Foundations of Information Systems

Final Exam

February 20, 2024

Last Name: ______________________________

First Name: ______________________________

Signature: ______________________________

1. Are the following identities true or false? If true, give a proof. If false, give a counterexample.

   (a) $(a \wedge b)' \wedge c = (a \vee b) \vee c'$

   (b) $((a \vee b) \wedge c)' = (a \wedge c)' \wedge (b \wedge c)'$

(5+5)

(a) Take $c = 0$ $\Rightarrow$ $(a \wedge b)' \wedge c = 0$ but $(a \vee b) \vee c' = 1$

$\Rightarrow$ The identity is false.

(b) $((a \vee b) \wedge c)' = (a \vee b)' \vee c'$ (De Morgan)

$= (a' \wedge b') \vee c'$ (De Morgan)

$= (a' \vee c') \wedge (b' \vee c')$ (distributivity)

$= (a \wedge c)' \wedge (b \wedge c)'$ (De Morgan)

This proves the identity as stated.

2. (a) Convert the decimal number 433 to hexadecimal.
   (b) Add the 4-bit two's complement binary numbers 1110 and 1011.
   (c) Confirm your result by converting all three numbers from (b) to decimal.
   (d) How do you detect overflow when adding two's complement binary numbers?

(5+5+5+5)

(a)

$$\begin{array}{r} 433 : 16 = 27 \\ -32 \\ \hline 113 \\ -112 \\ \hline 1 \end{array} \qquad \begin{array}{r} 27 : 16 = 1 \\ -16 \\ \hline 11 \end{array}$$

$\Rightarrow 433 = 1 \cdot 16^2 + 11 \cdot 16^1 + 1 = (1B1)_{16}$

(b)

$$\begin{array}{r} 1110 \\ 1011 \\ \hline 1001 \end{array}$$

In two's complement addition, the final carry is ignored, so result is 1001.

(c) Two's complement of 1110: $0001 + 1 = 0010 = (2)_{10}$

1011: $0100 + 1 = 0101 = (5)_{10}$

1001: $0110 + 1 = 0111 = (7)_{10}$

So the computation in (b) represents $(-2) + (-5) = -7$

(d) Addition of differently-signed numbers can never overflow.

Addition of same-signed numbers has overflow if and only if sign-bit of the result is different from that of the input numbers.

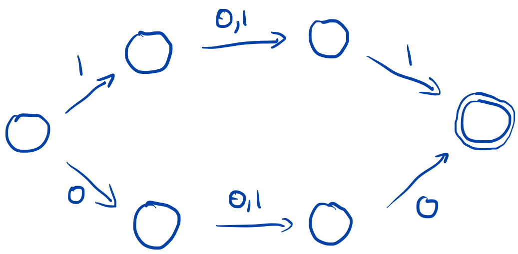
3. (a) Draw a finite state machine that can recognize whether a 3-bit binary string is a palindrome, i.e., reads the same backwards as forwards.

(b) State a regular expression that is equivalent to the machine from part (a).

(c) Is it possible to design a finite state machine that recognizes palindromes of arbitrary length? Explain!

(5+5+5)

(a)



(b) (0[01]0)|(1[01]1)

(c) Checking for the reverse-repeat of an arbitrary-length string requires an arbitrary number of states, which cannot be done with an FSA.

4. The following bit strings of a Hamming-(8,4) encoded message are received. Correct single-bit errors or detect double-bit errors as appropriate.

   (a) 11001011

   (b) 00010100

(5+5)

(a)

$p_0\ p_1\ p_2\ d_0\ p_4\ d_1\ d_2\ d_3$

1 1 0 0 1 0 1 1

$p_0 = 1$ ⇒ assume 1-bit error

$p_1 = 0$

$p_2 = 0$

$p_4 = 1$

} error in position $(100)_2 = 4$

⇒ p_4 is wrong

⇒ message is 0011

(b)

$p_0\ p_1\ p_2\ d_0\ p_4\ d_1\ d_2\ d_3$

0 0 0 1 0 1 0 0

$p_0 = 0$ ⇒ even number of errors

$p_1 = 0$

$p_2 = 1$ another error, so message is unrecoverable.

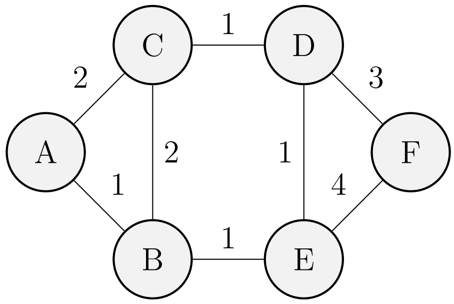
5. Suppose you have a file `data_A`. You create a hard link `data_B` and a soft link `data_C` to it. Now you delete `data_A`. Is the data lost? Can you still access it via `data_B`? Via `data_C`? (5)

The data can still be accessed via data_B.

Reason: Both hardlinks are direct references to the inode, which keeps a reference count. Inode data will only be deleted if all references to it are gone.

The soft link data_C, however, still points to data_A, so is now stale.

6. Consider the following router network which uses distance vector routing.



(a) State the optimal distance vector and routing table for router C. You do not need to compute anything as the network is simple enough to spot the answer directly.

(b) Now suppose that router C is malicious and wants to cut router F off the network. Can it do this? If so, what distance vector does it need to broadcast to its neighbors to attract all traffic destined for F?

(5+5)

(a)

From C to	A	B	C	D	E	F
distance	2	2	0	1	2	4
first hop	A	B	C	D	D	D

(b) If C advertises a small enough distance to F (e.g. d(C,F) = 1), F's direct neighbors will start routing traffic for F via C (for D) or via D-C (for E). Thus, C can drop all packets destined for F.

7. Suppose you have three identical disks. The most natural way to create redundant storage from these three disks is a RAID-5 array. Your remember from class that RAID-5 with large capacity drives has a high probability of hitting an unrecoverable error during rebuild after a single-drive failure, so you ponder if it's worth investing in a fourth disk and using RAID-10 ("stripe of mirrors") instead. To help making this decision, rate the performance of the four-disk RAID-10 *relative* to the three-disk RAID-5, with a brief explanation, in each of the following categories:

   (a) Usable capacity (*not* raw capacity!)

   (b) Read speed when sequentially reading a large file

   (c) Write speed when sequentially writing a large file

   (d) Write speed when writing random blocks

   (e) Probability of hitting an error during rebuild after a single-drive failure

   Extra credit: Based on these numbers, write a recommendation whether or not to purchase the extra drive and use it in RAID-10 configuration. (2+2+2+2+2+*5*)

(a) RAID-5 uses the capacity of one disk for parity } net capacity is same (two disks)
RAID-10 uses two disks as mirrors

(b) All disks can be used for reading ⇒ RAID-10 is $\frac{4}{3}$ faster than RAID 10

(c) Assuming that parity/redundant writing is free, sequential writes are limited by the effective number of disks carrying first copy of data (2 in both cases) ⇒ sequential write speed is the same.

(d) In RAID-10, worst case is that all blocks are on one mirrored pair, so we'd get single disk performance. If blocks are truely random and can be pre-sorted, performance can be up to twice as good.

In RAID-5, need to read old data + parity, then write both back.

Since this occupies two of the 3 disks, no interleaving of operations possible

⇒ RAID-5 can be up to 4 times slower than RAID-10 for random writes.

(e) RAID-5 rebuild requires reading all data from the two surviving disks.
RAID-10 rebuild requires making a copy of the mirror of the failed disk.
⇒ RAID-10 rebuild is twice as safe.

(Solution ctd./scratch paper)

Recommendation:

- If safety is a concern, RAID-10 is better, but a factor 2 is not a game changer here. Better to buy the extra disk, but operate in RAID-6.
- On the other hand, the speed advantage of RAID-10 may be significant, especially for workloads that perform frequent random writes. But should be used only if safety guarantees suffice.