Quantum Infortmation Processing, Problem Set 1

Alex Hahn February 24, 2014

Problem 1: Tensor products and positional notation.

Section 1.2 of the course text shows that if an integer x between 0 and $2^n - 1$ is represented by a 2^n component column vector with all components 0, except for a 1 in the position x places down from the top place (the top places down from itself), then if the number is represented in binary, that column vector is just the tensor product of the n 2-component column vectors that represent the values of its n bits. Give an argument that is true of decimal numbers as well, taking as your example, the number 527, showing that its representation as a 1000-component column vector (all 0's except for a 1 at position 527 places down from the top) is just the tensor product of the three 10-component column vectors representing (from right to left) the digits 7, 2, and 5. (Since it is hard to fit a 1000 entry column vetor on the page, this requires an explanation that is at least in part, verbal.)

In base 10 we simply let each column represent a power of 10 (10^0 being farthest to the right) with the coeffecient being the entry that is a 1 in the corresponding column vector

In base 10 we can represent the number 527 as the tensor product:

$$527 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

This tensor product would give us the 1000 length column vector with a 1 527 places down from the top (note that the first position is the zeroth position)

Problem 2: Manipulating elementary operators

a) Carry out the "more algebraic" derivation mentionaed at the top of page 13 of the relation

$$\mathbf{S}_{ij} = \mathbf{C}_{ij}\mathbf{C}_{ji}\mathbf{C}_{ij}$$

between the SWAP operator in the form (see section 1.4 of text for notation)

$$\mathbf{S}_{ij} = \mathbf{n}_i\mathbf{n}_j + (\mathbf{X}_i\mathbf{X}_j)(\mathbf{n}_i\overline{\mathbf{n}}_j + \overline{\mathbf{n}}_i\mathbf{n}_j),$$

and the cNot operator in the form

$$\mathbf{C}_{ij} = \overline{\mathbf{n}}_i + \mathbf{X}_j\mathbf{n}_i.$$

You should only use the identities $\mathbf{n}^2 = \mathbf{n}$, $\overline{\mathbf{n}}^2 = \overline{\mathbf{n}}$, $\mathbf{n}\overline{\mathbf{n}} = \overline{\mathbf{n}}\mathbf{n} = 0$, $\mathbf{n} + \overline{\mathbf{n}} = 1$, $\mathbf{n}\mathbf{X} = \mathbf{X}\overline{\mathbf{n}}$, $\overline{\mathbf{n}}\mathbf{X} = \mathbf{X}\mathbf{n}$, and $\mathbf{X}^2 = 1$, obeyed by the 1-bit NOT and number operators, and the fact that 1-bit operators acting on different Cbits commute.

$$\mathbf{S}_{ij} = \mathbf{C}_{ij}\mathbf{C}_{ji}\mathbf{C}_{ij}$$

$$= (\overline{\mathbf{n}}_i + \mathbf{X}_j\mathbf{n}_i)(\overline{\mathbf{n}}_j + \mathbf{X}_i\mathbf{n}_j)(\overline{\mathbf{n}}_i + \mathbf{X}_j\mathbf{n}_i)$$

$$= \overline{\mathbf{n}}_i\overline{\mathbf{n}}_j + \mathbf{X}_j\mathbf{n}_i\mathbf{X}_i\mathbf{n}_j\overline{\mathbf{n}}_i + \overline{\mathbf{n}}_i\mathbf{X}_i\mathbf{n}_j\mathbf{X}_j\mathbf{n}_i + \mathbf{X}_j\mathbf{n}_i\overline{\mathbf{n}}_j\mathbf{X}_j\mathbf{n}_i$$

using the identities above and changing the order (terms commute) of a few terms:

$$= \overline{\mathbf{n}}_i\overline{\mathbf{n}}_j + \mathbf{X}_i\mathbf{X}_j\overline{\mathbf{n}}_i\mathbf{n}_j + \mathbf{X}_i\mathbf{X}_j\mathbf{n}_i\overline{\mathbf{n}}_j + \mathbf{n}_i\mathbf{n}_j$$

Finally grouping terms we have:

$$\mathbf{n}_i\mathbf{n}_j + \overline{\mathbf{n}}_i\overline{\mathbf{n}}_j + (\mathbf{X}_i\mathbf{X}_j)(\overline{\mathbf{n}}_i\mathbf{n}_j + \mathbf{n}_i\overline{\mathbf{n}}_j)\ \checkmark$$

(From here on in assume everything but the subscripts are bold)

b) Show from the equation (1.35) above that $S_{ij}^2 = 1$ again only use the identities given in the first part.

$$(\mathbf{S}_{ij})^2 = (\mathbf{n}_i\mathbf{n}_j + (\mathbf{X}_i\mathbf{X}_j)(\mathbf{n}_i\overline{\mathbf{n}}_j + \overline{\mathbf{n}}_i\mathbf{n}_j))^2$$

$$= (n_in_jn_in_j + \bar{n}_i\bar{n}_jn_in_j + (X_iX_j)(\bar{n}_in_j + n_i\bar{n}_j)n_in_j) + (n_in_j\bar{n}_i\bar{n}_j + \bar{n}_i\bar{n}_j\bar{n}_i\bar{n}_j + (X_iX_j)(\bar{n}_in_j + n_i\bar{n}_j)\bar{n}_i\bar{n}_j)$$

$$+(n_in_j(X_iX_j)(\bar{n}_in_j + n_i\bar{n}_j) + \bar{n}_i\bar{n}_j(X_iX_j)(\bar{n}_in_j + n_i\bar{n}_j) + (X_iX_j)(\bar{n}_in_j + n_i\bar{n}_j)(X_iX_j)(\bar{n}_in_j + n_i\bar{n}_j))$$

$$= n_in_j) + (\bar{n}_i\bar{n}_j\bar{n}_i\bar{n}_j) + ((\bar{n}_in_j + n_i\bar{n}_j)(\bar{n}_in_j + n_i\bar{n}_j))$$

$$= n_i(n_j + \bar{n}_j) + \bar{n}_i(\bar{n}_j + n_j)$$

$$n_i + \bar{n}_i$$

$$= 1\ \checkmark$$

c) Verify $S_{ij} = C_{ij}C_{ji}C_{ij}$, by using repeated application of (1.21) and $x \oplus x = 0$.

To show for the Cbit form:

$$S_{10}\left|x\right\rangle\left|y\right\rangle = C_{10}C_{01}C_{10}\left|x\right\rangle\left|y\right\rangle$$

$$= C_{10}C_{01}\left|x\right\rangle\left|y \oplus x\right\rangle$$

$$= C_{10}\left|x \oplus y \oplus x\right\rangle\left|y \oplus x\right\rangle$$

$$= C_{10}\left|y\right\rangle\left|y \oplus x\right\rangle$$

$$= \left|y\right\rangle\left|y \oplus x \oplus y\right\rangle$$

$$= \left|y\right\rangle\left|x\right\rangle$$

I'll show this in a more generalized form (applicable to qbits as well!)

Given state $\alpha\beta\left|00\right\rangle + \alpha\delta\left|01\right\rangle + \beta\kappa\left|10\right\rangle + \beta\delta\left|11\right\rangle$

The first Cnot operator gives us:

$$\alpha\kappa\left|00\right\rangle + \alpha\delta\left|01\right\rangle + \beta\delta\left|10\right\rangle + \beta\kappa\left|11\right\rangle$$

the next Cnot operator gives us:

$$\alpha\kappa\left|00\right\rangle + \beta\kappa\left|01\right\rangle + \beta\delta\left|10\right\rangle + \alpha\delta\left|11\right\rangle$$

and finally the third Cnot returns

$$\alpha\kappa\left|00\right\rangle + \beta\kappa\left|01\right\rangle + \alpha\delta\left|10\right\rangle + \beta\delta\left|11\right\rangle$$

the coefficients are commutable so

$$= \kappa\alpha\,|00\rangle + \kappa\beta\,|01\rangle\,\delta\alpha\,|01\rangle\,\delta\beta\,|11\rangle$$

which is our original state but swapped.

d) Use $n = \frac{1}{2}(1 - Z), \bar{n} = \frac{1}{2}(1 + Z)$ in the expression (1.35) above for S_{ij} to show (1.49): $S_{ij} = \frac{1}{2}(! + Z_iZ_j) + \frac{1}{2}(X_iX_j)(1 - Z_iZ_j)$.

$$S_{ij} = n_in_j + \bar{n}_in_j + (X_iX_j)(\bar{n}_in_j + n_i\bar{n}_j)$$

$$= \frac{1}{2}(1 - Z_i)\frac{1}{2}(1 - Z_j) + \frac{1}{2}(1 + Z_i)\frac{1}{2}(1 + Z_j) + (X_iX_j)(\frac{1}{2}(1 + Z_i)\frac{1}{2}(1 - Z_j) + \frac{1}{2}(1 - Z_j) + \frac{1}{2}(1 + Z_j))$$

$$= \frac{1}{4}(1 - Z_i - Z_j + Z_iZ + j) + \frac{1}{4}(1 + Z_i + Z_j + Z_iZ_j) + (X_iX_j)(\frac{1}{4}(1 + Z_i - Z_j - Z_iZ_j) + \frac{1}{4}(1 - Z_i + Z_j - Z_iZ_j)$$

$$= \frac{1}{2}(1 + Z_iZ_j) + \frac{1}{2}(X_iX_j)(1 - Z_iZ_j)\ \checkmark$$

Problem 3: 2-Cbit gates As mentioned in class, there are 4!=24 possible classical operations on the four 2-Cbit basis states. How many of these are achievable via the classical operations S_{ij}, C_{ij}, and X_j (and compositions thereof)?

All 24 are achievable via these operations because they cover all permuations of the 2-Cbit basis (we also know that they can be mapped into each other because these operations are all reversible as stated many times in the chapter, we can make many of them singly through swap (exchanging the columns of the basis) (or 3 cnots) the rest (if odd parity) can be constructed through not, Cnots, or some combination of the three.

Problem 4: Some 3-Cbit gates a) The classical 3-Cbit gates $\mathbf{C}_{ij}$ operate on an 8 dimensional vector space of 3-Cbit states. Find the 8x8 matrices for $\mathbf{C}_{21}$ and $\mathbf{C}_0 2$ by generalizing to 3-Cbit states the procedure used to construct the 44 matrices of cNOT on 2-Cbit states on p.10 of chpt.1.

$$\mathbf{C}_{21} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\mathbf{C}_{02} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

b) The 3-Cbit controlled-controlled-NOT gate $\mathbf{T}_{ijk}$ (also called the Toffoli gate, after its inventor) plays a very important role in the classical theory of reversible computation. It has two control bits (i and j) and a target bit (k), and is defined to act as the identity unless both control bits are 1, in which case it acts

as NOT on the target bit. Find the 8x 8 matrices in the set of 3-Cbit states $|x_2\rangle\,|x_1\rangle\,|x_0\rangle$ of $\mathbf{T}_{210}$, $\mathbf{T}_{201}$, $\mathbf{T}_{102}$.

$$\mathbf{T}_{210} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\mathbf{T}_{201} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\mathbf{T}_{102} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Problem 5: Consider the following cooperative game played by Alica and Bob: Alice receives a bit x and Bob receives a bit y, with both bits uniformly random and independent. The players win if Alice outputs a bit a and Bob outputs a bit b, such that $a + b = xy(\text{mod}2)$ (equivalently, $a \oplus b = xy$, where bits take the values 0,1). They can agree on a strategy in advance of receiving x and y, but no subsequent communication between them is allowed.

a) Give a deterministic strategy by which Alice and Bob can win this game with probability 3/4.

The only strategy that repeatedly guarantees this probabiliy is if they both agree to play (and actually do play) 0 each time (look at the results table, it's clear that the only time this wouldn't win is if both x and y were 1).

b) Show that no deterministic strategy lets them win with more than probability 3/4. (Note that Alice has four possible deterministic strategies [**0, 1, x,** $\tilde{\mathbf{x}}$], and bob has four [**0, 1, y,** $\tilde{\mathbf{y}}$], so there's a total of 16 possible joint deterministic strategies.)

We already know that both choosing 0 no matter what gives 3/4 probability. I will show the remaining then argue by symmetry that we've covered all possibilities (switching the roles of Alice and Bob makes no real difference) and that none are greater than 3/4

Alice: 0 and Bob: 1 gives 1/4 (xy is only 1, 1/4 of the time
Alice: 0 and Bob: y gives 3/4 (only when x=0 and y=1 does it fail
Alice: 0 and Bob: $\tilde{y}$ gives 1/4 (only when x=0 and y=1 does it work

Alice: 1 and Bob: 1 gives 3/4 (only x=1 and y=1 doesn't work
Alice: 1 and Bob: y gives 1/4 (only x=1 and y=0 works
Alice: 1 and Bob: $\tilde{y}$ gives 3/4 (only x=0 and y=1 fails)
Alice: x and Bob: y gives 1/4 (only both x and y 0 works)
Alice: x and Bob: $\tilde{y}$ gives 3/4 (only both x and y 0 fails)
Alice $\tilde{x}$ and Bob: $\tilde{y}$ gives 1/4 (only both 0 works)

from here we can argue that if we simply switch the names of Bob and Alice we cover the rest of the cases by symmetry (it shouldn't matter who's "Bob" and who's "Alice"). None are over 75% success rate.

c) Show that no probabalistic strategy lets them win with more than probability 3/4. (In a probabalistic strategy, Alice plays her possible strategies with some fixed probabilities $p_0, p_1 p_x p_{\bar{x}}$, where $\sum_\alpha p_\alpha = 1$, same for bob.
We need to show that any linear combination of strategies doesn't yield an expected value greater than 3/4.

I believe that the above brute force method shows that you still can't get above a 3/4's chance of winning. If we assign arbitrary coefficients $0 \leq 0 \leq 1$ to all of her/their choices such that the sum of the probabilities of all choices is 1 every set of chosen strategies (one must be picked) still yields a 3/4 chance or lower as shown above. This is equivalent to saying that instead of talking about their strategies before hand they don't but they still have to decide on one to pick before they play and we've shown that every combination yields a $\leq 3/4$ chance of succeeding for random x and y (as stated).

(Effectively they have to pick some equivalent strategy to one enumerated above even if it is probalisticly).

*(Talked with Andrew Casey about a few of the problems)