

# 1 Query model

In the quantum query model we wish to compute some function  $f$  and we access the input through queries. The complexity of  $f$  is the number of queries needed to compute  $f$  on a worst-case input  $x$ . Unlike the classical case, however, we can now make queries in superposition.

The memory of a quantum query algorithm is described by three registers.

- The input register,  $H_I$ , which holds the input  $x \in \{0, 1\}^n$
- The query register,  $H_Q$ , which holds an integer  $0 \leq i \leq n$
- The working memory,  $H_W$ , which holds an arbitrary value.

The query register and working memory together form the accessible memory, denoted  $H_A$ . Thus the state of the algorithm is described by a vector

$$|\psi\rangle = \sum_{x,i,w} \alpha_{x,i,w} |x, i, w\rangle$$

where  $\sum_{x,i,w} |\alpha_{x,i,w}|^2 = 1$ .

The accessible memory of a quantum query algorithm  $A$  is initialized to a fixed state. For convenience, on input  $x$  we assume the state of the algorithm is  $|x, 0, 0\rangle$  where all qubits in the accessible memory are initialized to 0. The state of the algorithm then evolves through queries, which depend on the input register, and accessible memory operators which do not. We now describe these operations.

There are two common ways to define the notion of a query to the quantum setting where it must be a unitary operation. We will use the model where the oracle answer is given in the phase. This model is a unitary operator  $O$  which is defined by its action on the basis state  $|x\rangle|i\rangle|w\rangle$  as

$$O|x\rangle|i\rangle|w\rangle = (-1)^{x_i} |x\rangle|i\rangle|w\rangle.$$

For every  $x$ , we define  $x_0 = 0$ , thus querying  $i = 0$  is the identity operation or “null query” which is needed for an important technical reason.

An accessible memory operator is an arbitrary unitary operation  $U$  on the accessible memory  $H_A$ . This operation is extended to act on the whole space by interpreting it as  $I_{input} \otimes U$ , where  $I_{input}$  is the identity operation

on the input space  $H_I$ . Thus the state of the algorithm on input  $x$  after  $t$  queries can be written

$$|\phi_x^t\rangle = U_t O U_{t-1} \cdots U_1 O U_0 |x, 0, 0\rangle.$$

As the input register is left unchanged by the algorithm, we can decompose  $|\phi_x^t\rangle$  as  $|\phi_x^t\rangle = |x\rangle|\psi_x^t\rangle$ , where  $|\psi_x^t\rangle$  is the state of the accessible memory after  $t$  queries.

The output of a  $T$ -query algorithm  $A$  on input  $x$  is chosen according to a probability distribution which depends on the final state of the accessible memory  $|\psi_x^T\rangle$ . Namely, the probability that the algorithm outputs the bit  $b \in \{0, 1\}$  on input  $x$  is  $\|\Pi_b|\psi_x^T\rangle\|^2$ , for a fixed set of projectors  $\{\Pi_b\}$  which are orthogonal and complete, that is, sum to the identity. The  $\epsilon$ -error quantum query complexity of a function  $f$ , denoted  $Q_\epsilon(f)$ , is the minimum number of queries made by an algorithm which outputs  $f(x)$  with probability at least  $1 - \epsilon$  for every  $x$ .

## 2 Reduced density matrix

Consider a bipartite quantum system described by a density matrix  $\rho_{AB}$ . A *reduced density matrix*  $\rho_A$  is the density matrix of the subsystem  $A$  alone, when we *trace out* the other subsystem, i.e. forget about it or measure it without reading the outcome. One can compute  $\rho_A$  from  $\rho_{AB}$  using the operation called *partial trace*:  $\rho_A = \text{Tr}_B(\rho_{AB})$ .

Partial trace is a linear operator  $\text{Tr}_B : L(A \otimes B) \rightarrow L(A)$  mapping density matrices over the joint system  $AB$  to density matrices over  $A$ . We require that tracing out an unentangled system  $B$  leaves  $A$  as is, that is

$$\text{Tr}_B(\rho_A \otimes \rho_B) = \rho_A \cdot \text{Tr}(\rho_B).$$

This requirement uniquely characterizes the operation. For example, since  $\text{Tr}_B(|i\rangle\langle j|_A |\psi_i\rangle\langle\psi_j|_B) = |i\rangle\langle j| \cdot \langle\psi_j|\psi_i\rangle$ , by linearity

$$\text{Tr}_B\left(\sum_{i,j} |i\rangle\langle j|_A |\psi_i\rangle\langle\psi_j|_B\right) = \sum_{i,j} |i\rangle\langle j| \cdot \langle\psi_j|\psi_i\rangle.$$

## 3 Adversary lower bounds

Consider a bounded-error algorithm. Let  $|\psi_x^t\rangle$  denote the state of the algorithm after  $t$  queries. Since the computation starts in a fixed state,  $|\psi_x^0\rangle =$

$|\psi_y^0\rangle$  for all  $x, y$ , that is  $\langle\psi_x^0|\psi_y^0\rangle = 1$ . On the other hand, to be able to estimate the function output at the end of the computation by a measurement, we require that whenever  $f(x) \neq f(y)$ ,  $|\langle\psi_x^T|\psi_y^T\rangle| \leq c$  for some constant  $c < 1$ , otherwise no measurement could possibly distinguish these two states with good probability.

The *adversary lower bound* technique shows that these scalar products cannot change too much on the average, hence one needs many queries to sink from 1 to below  $c$ . We use a weighted average over input pairs that needs to be distinguished as a progress function. In particular, we define

$$W^t = \sum_{x,y} w_{x,y} \langle\psi_x^t|\psi_y^t\rangle, \tag{1}$$

where  $w(x, y) = 0$  for  $f(x) = f(y)$ ,  $w \geq 0$  are non-negative weights, and  $w$  is symmetric.

Historically, the first such lower bound is due to Bennett, Bernstein, Brassard, and Vazirani for the OR function. They put the all-zeroes input string on one side with  $n$  input strings of Hamming weight 1 on the other side, and computed an unweighted average of these scalar products. They got a tight  $\Omega(\sqrt{n})$  lower bound.

Ambainis extended this technique to all functions by considering a general bipartite graph with some 0-inputs on one side, and some 1-inputs on the other side. An edge between two input strings means that these two input strings need to be distinguished and thus are included in the average. One typically does not want to put edges between all input pairs, but only between those that are hard to distinguish, i.e. have small Hamming distance. This leads to better lower bounds.

Later, Barnum, Saks, and Szegedy, and independently Ambainis strengthened this method by allowing weights on the edges. This finer scale allows for much better lower bounds than just two options include an edge/don't include.

We express the progress function Equation (1) in terms of the reduced density matrix of the input register. Then we prove bounds on it.

## 4 State of computation

Denote the state of computation on input  $x$  after  $t$  queries by

$$|\psi_x^t\rangle = \sum_i |i\rangle_Q |\psi_{x,i}^t\rangle_W,$$

where  $|\psi_{x,i}\rangle$  is the substate corresponding to querying the  $i$ -th input bit in the next query. Note that  $|\psi_{x,i}\rangle$  is not normalized, but we instead absorb the complex coefficients into it.

We run the computation on a general superposition of inputs  $|\delta\rangle = \sum_x \delta_x |x\rangle_I$ , hence the global state after  $t$  queries is

$$|\psi^t\rangle = \sum_x \delta_x |x\rangle_I \sum_i |i\rangle_Q |\psi_{x,i}^t\rangle_W.$$

The density matrix  $\rho_{IQW}$  and the reduced density matrices  $\rho_{IQ}$  and  $\rho_I$  are

$$\begin{aligned} \rho_{IQW} &= \sum_{x,y} \delta_x \delta_y^* |x\rangle\langle y| \sum_{i,j} |i\rangle\langle j| \otimes |\psi_{x,i}^t\rangle\langle\psi_{y,j}^t|, \\ \rho_{IQ} &= \text{Tr}_W(\rho_{IQW}) = \sum_{x,y} \delta_x \delta_y^* |x\rangle\langle y| \sum_{i,j} |i\rangle\langle j| \cdot \langle\psi_{y,j}^t|\psi_{x,i}^t\rangle, \\ \rho_I &= \text{Tr}_{IQ}(\rho_{IQW}) = \sum_{x,y} \delta_x \delta_y^* |x\rangle\langle y| \sum_i \langle\psi_{y,i}^t|\psi_{x,i}^t\rangle = \sum_{x,y} \delta_x \delta_y^* |x\rangle\langle y| \cdot \langle\psi_y^t|\psi_x^t\rangle \end{aligned}$$

Take any *adversary matrix*  $\Gamma$  (i.e., satisfying  $\Gamma[x,y] \geq 0$ ,  $\Gamma$  is symmetric, and  $\Gamma[x,y] = 0$  for  $f(x) \neq f(y)$ ) and define

$$W^t = \langle\Gamma, \rho_I^t\rangle = \sum_{x,y} \Gamma_{x,y} \delta_x \delta_y^* \langle\psi_y^t|\psi_x^t\rangle.$$

We see that  $W^t$  is exactly the progress function from Equation (1) with  $w_{x,y} = \Gamma_{x,y} \delta_x \delta_y^*$ . The reason why we re-express  $W^t$  using the (seemingly more complicated) formalism of density matrices is that one can easily prove bounds on it once it's expressed this way.

## 5 Spectral lower bound

Consider an adversary matrix  $\Gamma$ . Let  $\delta$  be the principal eigenvector of  $\Gamma$ , i.e.  $|\delta| = 1$  and  $\Gamma\delta = \|\Gamma\|\delta$ . We run the computation on the superposition

of inputs  $|\delta\rangle$  and track the value of the progress function. Thus in a way  $|\delta\rangle$  is a hard input superposition, compare to the hard input distribution for classical probabilistic computation. The following holds:

1.  $W^0 = \langle \Gamma, \rho_I^0 \rangle = \langle \delta | \Gamma | \delta \rangle = \|\Gamma\| \langle \delta | \delta \rangle = \|\Gamma\|$ , because all scalar products are one.
2. Since  $\Gamma[x, y] \geq 0$ , it's easy to show that the principal eigenvector  $\delta$  is also non-negative. If the algorithm has bounded error at most  $\varepsilon$ , then all scalar products at the end have to be at most  $c = 2\sqrt{\varepsilon(1-\varepsilon)}$  (Bernstein, Vazirani). Hence

$$\begin{aligned}
|W^T| &= |\langle \Gamma, \rho_I^T \rangle| \\
&= \left| \sum_{x,y} \Gamma_{x,y} \delta_x \delta_y^* \langle \psi_y^T | \psi_x^T \rangle \right| \\
&\leq \sum_{x,y} \Gamma_{x,y} \delta_x \delta_y^* |\langle \psi_y^T | \psi_x^T \rangle| \\
&\leq c \sum_{x,y} \Gamma_{x,y} \delta_x \delta_y^* = c \langle \delta | \Gamma | \delta \rangle \\
&= c \|\Gamma\|.
\end{aligned}$$

3. To upper-bound  $|W^{t+1} - W^t|$ , we need to look at the effect of a query on the quantum state. Looking at  $\rho_I$  is not sufficient, because we need to consider the query register. Hence we re-express  $W^t$  in terms of  $\rho_{IQ}$  instead of just  $\rho_I$  alone. This requires just one more summation. We immediately get that

$$W^t = \langle \Gamma, \rho_I^t \rangle = \langle \Gamma \otimes I_n, \rho_{IQ}^t \rangle,$$

where  $\Gamma \otimes I_n$  is a block-diagonal matrix with  $n$  copies of  $\Gamma$  on the main diagonal, each corresponding to one value  $i \in \{1, \dots, n\}$  of the query register. Recall that

$$\rho_{IQ} = \sum_{x,y} \delta_x \delta_y^* |x\rangle \langle y| \sum_{i,j} |i\rangle \langle j| \cdot \langle \psi_{y,j}^t | \psi_{x,i}^t \rangle,$$

After one query and an arbitrary unitary  $U$ , the state is mapped to  $|x, i\rangle |\psi_{x,i}\rangle \rightarrow (-1)^{x_i} |x, i\rangle U |\psi_{x,i}\rangle$ . In the density matrix  $\rho_{IQ}$ , the unitary  $U$  in the scalar product cancels with itself:  $\langle \psi_{y,j}^t | U^* U | \psi_{x,i}^t \rangle =$

$\langle \psi_{y,j}^t | \psi_{x,i}^t \rangle^1$ , and we get

$$\rho'_{IQ} = \sum_{x,y} \delta_x \delta_y^* |x\rangle \langle y| \sum_{i,j} |i\rangle \langle j| \cdot (-1)^{x_i+y_j} \langle \psi_{y,j}^t | \psi_{x,i}^t \rangle,$$

which can be rewritten as

$$= Z \rho_{IQ} Z,$$

where  $Z = Z_1 \oplus \dots \oplus Z_n$  is a unitary diagonal matrix consisting of  $n$  blocks  $Z_i$  with  $Z_i[x, x] = (-1)^{x_i}$ . Note that the oracle query does not depend on the algorithm workspace  $W$  and this allows us to track the progress function even after tracing out the register  $W$ . Now, subtract  $\rho$  from  $\rho'$ :

$$\rho'_{IQ} - \rho_{IQ} = 2 \sum_{x,y} \delta_x \delta_y^* |x\rangle \langle y| \sum_{\substack{i,j \\ x_i \neq y_j}} |i\rangle \langle j| \cdot \langle \psi_{y,j}^t | \psi_{x,i}^t \rangle,$$

because the other terms get cancelled.

We are ready to bound the progress function. Let  $G = \Gamma \otimes I_n$  and denote  $\rho = \rho_{IQ}^t$  and  $\rho' = \rho_{IQ}^{t+1}$ .

$$W^{t+1} - W^t = \langle G, \rho' - \rho \rangle$$

Let  $A \circ B$  denote the entry-wise product  $(A \circ B)[x, y] = A[x, y] \cdot B[x, y]$ . Let  $D = D_1 \oplus \dots \oplus D_n$  be a 0-1 block-diagonal matrix with  $D_i[x, y] = 1$  iff  $x_i \neq y_i$  on the main diagonal. Thanks to the cancellation property above we can replace  $G$  by  $G \circ D$  and the scalar product stays the same—the entries zeroed-out on the left are exactly those that are already zero on the right. However  $G \circ D$  has smaller spectral norm,

---

<sup>1</sup>Note that this only works thanks to the unitarity of quantum computing. Here we crucially use that unitary operations only have complex units as eigenvalues.

which is what we want.

$$\begin{aligned}
W^{t+1} - W^t &= \langle G \circ D, \rho' - \rho \rangle && \langle A, B \rangle \leq \|A\| \cdot \|B\|_{tr} \\
&\leq \|G \circ D\| \cdot \|\rho' - \rho\|_{tr} && \rho' = Z\rho Z \\
&= \|G \circ D\| \cdot \|Z\rho Z - \rho\|_{tr} && \text{triangle inequality} \\
&\leq \|G \circ D\| \cdot (\|Z\rho Z\| + \|\rho\|_{tr}) && Z \text{ is unitary} \\
&= 2\|G \circ D\| \cdot \|\rho\|_{tr} && \rho \text{ is a density matrix} \\
&= 2\|G \circ D\| \\
&= 2 \max_i \|G \circ D_i\|.
\end{aligned}$$

We conclude that if  $\Gamma$  is an adversary matrix, then

$$Q_\varepsilon(f) \geq \left( \frac{1}{2} - \sqrt{\varepsilon(1-\varepsilon)} \right) \frac{\|\Gamma\|}{\max_i \|\Gamma_i\|}.$$

This result is due to Barnum, Saks, and Szegedy.

## 6 Applications

### 6.1 Unordered search

Consider  $(n+1) \times (n+1)$  matrix  $\Gamma$  indexed by the following input strings: all-zeroes string 0, and  $n$  strings  $e_i$  with exactly one 1. Then set all weights between 0 and  $e_i$  to 1. Then  $\|\Gamma\| = \sqrt{n}$  and  $\|\Gamma \circ D_i\| = 1$ .

### 6.2 Ordered search

This is more interesting, because here the weights are crucial to get a good bound. Consider  $(n+1) \times (n+1)$  matrix  $\Gamma$  indexed by the input strings of the form  $0^i 1^{n-i}$  for  $i = 0, \dots, n$ . The output of the algorithm should be  $i$ . Here we only require that the main diagonal is zero. We want to put more weight on input pairs with small Hamming distance, because those are harder to distinguish. On the other hand, we cannot put all weight to Hamming distance 1, because then the adversary bound turns out to be constant.

A natural choice is setting  $\Gamma[x, y] = \frac{1}{|x-y|}$ , i.e. the weight is reciprocal to the Hamming distance. Now,  $\|\Gamma\| = \Omega(\log n)$ , witnessed by multiplying  $\Gamma$  from both sides by  $\delta = (1, \dots, 1)$ . On the other hand,  $\|\Gamma \circ D_i\|$  is a submatrix

of the (infinitely large) Hilbert matrix  $H[x, y] = \frac{1}{x+y-1}$ , which has spectral norm  $\|H\| = \pi$ , hence  $\|\Gamma \circ D_i\| \leq \pi$ .