CS 70 Discrete Mathematics for CS
Spring 2000 Papadimitriou/Russell/Sinclair Final

You have three hours. The exam is open-book, open-notes. There are **seven** *questions in all, for a total of 100 points. The point score for each question part is indicated in the margin.*

Avoid spending too long on any one question. Length and degree of difficulty of questions may vary. Write your answers in blue books. Check you haven't skipped any by accident. Hand them all in. Panic not.

1. [16 pts] The resolution rule

The *resolution* inference rule for propositional logic applies to two *clauses*, $A = (a_1 \vee \cdots \vee a_m)$ and $B = (b_1 \vee \cdots \vee b_n)$, where the a_is and b_js are literals, and generates a third clause which is called a *resolvent* of A and B. The general pattern is as follows:

$$\frac{a_1 \vee \cdots \vee a_m \qquad b_1 \vee \cdots \vee b_n}{a_1 \vee \cdots \vee a_{i-1} \vee a_{i+1} \vee \cdots \vee a_m \vee b_1 \vee \cdots \vee b_{j-1} \vee b_{j+1} \vee \cdots \vee b_n}$$

where the literals a_i and b_j must be *complementary*, i.e., one is the negation of the other. The resolvent contains all the literals of the two input clauses except for the complementary literals a_i and b_j. Here is a concrete example:

$$\frac{a \vee b \vee \neg c \qquad b \vee c \vee \neg d}{a \vee b \vee b \vee \neg d}$$

(a) Show that modus ponens is a special case of resolution. **2pts**

(b) Let $A = (x \vee y)$ and $B = (\neg y \vee z)$, where x, y, and z are proposition symbols. Calculate C, the resolvent of A and B, and prove that $A \wedge B \models C$. **4pts**

(c) Prove that, for any two clauses, one of the following must hold: **4pts**

i. there are no possible resolvents; or

ii. there is exactly one resolvent; or

iii. all possible resolvents are logically equivalent to T.

(d) Prove that the resolution rule is *sound*, i.e., that for any clauses A and B, if C is a resolvent of A and B then $A \wedge B \models C$. **6pts**

[continued overleaf]

2. [10 pts] Russian Multiplication

The so-called *Russian algorithm* for multiplying two positive integers a, b is defined recursively as follows:

```
algorithm mult(a, b)
  if a = 1 then return(b)
  else return(mult(⌊a/2⌋, 2b) + b ⋆ (a mod 2))
```

Your task is to prove, using (strong) induction on the first input a, that the program correctly outputs the product ab.

(a) Write down the statement $P(a)$ that has to be proved by induction. **2pts**

(b) Prove the base case $P(1)$. **2pts**

(c) Prove $P(a)$ for all $a > 0$ by induction. **6pts**

3. [16 pts] The Candy Bar Problem

A candy bar of total length L is made up of a linear sequence of n equal-length blocks. Assume that n is odd. Suppose you cut the bar at one of the $L-1$ boundaries between two blocks chosen uniformly at random. Let the r.v. X be the length of the *longer* of the two resulting pieces.

(a) Compute $\mathrm{E}(X)$ in the case $n = 5$. **4pts**

(b) Compute $\mathrm{E}(X)$ as a function of n. Check your answer against the value you obtained in part (a). [Hint: You may use the fact that the sum of the first m positive integers is $\sum_{i=1}^{m} i = \frac{1}{2}m(m+1)$.] **4pts**

(c) Compute the variance $\mathrm{Var}(X)$ as a function of n. [Hint: You may use the fact that the sum of the squares of the first m positive integers is $\sum_{i=1}^{m} i^2 = \frac{1}{6}m(m+1)(2m+1)$.] **4pts**

(d) Use Chebyshev's inequality together with parts (b) and (c) to derive an upper bound on the probability that the longer piece has length at least $\frac{7L}{8}$. **4pts**

[continued overleaf]

4. [16 + 5 pts] The Maximum Satisfiability Problem

Recall the CNF Satisfiability problem: given a Boolean formula ϕ in CNF, is there a truth assignment to the variables of ϕ that makes ϕ true? In this problem we consider a variant of this problem, known as Maximum Satisfiability. Our task is to find a truth assignment that makes the *maximum possible number of clauses of* ϕ true. For example, in the formula

$$\phi = (a \vee \neg b) \wedge (\neg b \vee \neg a) \wedge (c \vee b) \wedge (b \vee \neg c)$$

it is possible to satisfy three of the clauses, but not all four.

(a) Let the number of different variables appearing in ϕ be n. Suppose we pick a truth assignment randomly, by flipping a fair coin independently for each variable. What is the size of the sample space, and what is the probability of each point? **4pts**

(b) Assume that ϕ has exactly two literals in every clause, and that no clause contains any variable more than once (as in the above example). If we pick the truth assignment randomly as in part (a), what is the probability that any given clause of ϕ will be satisfied? **4pts**

(c) If the number of clauses in ϕ is m, what is the *expected* number of clauses satisfied by our random assignment? **4pts**

(d) Prove that, for any Boolean formula obeying the assumptions in part (b), there must always exist a truth assignment that satisfies at least $\frac{3}{4}$ of the clauses. **4pts**

(e) **[Extra credit]** Suppose we pick a random truth assignment as in part (a). Use Markov's inequality together with your answer to part (c) to deduce an upper bound on the probability that the assignment satisfies less than $\frac{1}{2}$ of the clauses. [Hint: Consider the random variable which counts the number of *unsatisfied* clauses.] **+5pts**

5. [14 pts] Fake coins

Suppose you are given a bag containing n unbiased coins. You are told that $n - 1$ of these are normal coins, with heads on one side and tails on the other; however, the remaining coin has heads on both its sides.

(a) Suppose you reach into the bag, pick out a coin uniformly at random, flip it and get a head. What is the (conditional) probability that this coin you chose is the fake (i.e., double-headed) coin? **6pts**

(b) Suppose you flip the coin k times after picking it (instead of just once) and see k heads. What is now the conditional probability that you picked the fake coin? **4pts**

(c) Suppose you wanted to decide whether the chosen coin was fake by flipping it k times; the decision procedure returns FAKE if all k flips come up heads, otherwise it returns NORMAL. What is the (unconditional) probability that this procedure makes an error? **4pts**

[continued overleaf]

6. [12 pts] Arithmetic/polynomials true or false

For each of the following statements, say whether the statement is true or false. You need not provide any justification for your answer; however, you may be awarded partial credit for an incorrect answer if you attempt a justification.

(a) If p is a prime, then for any other prime $q > p$, there do not exist integers $a, b \neq 1$ such that $ab = p \bmod q$. **2pts**

(b) If n is not a prime and $a > 0$, then it is always the case that $a^n \neq a \bmod n$. **2pts**

(c) If we take all powers of 3 mod 55, then we get a permutation of the numbers $1, 2, \ldots, 54$. **2pts**

(d) In the field modulo 29, there is exactly one polynomial of degree 5 that passes through 6 given points. **2pts**

(e) In the field modulo 29, there are exactly 29 polynomials of degree 5 that pass through 5 given points. **2pts**

(f) Using the Berlekamp/Welsh coding, we can recover lost digits as long as not more than one quarter of the digits are lost. **2pts**

7. [16 + 5 pts] A new encryption algorithm

Consider the following encryption algorithm. For Alice to transmit a secret message m to Bob, they first agree (publically) on a large prime p ($p \gg m$). Then, Alice privately picks a random number a between 2 and $p-1$, and relatively prime to $p-1$, and Bob likewise privately picks a random number b between 2 and $p-1$, and relatively prime to $p-1$. The protocol (with some missing details that you will figure out below) is then as follows:

(i) Alice sends $c_1 = m^a \bmod p$ to Bob.

(ii) Bob sends $c_2 = c_1^b \bmod p$ to Alice.

(iii) Alice sends $c_3 = c_2^d \bmod p = m^b \bmod p$ to Bob.

(iv) Bob recovers the message m using only the information (p, b, c_1, c_2, c_3) available to him.

Answer the following questions:

(a) What should Alice choose for d in step (iii) of the algorithm, in order to make $c_2^d = m^b \bmod p$? Show how Alice can compute d in polynomial time and using only the information available to her at that time. **8pts**

(b) How does Bob recover the message m in step (iv)? **8pts**

(c) **[Extra credit]** Recall that the RSA scheme fails if there is an eavesdropper who can factor efficiently. By contrast, this algorithm fails if there is an eavesdropper who can efficiently solve the "discrete log" problem. More precisely, suppose there is an algorithm that, given as input x, y and a prime p, returns in polynomial time a number z such that $x^z = y \bmod p$ (whenever such an integer z exists). [Note: No such algorithm is known to exist.] Show how an eavesdropper who knows only p, c_1, c_2, and c_3 could use such an algorithm to recover m in polynomial time. **+5pts**

[The End!]