The Structure Necessary for Quantum Speedups

Shalev Ben-David

University of Maryland, College Park

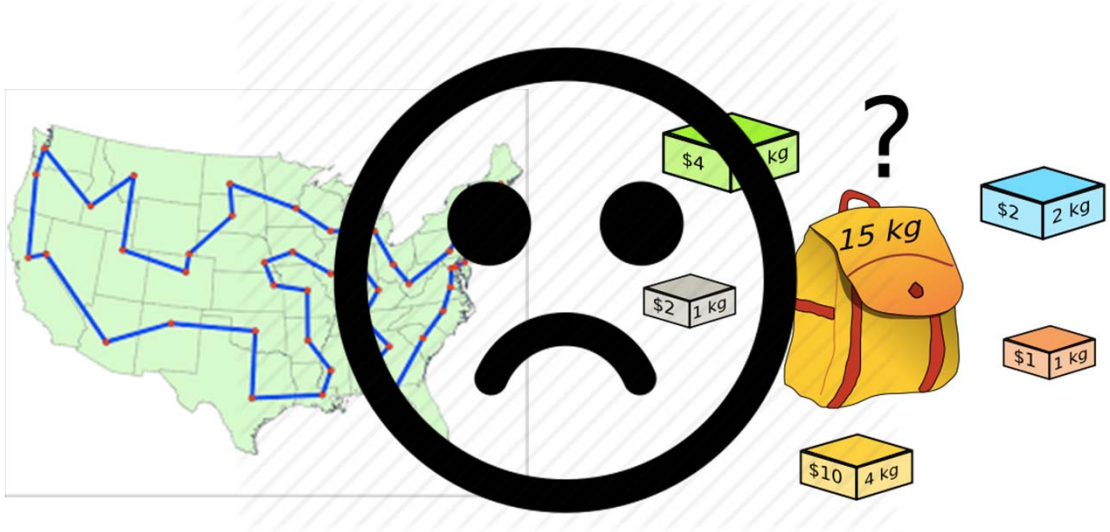
What can we do with quantum computers?

- Quantum computers seem to give exponential speedups for some problems
- Shor's algorithm (1994): factoring in polynomial time

- Caveat: we can't rule out a fast classical algorithm

What can we do with quantum computers?

- Quantum computers don't seem to give exponential speedups for NP-complete problems



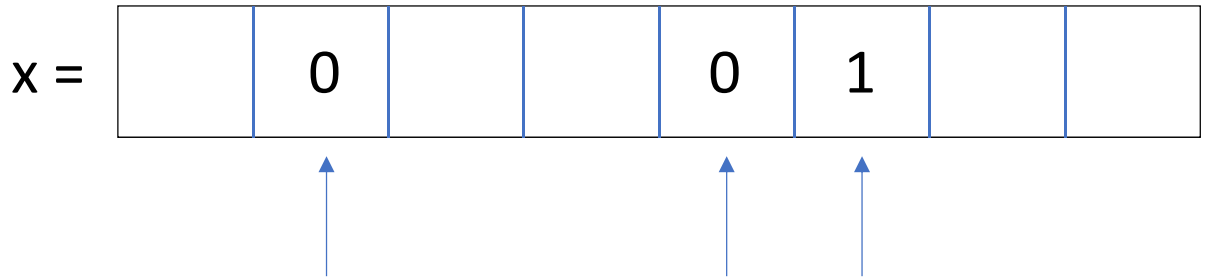
- Caveat: we can't prove this either
- Grover's algorithm (1996): search n elements in √n time
- Intuition: "structure" is necessary for exponential speedups

But what kind of structure?

- Some possibilities:
  - Quantum hates us: speedups are only possible for tasks we don't want solved
  - There is some underlying pattern
- Difficulty: we can't prove, like, anything
  - Can't even settle P vs NP or BPP vs BQP
  - How can we hope to tackle more structural questions?
- Solution: query complexity
  - Query complexity is easy, so anything we can't solve there is our problem

The query (blackbox) model

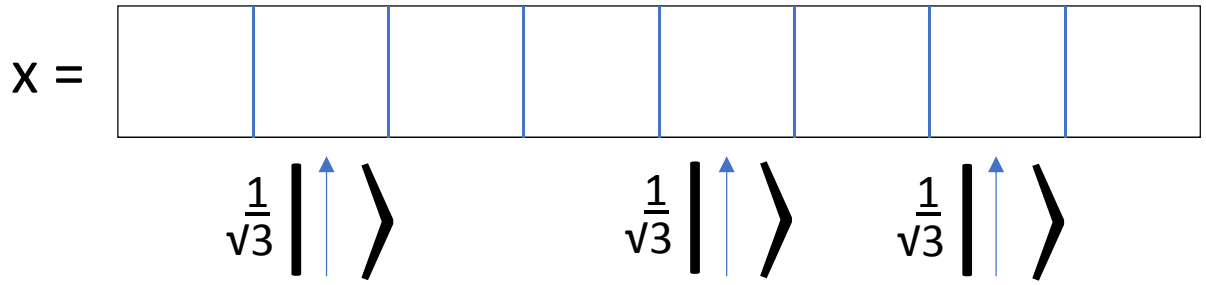
- There is a known function $f:\{0,1\}^n \longrightarrow \{0,1\}$, say f = OR
- Goal: compute f(x) by making queries to the bits of x



- The worst-case number of queries is denoted by
  - D(f) for deterministic algorithms
  - R(f) for randomized algorithms
  - Q(f) for quantum algorithms

The query (blackbox) model

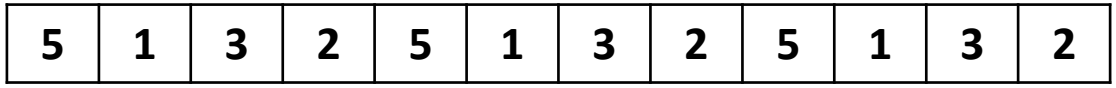
- There is a known function $f:\{0,1\}^n \longrightarrow \{0,1\}$, say f = OR
- Goal: compute f(x) by making queries to the bits of x



- The worst-case number of queries is denoted by
  - D(f) for deterministic algorithms
  - R(f) for randomized algorithms
  - Q(f) for quantum algorithms

Advantages of the query model

- We can actually prove things
- It's the very simplest model
- Captures most quantum algorithms
  - Grover's algorithm
  - Shor's algorithm works by reducing factoring to period finding, and period finding is a query problem



- Has plenty of connections with the rest of complexity theory

The role of promises

- Shor's period-finding algorithm required a promise
  - The function is periodic
- Grover's algorithm did not require a promise
- With a promise, exponential randomized speedups are also possible (is the string 2/3 ones or 2/3 zeros?)
- Can construct f with $Q(f)=1$, $R(f)=\Omega\tilde{}(n^{1/2})$ [AA'14]
- Functions with a promise are also called partial functions

Total functions

- Functions with no promise
- Beals, Buhrman, Cleve, Mosca, de Wolf ‘98:

$$Q(f) \leq R(f) \leq D(f) \leq Q(f)^6$$

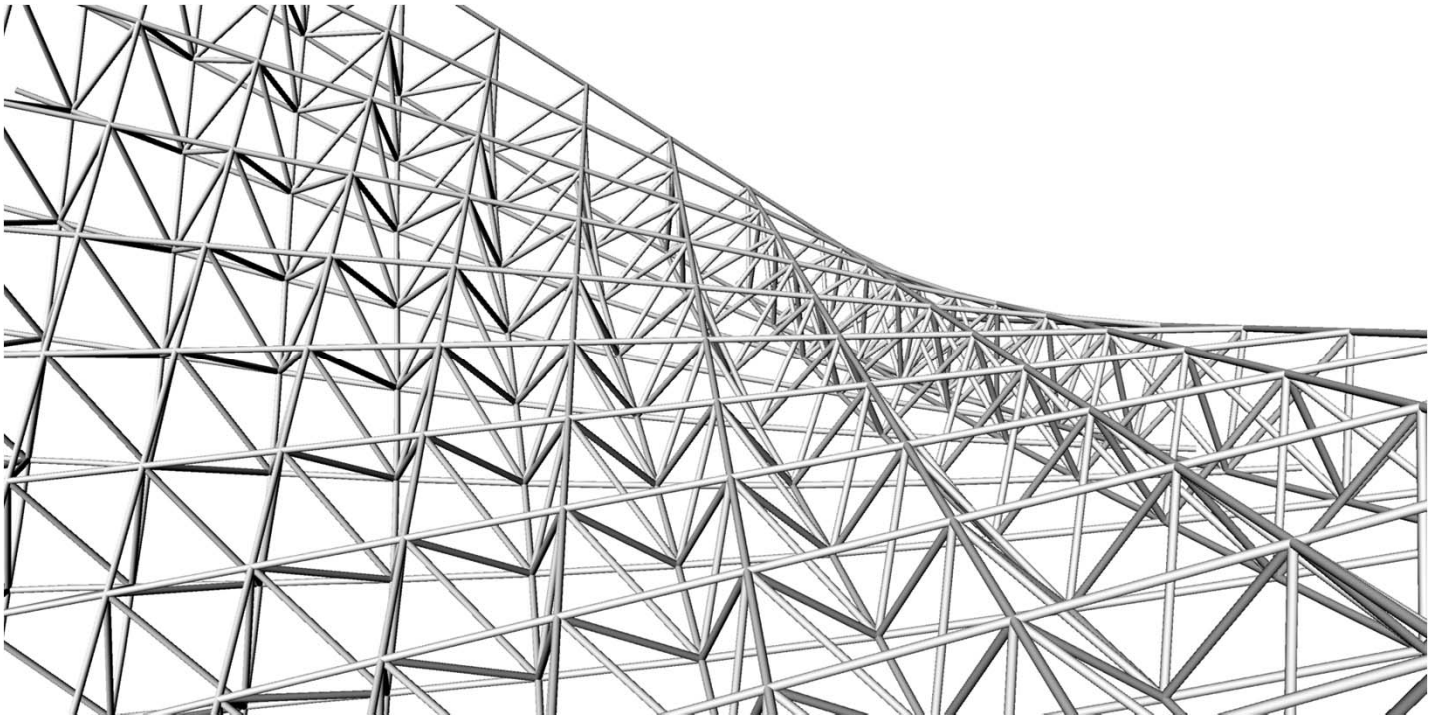
- For total functions, no exponential speedups at all!

On the previous episode of quantum query complexity

- [AA '14]: Partial function with
  $Q(f) = O(1)$ $R(f) = \Omega\tilde{}(n^{1/2})$
- [ABBLSS '15]: Total function with
  $D(f) \approx R(f)^2 \approx Q(f)^4$
  (previous best known was 1.3 and 2)
- [B. '15, ABS '15]: Total function with
  $R(f) \approx Q(f)^{2.5}$
- We're quibbling over polynomial factors!
  - Real complexity theorists only care about super-polynomial speedups

The Structure Program

Characterize the structure necessary for super-polynomial quantum speedups



Douglas Adams:

“The History of every major Galactic Civilization tends to pass through three distinct and recognizable phases, those of Survival, Inquiry and Sophistication, otherwise known as the How, Why, and Where phases.

For instance, the first phase is characterized by the question 'How can we eat?’
the second by the question 'Why do we eat?’
and the third by the question 'Where shall we have lunch?’”

How, Why, and Where for Quantum Query Complexity

- How much speedup is possible?
  - Exponential for partial functions, polynomial for total functions
- Why are speedups possible?
  - Because of entanglement and stuff
- Where are exponential speedups possible?
  - On what functions can you expect them?

Symmetric Functions

- Collision problem:

5	6	11	7	9	3	8	4
5	2	1	5	2	1	20	20

- Intuitively structureless
- Aaronson ‘02: no exponential quantum speedup
- Can we generalize this to other symmetric functions?

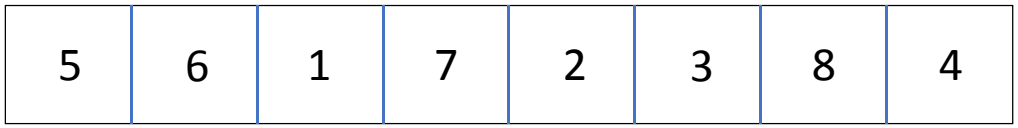
Symmetric Functions

- Can we generalize this to other symmetric functions?
- Aaronson-Ambainis ‘09: Yes
- If a function f is doubly symmetric
  - f(x) does not change if we permute the string x
  - f(x) does not change if we rename all the alphabet elements, e.g. exchange `2’ with `11’ everywhere
- Then $R(f) = \tilde{O}(Q(f)^7)$

- Note: this requires both the function and the promise to be symmetric

Permutation promises

- Permutation inversion: given a permutation, find the 1



- Intuitively structureless
- Ambainis ’00: no super-polynomial quantum speedup
- Can we generalize this to other functions defined on a permutation promise?

Permutation promises

5	6	1	7	2	3	8	4

- Can we generalize this to other functions defined on a permutation promise?
- B. ’14: Yes
- For any function f defined on a permutation promise, $D(f) = O(Q(f)^{16})$
  - Is the permutation even or odd?
  - Find the length of the smallest cycle
  - Etc.

Proof Idea

- Start with the proof of $D(f) = O(Q(f)^6)$ for total functions
- Sketch:
  - If Q(f) is small, there are no large copies of the OR function in f
  - If there are no large copies of OR in f, then there are small certificates for each input
  - If there are small certificates for each input, then D(f) is small
- Issue: on permutations, certificates can be much smaller than D(f)!
  - E.g. “find the 1”: C(f)=1, D(f)=n

Proof Idea

- Key lemma:
  - There is a small set $S \subseteq \{1,2,\ldots,n\}$ such that all small certificates reveal a number in S
- If certificates are small, the problem sort of looks like permutation inversion!
- For permutation inversion, we have quantum lower bounds [Ambainis ‘00]

Can we generalize both?

- Aaronson-Ambainis ’09 handles any symmetric promise, but only for symmetric functions
- B. ‘14 handles only the permutation promise, but for any function
- The set of permutations is an orbit of the symmetric group action
- B. ‘14 generalizes to any promise that is an orbit of the symmetric group action
  - It even generalizes to constant-sized unions of orbits
- Open: generalize to arbitrary symmetric promises

Can we generalize both?

- Open: generalize to arbitrary symmetric promises
- B. '14 solves this for the special case where the alphabet size is constant
  - E.g. if the alphabet is {0,1}, a symmetric promise is a promise on the Hamming weight of the string
  - "The input string has Hamming weight 3 or n/2"
- Idea:
  - use a randomized algorithm to estimate the Hamming weight
  - The possible remaining Hamming weights are now small union of orbits

Sculpting

Sculpting

- Fix a function f in advance and choose the promise afterwards. For which f can you engineer a quantum speeup?
- [Aaronson, B. ‘16] characterize all the total functions that allow an exponential speedup given some promise
- The promise may be quite artificial
- If a promise exists, we say f can be sculpted

In other words: there is probably no quantum speedup for 3-SAT. But is there a set of instances of 3-SAT that are particularly quantum-friendly?

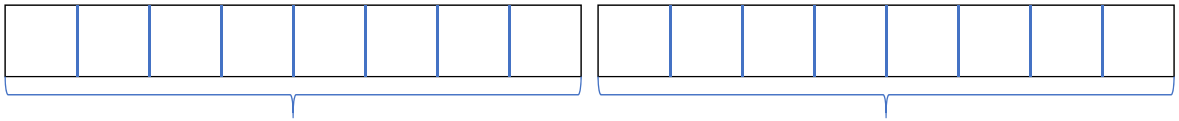
Example: OR

- Can we restrict OR to a promise such that on inputs from that promise, there is an exponential quantum speedup?

- Aaronson ‘04: No. Quadratic speedup on all promises

Example: parity

- Can we restrict parity to a promise such that on inputs from that promise, there is an exponential quantum speedup?



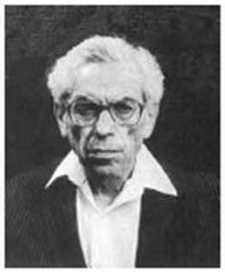
Period finding

Promise: input here makes the total parity equal to the value of period finding

H Index

- Used to measure research output
- Maximum number k such that you have at least k publications with at least k citations each

- H Index variant: maximum number k such that you have at least 2^k publications with at least k citations each



Paul Erdős

Follow

Mathematics

number theory, combinatorics, probability, set theory, mathematical analysis

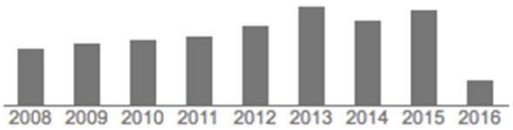
No verified email - Homepage

Title 1–20	Cited by	Year
On random graphs I. P ERDdS, A R&WI Publ. Math. Debrecen 6, 290-297	11979 *	1959
On the evolution of random graphs P Erdos, A Rényi Bull. Inst. Internat. Statist 38 (4), 343-347	~~7475~~	1961
On random graphs P Erdős, A Rényi Publicationes Mathematicae Debrecen 6, 290-297	~~6819~~ *	1959
A combinatorial problem in geometry P Erdös, G Szekeres Compositio Mathematica 2, 463-470	1209	1935

Google Scholar

Get my own profile

Citation indices	All	Since 2011
Citations	[illegible]	21453
h-index	108	59
i10-index	[illegible]	328



Co-authors View all…

Ralph Faudree

András Sárközy

Janos Pach

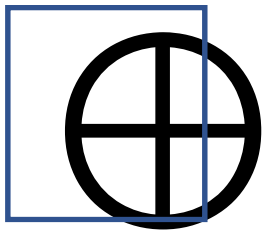
Laszlo Lovasz

$\bigvee$ OR_n

Publications (inputs)	(value)	Citations (certificates)
00…0000	0	n
00…0001	1	1
00…0010	1	1
00…0011	1	1
00…0100	1	1
00…0101	1	1
00…0110	1	1
…	…	…
11…1111	1	1

2^n

Most cited (certificate complexity)	n
h-index	1



$PARITY_n$

Publications (inputs)	(value)	Citations (certificates)
00…0000	0	n
00…0001	1	n
00…0010	1	n
00…0011	0	n
00…0100	1	n
00…0101	0	n
00…0110	0	n
…	…	…
11…1111	?	n

2^n

Most cited (certificate complexity)	n
h-index	n

Characterization Result

- $H(C_f)$ is the H-index of the vector of certificate sizes for f
- "Sculpting quantum speedups is possible iff $H(C_f)$ is large"

$$\forall_f \; \exists_P \;\; R(f|_P) = \Omega\left(\frac{H(C_f)^{1/6}}{\log^3 n}\right), \quad Q(f|_P) = O(\log^2 H(C_f))$$

$$\forall_f \; \forall_P \;\; R(f|_P) = O(Q(f|_P)^2 H(C_f)^2)$$

- Sculpting randomized speedups is <u>always</u> possible

Why Certificates?

- Actually, the sculpting construction uses $H(bs_f)$ instead of $H(C_f)$
- The two are quadratically related

- Intuitively, these measure whether the function is difficult in only one spot (like OR), or everywhere (like parity)

Proof sketch: sculpting impossibility

- Want to show $R(f|_P)=O(Q(f|_P)^2 H(C_f)^2)$
- "If there are few large certificates, R and Q are quadratically related"

- Step 1: use the standard $D \leq C^2$ algorithm to kill small certificates
- we have few 1-inputs left
- Step 2: show that $R \leq Q^2$ on any function with few 1-inputs

Side Result

$$Q(f) = \Omega\left(\frac{\sqrt{D(f)}}{\log|Dom(f)|}\right)$$

- Example: OR
- Proof idea: generalize $RC \leq QC^2$, and show C=RC when the domain is small

Proof sketch: sculpting existence

- Given f, want P such that
  $R(f|_P) \geq \text{poly}(H(C_f)), \quad Q(f|_P) \leq \text{polylog } H(C_f)$

- "If there are many hard inputs, there is a promise P with exponential quantum speedup for $f|_P$"

- Step 1: replace $H(C_f)$ with $H(bs_f)$
- Step 2: Sauer's lemma
- Step 3: reduce to communication

Step 2: Sauer's lemma

- For any $S \subseteq \{0,1\}^n$, there is a set of bits of size
  ~ log |S|/log n with all possible actions

001000
101111
110001
101110
101010

Step 2: Sauer's lemma

- Hard inputs look like:

- The x part can be any string
- Since there are many hard inputs, the x part is large
- We define a promise problem on the x part that has a quantum speedup
- What if the s(x) part lets the classical algorithm cheat?
- Is it possible for s(x) to contain the answers to all possible problems that give a quantum speedup?

Step 3: reducing to Communication

- Hard inputs look like:

- Take a communication task that can be solved quantumly but not randomly (Klartag and Regev 2011)
- Give x to Bob
- Give a different string y to Alice so that (x,y) satisfies the promise
- Consider strategies in which Alice sends Bob randomized queries to x or s(x) (log n bits each)
- This strategy must fail for some y; this y defines the desired function

Sculpting Conclusions

- A full characterization of sculpting: which problems can be restricted to a promise that gives rise to an exponential quantum speedup
- "Quantum computers give an exponential speedup for *some* 3-SAT instances"

  ✔ **Complexity Theorist Approved**

- Most Boolean functions are sculptable
- "Quantum speedups are not about the function, they are about the promise"

Promises summary: Which promises don't give exponential speedups?

- Null promise [BBCMdW '98]
- Very small promise sets [Aaronson, B. '16]
- Hamming weight promises [B. '14]
- Permutation promises [B. '14]
- Random promises [unpublished]
  - Only with constant probablity of including each string

- Wanted: full characterization, like for sculpting

Approachable open problems

- Show that any function defined on a symmetric promise set does not exhibit exponential quantum speeups
  - Generalizes AA09 and B14
- Is there a search problem (TFNP) that has an exponential quantum speedup?
  - Possible applications to proof complexity
- Is there an exponential quantum speedup on small random promises?
- Is there an exponential quantum speedup on a graph property (adjacency matrix model)?

Thanks!

- Idea: check if f has a 0-input that can be changed to a 1-input by modifying many different locations (or a 1-input that can be changed to a 0-input).
  - If so, prove a quantum lower bound using the lower bound on Grover search
  - If not, construct a fast deterministic algorithm

Specific structureless functions

- Blackbox search
  - Even when promised the number of marked items is k
- Collision problem
  - Are the items unique, or does each occur twice?
- Permutation inversion

Generalizing Collision

- Aaronson-Ambainis:
- Any fully-symmetric function does not give rise to an exponential quantum speedup
  - $R(f) \leq \tilde{O}(Q(f)^7)$
  - A fully-symmetric function can be a promise problem, but the promise also has to be symmetric
  - Collision is fully-symmetric

- Note: requires both the function and the promise to be structureless

Generalizing permutation-inversion

- Let P be the set of permutations:

- No function defined on P exhibits an exponential quantum speedup
  - $R(f) \leq O(Q(f)^{16})$
- Open: Does any function defined on a fully-symmetric promise exhibit an exponential quantum speedup?
  - True if the alphabet size is constant
  - If the promise is only on the Hamming weight of a Boolean string, there is no exponential speedup

Versions of the structure question

Version A: characterize all the partial functions admitting an exponential randomized/quantum speedup.

- Too hard

Version B: characterize all the promises on which some function has an exponential speedup.

- No characterization, but some progress

Version C: characterize all the total functions that allow an exponential speedup given some promise.

- Solved! (Aaronson, B. ‘16)

Give examples of promises

Mention random promise result