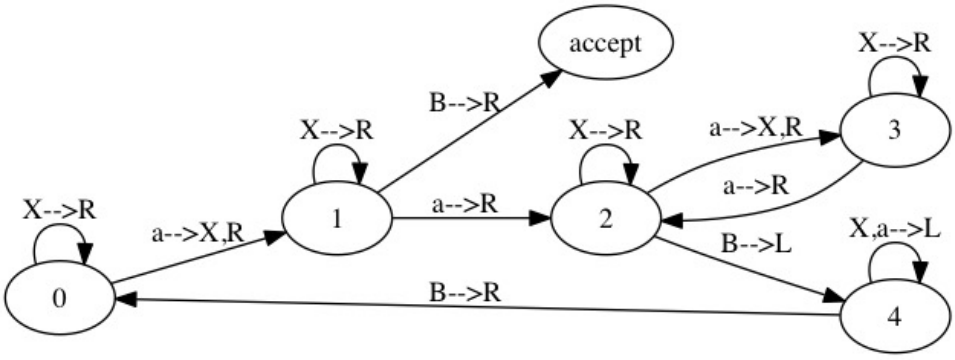

Figure 4: Turing machine for Problem 8