Discrete Math in CS	Problem Set 5
CS 280 Fall 2005	**Due October 21, 2005**

Reading: Rosen, Sections 5.1-5.2.

---

(1) In class, we've used the problem of load-balancing jobs on processors as an on-going application of probabilistic analysis. Suppose we have n identical computing servers, and n jobs that need to be performed. However, because we need to route the jobs quickly, with limited knowledge of the global state of the system, we simply assign each job independently and uniformly at random to one of the servers. That is, the jobs arrive in sequence, and each is assigned uniformly at random to one of the servers, independent of the assignment of previous jobs.

Determine the probability that each server is assigned exactly one job, by giving a formula for this quantity in terms of n. In particular, describe the underlying sample space, the event in question, and its probability. (You do not need to write out the entire sample space or the event; it is enough to give a precise mathematical formulation of it.)

Also, show that this probability converges to 0 as n increases.

---

(2) A biomedical lab runs a cluster of machines that measure and analyze data, and due to the sensitivity of the data, they need to back it up every night at a secure off-site location.

They've set up the following automated system for doing this, involving four computers: a Data Aggregator, a Relay Server, a Firewall Server, and an Offsite Backup. By 11:00 PM every night, the results from all the day's experiments reside on the Data Aggregator as a single large data file. At 11:00 PM the Data Aggregator opens a connection to the Relay Server to send it this data. At 11:15 PM, the Relay Server opens a connection to the Firewall Server, and sends the data to it. At 11:30 PM, the Data Aggregator connects to the Firewall Server and sends a duplicate copy of the data. (This is designed to provide some extra redundancy in the backup, just in case.) Finally, at 11:45 PM, the Firewall Server connects to the Offsite Backup and sends the data to it (either the original or backup copy, if it has both – it doesn't matter which). If successful, each of these individual transmissions lasts at most 15 minutes (and so the final transmission is done by midnight).

Now, there's some probability that each of these connections fails, in which case the data doesn't get transmitted across that particular connection. Suppose that the connection from the Data Aggregator to the Relay Server fails with probability .1, the connection from the Relay Server to the Firewall Server fails with probability .1, the connection from the Data Aggregator to the Firewall Server fails with probability .05, and the connection from the Firewall Server to the Offsite Backup fails with probability .02. Assume that the successes of each of these network connections are mutually independent as events.

What is the probability that a copy of the data file containing the day's results successfully ends up at the Offsite Backup at midnight? Provide an explanation for your answer.

(3) Some friends of yours are producing a specialized piece of computer hardware that implements the Boolean function

$$(a \wedge b) \vee (c \wedge d),$$

to be used a component in a much larger device. (This function is depicted in Figure 1.) The hardware takes four inputs, corresponding to a, b, c, and d, and it produces the corresponding Boolean output.

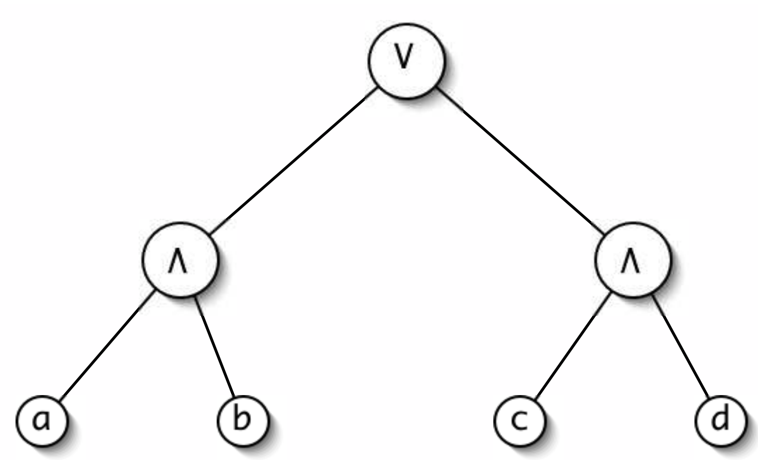

Figure 1: The Boolean formula in Question 3.

To test different versions of the hardware (and in particular to make sure the hardware keeps functioning over long periods of heavy use), they're feeding in random values of a, b, c, and d, and then checking that the output is correct.

In the process of doing this, they've noticed something that strikes them as strange. For each of the four variables, they're setting it to T with probability $\frac{1}{2}$ and to F with probability $\frac{1}{2}$; the settings of all variables are mutually independent. However, when they observe the output of the hardware over the course of many runs, it produces the output F noticeably more often than it produces the output T. "Each individual run of the hardware looks fine," they tell you, "and we're setting each variable to T or F with equal probability, so why doesn't it produce T and F as output essentially equal fractions of the time?"

(a) Work out the probability the hardware produces the output T, and explain how this is consistent with the phenomenon your friends are observing.

(b) Suppose, more generally, that for some value of x, one were to set each variable independently to T with probability x and to F with probability $1-x$. Give a value of x for which the output is T with probability exactly $\frac{1}{2}$. Provide an explanation for your answer (including an explanation for how the output probability depends on x, and why).

(4) You're monitoring radio signals on 500 different channels; each is transmitting a signal consisting of a sequence of 0's and 1's. (We'll refer to each 0 or 1 in one of these sequences as a *bit*.) On 495 of the channels, this signal is produced by a random process that produces each bit in the sequence independently, generating a 0 with probability $\frac{1}{2}$ and generating a 1 with probability $\frac{1}{2}$. On the remaining 5 of the channels, the signal is produced simply by sending the value 1 over and over, producing the sequence $1111111\cdots$.

If you tune into one of these channels at random, and you start receiving many 1's in a row, you'll begin to suspect that you've tuned into one of the rare "all-ones" channels. And the more 1's in a row you receive, the more strongly you'll suspect this. Let's consider how we might make this quantitative.

Suppose you tune to one of these channels selected uniformly at random and you listen to it until you've received k bits; and suppose that all k of these bits are equal to 1. What is the smallest value of k such that, conditioned on this signal, there's a 90% probability that you were listening to one of the "all-ones" channels, rather than one of the "random" channels? Provide an explanation for your answer.