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

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

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

(5+5)

(a) $a \vee (a \wedge b) = (a \wedge 1) \vee (a \wedge b)$ (identity)

$= a \wedge (1 \vee b)$ (distributivity)

$= a$ (Theorem 2 from class, identity)

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

$= 1 \wedge (a \vee b)$ (complement)

$= a \vee b$ (identity)

So both expressions are true.

2. (a) What is the largest number you can represent in 6-bit two's complement binary representation? What is the smallest such number?

(b) Convert the decimal number 9.5625 to binary.

(c) Compute

$$(1.2 \cdot 10^{-2} - 7 \cdot 10^{2}) + 6.99 \cdot 10^{2}$$

in *decimal* floating point arithmetic with 4 significant digits. Can you reduce the relative error by reordering the operations?

(5+5+5)

(a) first bit is the sign bit, so we have 2^5 negative numbers $(-1, \ldots, -32)$ and 2^5 non-negative numbers $(0, \ldots, 31)$. So 31 is the largest, −32 the smallest number.

(b) $(9)_{10} = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = (1001)_2$

$0.5625 = \frac{1}{2} \cdot 1.125 = \frac{1}{2}(1 + 0.125) = \frac{1}{2} + \frac{1}{16} = 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} = (0.1001)_2$

$\Rightarrow (9.5625)_{10} = (1001.1001)_2$

(c)

$$\begin{array}{r} 0.012 \\ -700.0 \\ \hline -700.0 \end{array} \qquad \begin{array}{r} 699.0 \\ -700.0 \\ \hline -1.0 \end{array}$$

↑ truncate

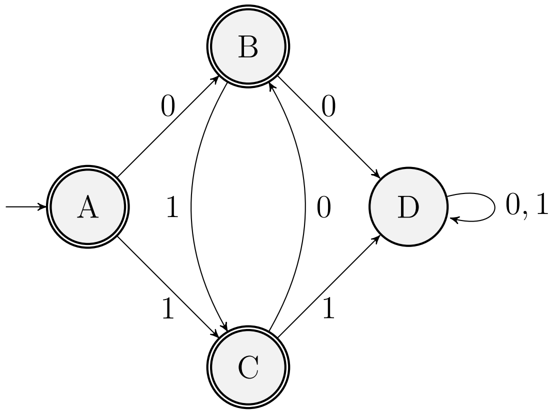
So the result is $-1 \cdot 10^0$, with about 1% relative error.

If we re-order:

$$\begin{array}{r} 699.0 \\ -700.0 \\ \hline -1.0 \end{array} \qquad \begin{array}{r} -1.000 \\ 0.012 \\ \hline -0.988 \end{array}$$

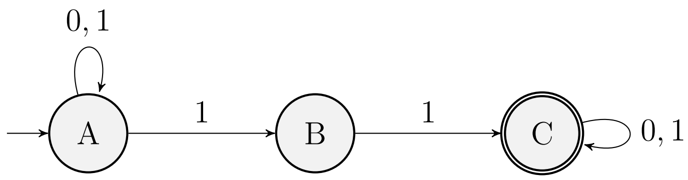
So the result is $-9.880 \cdot 10^{-1}$, which is correct.

3. (a) Which strings does the following finite state machine accept?



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

(c) Convert the following non-deterministic finite state machine into a deterministic one.

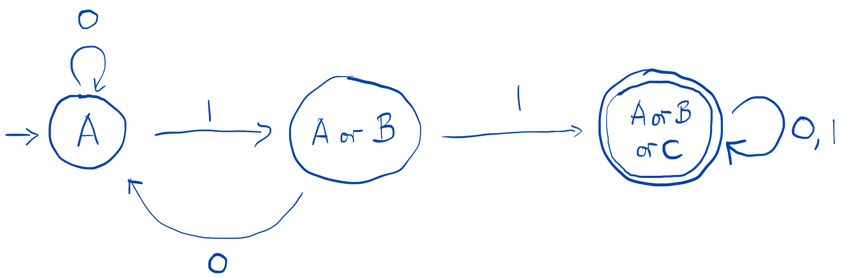


(5+5+5)

(a) Any alternating sequence of 0 and 1.

(b) $((10)^{*}1?) \mid (01)^{*}0?)$

(c)



(Note: The label "A or B or C" is not completely accurate because a 0-transition from there must exclude B, but this does not impact the function of the FSA.)

4. (a) Encode the string 0110 as a Hamming-(8,4) encoded message. Your answer should have a clear indication of the bit ordering of the code word.

(b) Can you design a code that can detect and correct a single-bit error for 16 bits of data in a message of 20 bits total length? Explain!

(5+5)

(a)

p_0	p_1	p_2	d_0	p_4	d_1	d_2	d_3	
			0		1	1	0	
	—		—		—		—	$p_1 = 1$
		—	—			—	—	$p_2 = 1$
				—	—	—	—	$p_4 = 0$

} $p_0 = 0$ (overall parity)

So the code word is 01100110

(b) No, there can only be 4 parity bits, which can encode 16 different error situations. But the message is 20 bits long, so there are 20 possible single-bit errors. Thus, some errors cannot be located with certainty whatever encoding is chosen.

5. (a) Explain the difference between a soft link and a hard link in a file system.

(b) What do you think if you find a USB drive with a Unix file system on it which contains a file named `Funny_Cat_Picture.jpg` which is a symbolic link to `/etc/passwd`?

(5+5)

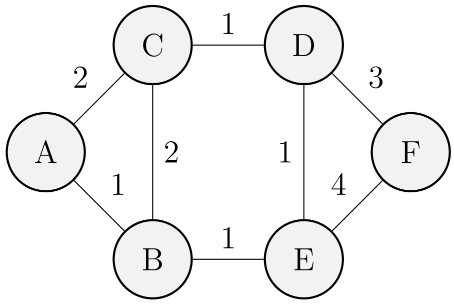
(a) Hard link: Additional reference to an inode (which contains file metadata and references the data), reference-counted. (Also: cannot cross filesystem boundaries, hard-links to directories are not allowed.)

Soft link: File data contains the character representation of the path to the linked file. No reference count.

(b) It's an attempt to make a soft link appear like a regular file with "interesting" content to entice a user to, e.g., attach it to an email. Instead, the target data (here /etc/passwd) would be attached and possibly leak sensitive information.

(Remark: on modern Unix systems, /etc/passwd does not contain passwords, not even in hashed form, but it does contain user names. So a leak would not directly compromise the system, but might still be useful for an attacker trying to infiltrate an organisation.)

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



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

(b) Now suppose that router A is malicious and wants to intercept traffic destined for router F. Which routers can A prevent from communicating with F by advertising a false link? Assume that link costs must be positive integers.

(5+5)

(a)

From A to	B	C	D	E	F
next hop	B	C	B or C	B	B or C
cost	1	2	3	2	6

(b) The worst A can do is to advertise a link at cost 1 to F. Then C and B would route via A, and E would route via B-A. Only D would still find it cheaper to directly route to F.

7. You are given the following relational database schema of an online shop:

```
ITEM(ITEM_ID, DESCRIPTION)
CUSTOMER(CUSTOMER_ID, NAME, ADDRESS)
ORDER(CUSTOMER_ID, ITEM_ID)
```

(a) Write a query, either using relational algebra or SQL, to find the names of all customers who bought a "Superbike" (`DESCRIPTION='Superbike'`).

(b) In a real online shop, an order can contain any number of items. ~~Extend~~ Modify the given schema to represent such orders.

(5+5)

(a) $\pi_{NAME}\ \sigma_{DESCRIPTION = 'Superbike'}\ (ITEM \bowtie ORDER \bowtie CUSTOMER)$

Alternatively, in SQL:

```
SELECT NAME FROM ITEM, ORDER, CUSTOMER WHERE ITEM.ITEM_ID = ORDER.ITEM_ID
    AND CUSTOMER.CUSTOMER_ID = ORDER.CUSTOMER_ID
    AND DESCRIPTION = 'Superbike'
```

(b) Keep ITEM and CUSTOMER as above. Then

ORDER_ITEM(ORDER_ID, ITEM_ID, QUANTITY) — QUANTITY optional

ORDER(ORDER_ID, CUSTOMER_ID, DATE, ...) — DATE, ... optional