Quantum Information - Homework 04

Roei Rosenzweig 313590937 $\lozenge$ 205798440 Roey Maor

06/08/2017

---

1 Purity of qubit

1. Let $p = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ be a density matrix. The 3 condition that must hold are:

   (a) p is hermitic - meaning $p = p^{\star}$, and in terms of individual elements:
   $$c = b^{\star}$$

   (b) $tr\,(p) = 1$ (normalization):
   $$a + d = 1$$

   (c) eigenvalues are non-negative. from the previous 2 conditions we can represent $p = \begin{pmatrix} a & b \\ b^{\star} & 1-a \end{pmatrix}$. Let us denote the 2 eigenvalues of p as λ_1, λ_2 (which may be the say eigenvalue). We know that:
   $$tr\,(p) = \lambda_1 + \lambda_2 = 1$$
   $$det\,(p) = \lambda_1 \lambda_2 = a - a^2 - b^2$$
   from $\lambda_1 + \lambda_2 = 1$ we know already that **at least** one eigen value is positive. Without lose of generality, lets say $\lambda_1 > 0$. Thus, $\lambda_2 > 0 \iff \frac{a-a^2-b^2}{\lambda_1}$, and because $\lambda_1 > 0$ we get the inequality
   $$a - a^2 - b^2 > 0$$

   We saw that any density matrix p' is representable as $p' = \frac{I + \vec{r} \cdot \vec{\sigma}}{2}$ where $\vec{\sigma} = (\sigma_x, \sigma_y, \sigma_z)$. Reminder: $\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ $\sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$ $\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$. To find the $\vec{r}$ corresponding to our earlier-mentioned p, we calculate:
   $$\begin{aligned} p &= \frac{I + \vec{r} \cdot \vec{\hat{\sigma}}}{2} && \iff \\ \begin{pmatrix} a & b \\ c & d \end{pmatrix} &= \frac{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + r_x \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} + r_y \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} + r_z \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}}{2} && \iff \\ \begin{pmatrix} a & b \\ b^{*} & 1-a \end{pmatrix} &= \frac{1}{2} \begin{pmatrix} 1 + r_z & r_x - i r_y \\ r_x + i r_y & 1 - r_z \end{pmatrix} \end{aligned}$$

   From which we get the equations:
   $$r_z = 2a - 1$$
   $$r_x = 2b + i r_y$$

   We can explicitly represent r_x and r_y with a and b. But these equations are sufficient to show that $|r| \leq 1$:
   $$\begin{aligned} |\vec{r}|^2 &= r_x^{\star} r_x + r_y^{\star} r_y + r_z^{\star} r_z = 4b^2 - |r_y|^2 + |r_y|^2 + 4a^2 - 4a + 1 \\ &= 4b^2 + 4a^2 - 4a + 1 = -4(a - a^2 - b^2) + 1 \leq 1 \end{aligned}$$

   The justification for the last step is that $a - a^2 - b^2 > 0$ (as we saw earlier).

2. The **purity** of quantum state p is defined as $\gamma \triangleq tr\left(p^2\right)$. We can represent γ as a function of $|\vec{r}|$:

$$\begin{aligned}\gamma = tr\left(p^2\right) = tr\left(\left(\frac{I+\vec{r}\cdot\vec{\hat{\sigma}}}{2}\right)^2\right) &= \frac{1}{4}tr\left(I^2 + 2\vec{r}\cdot\vec{\hat{\sigma}} + \left(\vec{r}\cdot\vec{\hat{\sigma}}\right)^2\right) \\ &= \frac{1}{4}\left[tr\left(I^2\right) + 2tr\left(\vec{r}\cdot\vec{\hat{\sigma}}\right) + tr\left(\left(\vec{r}\cdot\vec{\hat{\sigma}}\right)^2\right)\right] \\ &= \frac{1}{4}\left[2 + 2tr\left(r_x\sigma_x + r_y\sigma_y + r_z\sigma_z\right) + |\vec{r}|^2 \cdot tr\left(I\right)\right] \\ &= \frac{1}{4}\left[2 + 2\left|\vec{r}\right|^2\right] = \frac{1+|\vec{r}|^2}{2}\end{aligned}$$

For a **pure** quantum state, $|\vec{r}| = 1$ - the state is sitting on the surface on Poincare sphere. From that, and from the presentation of γ that we calculated before, we arrive to the conclusion that for pure states $\gamma = 1$.

For the completely-mixed state $p' = \frac{I}{2}$, the representing-vector in Poincare sphere points exactly at the middle. In other words, $|\vec{r}| = 0$, and the purity is $\frac{1}{2}$.

3. For any unitary operator U and density matrix p, $\left(U \cdot p \cdot U^*\right)^2 = U \cdot p^2 \cdot U^*$. Then, the purity of the state $U \cdot p \cdot U^*$ is $\gamma_{(U \cdot p \cdot U^*)} = tr\left(U \cdot p^2 \cdot U^*\right) = tr\left(p^2\right) = \gamma_p$ (the 2nd equality is justified because trace is preserved under change of basis). We learn that the purity is preserved under unitary operators. because the purity is linear in $|\vec{r}|^2$, we also conclude that $|\vec{r}|$ is preserved under unitary transformation. Unitary operators can only **rotate** the Poincare vector.

2 Universality - Implementation of CZ

1. $CZ = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & -1\end{pmatrix}$

2. A good guess might be: $CZ = \left(I_2 \otimes H^{-1}\right) \cdot CNOT \cdot \left(I_2 \otimes H\right)$, where H is the hadamard transformation (change of basis $\{|0\rangle, |1\rangle\} \to \{|+\rangle, |-\rangle\}$. Lets validate that these operators transform the $|ij\rangle$ states correctly

$$\begin{pmatrix}1/\sqrt{2} & 1/\sqrt{2} & 0 & 0\\ 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0\\ 0 & 0 & 1/\sqrt{2} & 1/\sqrt{2}\\ 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2}\end{pmatrix}\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}\begin{pmatrix}1/\sqrt{2} & 1/\sqrt{2} & 0 & 0\\ 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0\\ 0 & 0 & 1/\sqrt{2} & 1/\sqrt{2}\\ 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2}\end{pmatrix} = \begin{pmatrix}1/\sqrt{2} & 1/\sqrt{2} & 0 & 0\\ 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0\\ 0 & 0 & 1/\sqrt{2} & 1/\sqrt{2}\\ 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2}\end{pmatrix}\begin{pmatrix}1/\sqrt{2} & 1/\sqrt{2} & 0 & 0\\ 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0\\ 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2}\\ 0 & 0 & 1/\sqrt{2} & 1/\sqrt{2}\end{pmatrix}$$

$$= \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & -1\end{pmatrix} = CZ$$

It does seems consistent with CZ.

3 Universality - Implementation of operator for 2 basic vectors

1. $V = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$. This is the hadamard transformation matrix of rank 2 H_2.

2. $c - V$ can be represented in a straight-forward way (similar to $CNOT$):

$$c - V = \begin{pmatrix}1 & 0 & \vec{0}\\ 0 & 1 & \vec{0}\\ \vec{0} & \vec{0} & V\end{pmatrix} = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}}\end{pmatrix}$$

We offer the following permutation: $A : |00\rangle \to |10\rangle \to |00\rangle$. It makes sense because it transform $|00\rangle, |11\rangle$ to $|10\rangle, |11\rangle$, which are not-trivially transformed by $c - V$.

$$A = \begin{pmatrix}0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\\ 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}$$

3. Lets try $U = A \cdot (c - V) \cdot A$. (transform $|00\rangle, |11\rangle$ to $|10\rangle, |11\rangle$, apply $c - V$, and permute everything back. notice that $A^{-1} = A$, the permutation is of rank 2).

$$A^3 \cdot (c-V) \cdot A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & 0 & -\frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 & -\frac{1}{\sqrt{2}} \end{pmatrix} = U$$

4. Lets first prove a helpful claim: $CNOT$ operates on the base $\{-, +\}$ as a controlled-not, but where the right bit is the control and the left is the values:

$$CNOT \cdot |-+\rangle = CNOT \circ \frac{1}{2}\begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \end{pmatrix} = |++\rangle$$

$$CNOT \cdot |++\rangle = CNOT \circ \frac{1}{2}\begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} = |-+\rangle$$

$$CNOT \cdot |--\rangle = CNOT \circ \frac{1}{2}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = |--\rangle$$

$$CNOT \cdot |+-\rangle = CNOT \circ \frac{1}{2}\begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \end{pmatrix} = |+-\rangle$$

The operator A is just a $CNOT$ where the inputs are flipped (the right bit is control) and the control arms the gate when its $|0\rangle$, not $|1\rangle$. Using this fact, and the claim we saw earlier, we can do the following:

$$A = (I_2 \otimes \sigma_x) \circ (H_2 \otimes I_2) \circ (I_2 \otimes H_2) \circ CNOT \circ (H_2 \otimes I_2) \circ (I_2 \otimes H_2) \circ (I_2 \otimes \sigma_x)$$

In this formula, $(H_2 \otimes I_2) \circ (I_2 \otimes H_2) \circ CNOT \circ (H_2 \otimes I_2) \circ (I_2 \otimes H_2)$ is a $CNOT$ gates with the control bit as the right bit (because we apply $CNOT$ after change of basis to $\{+-\}$), and we wrap it with a NOT of the right bit to switch the control bit behaviour. To be extra safe, **we ran the numbers** in Octave:

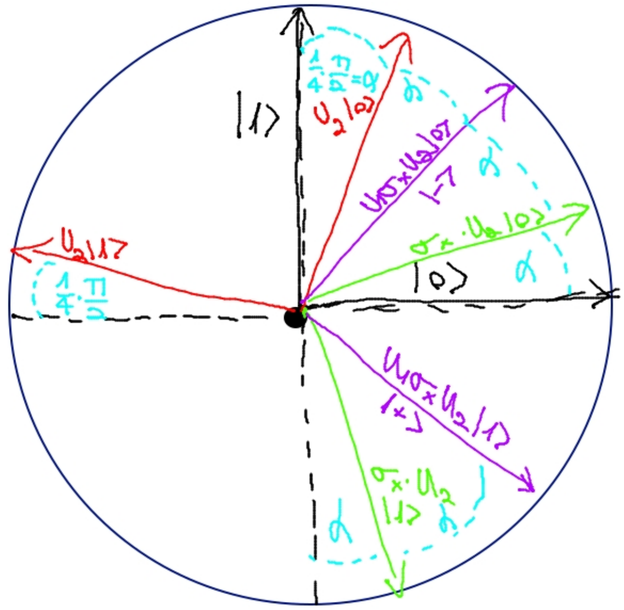
```
>> sxb2 * H * CNOT * H * sxb2
ans =

   0.00000   0.00000   1.00000   0.00000
   0.00000   1.00000   0.00000   0.00000
   1.00000   0.00000   0.00000   0.00000
   0.00000   0.00000   0.00000   1.00000

>> |
```

Figure 1: Octave validation

5. If we find $U_{1,2}$ unitary matrices s.t. $V = U_1 \sigma_x U_2$ and $U_1 U_2 = I$, we also find $c - V = (I_2 \otimes U_1) \circ CNOT \circ (I_2 \otimes U_2)$. Lets think in terms of rotation: Let $U_2 = Rot\left(\frac{3\pi}{2} + \frac{3}{4}\frac{\pi}{2}\right)$ and $U_1 = Rot\left(\frac{3\pi}{2} + \frac{1}{4}\frac{\pi}{2}\right)$. Indeed, $U_1 U_2 = Rot\,(2\pi) = I$. Now, lets see what $U_1 \sigma_x U_2$ does to the states $|0\rangle$ and $|1\rangle$, graphically:



4 Euclid's Algorithm

1. If $r_i \equiv r_{i-2} \mod r_{i-1}$, then $r_i = r_{i-1} \cdot d + r_{i-2}$ for some non-negative natural number d. We get: $\gcd(r_{i-1}, r_i) = \gcd(r_{i-1}, r_{i-1} \cdot d + r_{i-2}) = a$.

   - $a = \gcd(r_{i-1}, r_{i-1} \cdot d + r_{i-2})$ **IFF**
   - $a|r_{i-1}$ (a divides r_{i-1}) and $a|\left(r_{i-1} \cdot d + r_{i-2}\right)$ and no $b > a$ exists that achieve the same **IFF**
   - $r_{i-1} = k \cdot a$ and $r_{i-1} \cdot d + r_{i-2} = t \cdot a$ for some integers k, t and no $b > a$ exists that achieve the same **IFF**
   - $r_{i-1} = k \cdot a$, $k \cdot a \cdot d + r_{i-2} = ta$ for some integers k, t and no $b > a$ exists that achieve the same **IFF**
   - $r_{i-1} = k \cdot a$, $r_{i-2} = (t - kd)\, a$ for some integers k, t and no $b > a$ exists that achieve the same **IFF**
   - $r_{i-1} = k \cdot a$, $r_{i-2} = ma$ for some integers m, k and no $b > a$ exists that achieve the same **IFF**
   - $a|r_{i-1}$, $a|r_{i-2}$ and no $b > a$ exists that achieve the same **IFF**
   - $a = \gcd(r_{i-1}, r_{i-2})$

   To conclude, we get $\gcd(r_{i-1}, r_i) = \gcd(r_{i-2}, r_{i-1})$

2. For some integer $x \geq 0$:

   (a) $x|x$ and $x|0$ (because $1 \cdot x = x$ and $0 \cdot x = 0$....)

   (b) for $y > x$, $y \nmid x$.

   Then by definition, $gcd(x, 0) = x$.

3. We prove the algorithm is **correct** (ends & yields the right result) by **3** claims:

   (a) At each iteration of the algorithm (for any i) it holds that $\gcd(r_{i-1}, r_{i-2}) = \gcd(r_0, r_1)$. Proof: by induction.
   - **base**: for $i = 2$ its an identity.
   - **step**: if we assume $\gcd(r_{i-2}, r_{i-3}) = \gcd(r_0, r_1)$, then by the claim from section (1) we get $\gcd(r_{i-1}, r_{i-2}) = \gcd(r_0, r_1)$.

   (b) The algorithm **stops** eventually. We show this by proving an upper bound $r_{i-1} \leq r_1 - i + 2$. Proof: by induction.
   - **base**: for $i = 2$ its an equality.
   - **step**: Assume $r_{i-2} \leq r_1 - (i-1) + 2$. (notice: we changed the variable name). r_{i-1} is computed, according to the algorithm, by $r_{i-1} = r_{i-3} (\mod r_{i-2})$. this means that $r_{i-1} < r_{i-2}$, and because we are handling integers, $r_{i-1} \leq r_{i-2} - 1$. Plugging it together, we get $r_{i-1} \leq r_1 - i + 2$. Eventually, we reach i big enough so that $r_{i-1} \leq 0$, and the algorithm stops and returns r_{i-2}.

(c) The algorithm returns the correct result. Proof: We know from claim (b) that the algorithm reachs i s.t. $r_{i-1} = 0$. We also know, from claim (a), that for this i it holds $\gcd(0, r_{i-2}) = \gcd(r_0, r_0)$. From section (2), we know that $\gcd(0, r_{i-2}) = r_{i-2}$. Conclusion: The algorithm **stops** at certain i, and returns $r_{i-2} = \gcd(r_0, r_1)$.

■

4. We start by making a helpful claim:

   (a) $r_i < r_{i-1} < r_{i-2}$ for any $i > 2$. Proof: we know that $r_i < r_{i-1}$, because $r_i = r_{i-2} \pmod{r_{i-1}}$. In a similar way, $r_{i-1} < r_{i-2}$ because $r_{i-1} = r_{i-3} \pmod{r_{i-2}}$. (This is why we require $i > 2$ and not $i \geq 2$).

   Now, recall that for $i > 2$, $r_i = r_{i-2} \mod (r_{i-1})$, which means that for some integer q, we can write $r_{i-2} = q \cdot r_{i-1} + r_i$. From the helpful claim, we know that $r_i < r_{i-2}$, so $q \cdot r_{i-1} > 0$. More specifically, $q \geq 1$. So we infer the following in-equality: $r_{i-2} \geq r_{i-1} + r_i$. By using $r_i < r_{i-1}$, we can upgrade our in-equality to $r_{i-2} > 2 \cdot r_i$, which is essentialy $\boxed{r_i < \frac{r_{i-2}}{2}}$. This proves that the algorithm perfoms **at most** $1 + 2\log_2(r_1 + r_0)$ iterations: it must hold that $r_{i-1} > 0$ for any iteration (except the last). if $r_{i-1} < \frac{r_{i-3}}{2}$, then $r_{i-1} < \frac{1}{2^{i/2}}(r_1 + r_0)$. Thus, for r_0 and r_1 of lengths (number of bits) b_0 and b_1, the maximum number of iterations is $\sim 1 + 2(b_0 + b_1)$. (Note: its not a tight bound, only approximate upper bound).