

# CMPE 598 - Lecture 3

Alper Ahmetođlu  
Alptekin Orbay

February 20, 2018

## 1 Recall

We closed the last lecture with showing that QFA can recognize languages that DFA recognize. We also want to know whether there are languages that QFA can recognize while DFA cannot. If that is the case, we prove that QFA is indeed more powerful than DFA in terms of computability.

## 2 QFA on a problem with one-sided error

Consider the following language that is defined on the alphabet  $\Sigma = \{a, b\}$ :

$$L = \{\mathbf{w} \mid \text{the number of } a\text{'s is not equal to the number of } b\text{'s in the input string } \mathbf{w}\}$$

We want to build a machine with one-sided error. It will say "YES" to the members of the language with a non-zero probability, "NO" to the non-members of the language with a  $p = 1$ . One should be careful here. We need infinite precision for our design for the QFA. In real world, this is not the case. Someone may say that quantum machines are superior because of the preciseness. Let us also say that we are somehow able to provide infinite precision for every model, QFA, PFA and DFA.

From our earlier knowledge, we know that complement of a language that can be recognized by a DFA, can also be recognized by some DFA (just swap the accept and reject states). For our convenience let us handle with the task of recognizing:

$$\bar{L} = \{\mathbf{w} \mid \text{the number of } a\text{'s is equal to the number of } b\text{'s in the input string } \mathbf{w}\}$$

We know that this is not a regular language. Therefore there is no DFA that can recognize it. We also know that this is also the case for a PFA since PFA with one-sided error is equivalent to DFA. So let's try to build a QFA.

Our 5-tuple is:  $Q = \{q_1, q_2\}$ ,  $\Sigma = \{a, b\}$ ,  $q_1$  (the initial state),  $F = \{q_2\}$  (set of accept states),  $\{E_a, E_b\}$  (operators), where  $E_a$  is defined as follows:

$$E_a = \begin{bmatrix} 3/5 & -4/5 \\ 4/5 & 3/5 \end{bmatrix}$$

When we have an input string  $a$ , our new state will be  $E_a |q_1\rangle = \frac{3}{5} |q_1\rangle + \frac{4}{5} |q_2\rangle$ . In this case, probability that we are on state  $q_1$  and  $q_2$  are respectively  $\frac{9}{25}$ ,  $\frac{16}{25}$ .  $q_2$  is our accept state. Our machine will say "yes" to the to the input string  $a$  with a non-zero probability. So far it is fine. Our only consideration is that when we give the machine the string  $ab$ , it should say "no" with a probability 1.

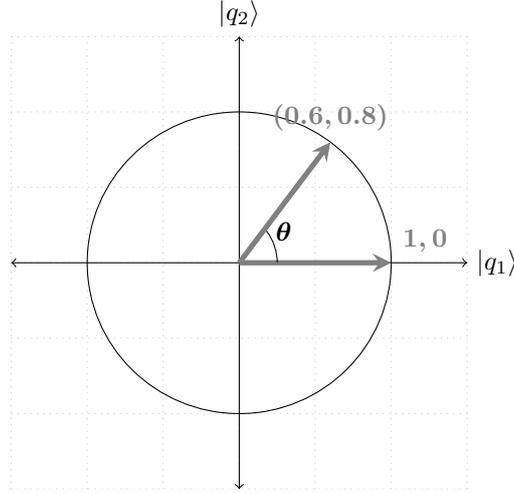


Figure 1: Change of vector  $|q_1\rangle$  when it is multiplied by a rotation matrix

We have not yet defined a transition matrix for the symbol  $b$ . Before constructing this matrix, let's focus on transition matrix for the symbol  $a$ . Columns of this matrix should be orthonormal to each other. This is rather a special matrix, a rotation matrix. We can parameterize it as follows:

$$E_a = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

where  $\theta$  is the angle we rotate our vector in counter-clockwise direction. So every time we see an input symbol  $a$ , our state vector rotates by  $\theta$  angle in the unit circle. When it takes an input symbol  $b$ , it should undo this rotation. This can be easily done by defining the same transition matrix but with  $-\theta$ . If we also select this  $\theta$  to be an irrational multiple of  $\pi$ , then the machine gives "no" answer with a probability of 1 if and only if the number of  $a$ 's are equal to the number of  $b$ 's in the input string  $w$ .

Why do we need  $\theta$  to be an irrational multiple of  $\pi$ ? If  $\theta$  is not an irrational multiple of  $\pi$ , then there is a rational number  $k$  such that  $k\theta = \pi$ . So if our machine will get such an input

$$\underbrace{aa \dots a}_{2k}$$

the initial state will be rotated by  $2k\theta = 2\pi$ . We can also see that  $E_a(0) = E_a(2\pi)$ . So our machine will say no with a probability of 1 although the input string is not in our defined language.

We should also prove that for our specific choice of  $\theta$ , there are no integers  $a, b$  such that  $\frac{\theta}{2\pi} = \frac{a}{b}$ . Let us assume that there are such integers  $a$  and  $b$ . We know that,

$$\cos \theta + i \sin \theta = e^{i\theta}$$

and  $i\theta b = ia2\pi$ . Then  $e^{i\theta b} = 1$ .

$$\left(\frac{3}{5} + \frac{4}{5}i\right)^b = 1$$

$$(3 + 4i)^b = 5^b$$

Modulus operation is extended to complex numbers as follows:

$$(x + yi \equiv q + wi \pmod n) \triangleq (q \equiv x \pmod n) \text{ and } (w \equiv y \pmod n)$$

If two complex numbers are equal, then we also have  $c_1 = c_2 \pmod n$  for any  $n$ . We will use this to show that no such  $b$  exists that satisfy  $(3 + 4i)^b = 5^b$ . However,

$$(3 + 4i)(3 + 4i) = -7 + 24i$$

$$-7 + 24i \equiv 3 + 4i \pmod 5$$

$$(3 + 4i)^b = 3 + 4i \pmod 5$$

Therefore,

$$(3 + 4i)^b \not\equiv 5^b \pmod 5$$

So  $\theta$  is not an irrational multiple of  $\pi$ .

We should note that in order to implement this, we need to represent the  $\theta$  precisely. We also show the "quantum supremacy" only for one-sided bounded error.

### 3 Promise Problems

In this type of problems, it is promised that candidate inputs do not contain certain sort of strings.

#### 3.1 Example

For a natural number  $k$ , we are promised that the input will be of the form  $1^{i2^k}$  for some natural number  $i$ . Alphabet consists of only 1. In this example, strings is accepted only if  $i$  is even and only 1 probability. In other words, error is prohibited. In rtQFA can handle this problem with only two states which are accept state  $|q_1\rangle$  and reject state  $|q_2\rangle$ . This is accomplished with matrix

$$E_1 = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

where  $\theta = \frac{\pi}{2^{k+1}}$  and  $E_1$  is a legitimate matrix for all  $\theta$ . The idea is that after consuming each 1, the current state is rotated counter-clockwise with  $\theta$  angle and intersects with  $|q_1\rangle$  axis when  $i$  is even which means 1 probability.

Eventually, rtQFA is able to solve this problem with 2 state, rtDFA requires many more states for this problem.

**Theorem 1.** *Any DFA solving this problem must have at least  $2^{k+1}$  states.*

*Proof.* Let  $N = 2^k$  and  $D$  be an  $m$ -state DFA solving the problem. We show that  $m$  cannot be less than  $2N$ .

Since both the set of strings to be accepted and the strings to be rejected are infinite, there must be a chain of  $t$  states say  $s_0, s_1 \dots s_{t-1}$ , such that, for sufficiently long strings  $D$  enters this chain in which  $D$  jumps from  $s_i$  to  $s_{(i+1 \pmod t)}$  when reading a 1, for  $0 \leq i \leq t-1$ , and  $0 < t \leq m$ .

Without loss of generality, we assume that  $D$  accepts the input if it is in  $s_0$  at the end. So,  $D$  rejects the input if it is in  $S_{(n \pmod t)}$  at the end.

Let  $S_a$  be the set  $\{s_{(i2N \pmod t)} | i \geq 0\}$ .

Note that  $s_{(N \pmod t)} \notin S_a$ .

Let  $d = \gcd(t, 2N)$ ,  $t' = \frac{t}{d}$ , and let  $S'$  be the set  $\{s_{id} | 0 \leq i \leq t'\}$ .

**Claim 1.**

$$S_a = S'$$

*Proof.* Since  $S_a \subseteq S'$ ,  $\|S'\| = t'$ , we can conclude that  $S_a = S'$  if we show  $\|S_a\| \geq t'$ .

- First, we show that each  $i$  satisfying  $(i2N \equiv 0 \pmod{t})$  must be multiple of  $t'$ . For such an  $i$ , there exists a  $j$  such that  $i2N = jt$ . By dividing both sides by  $t = dt'$ , we get  $\frac{i}{t'} \frac{2N}{d} = j$ . This implies that  $i$  must be a multiple of  $t'$ , since the left hand side must be an integer and  $\gcd(t', 2N) = 1$
- Second, we show that there is no  $i_1$  and  $i_2$  such that  $t' \geq i_1 \geq i_2 \geq 0$ , where  $(i_1 2N \equiv i_2 2N \pmod{t})$ , if so, we have  $(i_1 2N - i_2 2N \equiv 0 \pmod{t})$ , i.e.,  $((i_1 - i_2) 2N \equiv 0 \pmod{t})$ . This implies that  $(i_1 - i_2)$  must be multiple of  $t'$ , which is a contradiction.

$$(i_1 - i_2)2N = kt = kdt'$$

$$\frac{(i_1 - i_2)2N}{d} = kt'$$

Note that  $t'$  and  $\frac{2N}{d}$  relatively prime.

- So, for each  $i \in \{0, \dots, t' - 1\}$ , we obtain a different value of  $(i2N \pmod{t})$ , so it contains at least  $t'$  elements.

Suppose that  $\gcd(t, N) = d$ . Since  $d$  divides  $(N \pmod{t})$ ,  $(N \pmod{t}) = N - kt$ , where  $d$  divides  $N$  and  $kt$ .

$S_{(N \pmod{t})}$  becomes a member of  $S'$  by definition. This is a contradiction since we know that  $S_{(N \pmod{t})}$  should be rejecting, yet  $S' = S_a$  contains only accept states.

So,  $\gcd(t, N)$  must be different than  $d = \gcd(t, 2N)$ .

$N = 2^k$  and  $2^{k+1} = 2N$  if  $t \geq 2N$ . □

□

Therefore, It is impossible that DFA smaller than  $2^{k+1}$  solves this problem. Also, zero error approach cannot reduce state size of PFA as PFA can be converted to DFA as showed in first lecture and DFA must have at least  $2^k + 1$  states as proved above.