Programming Quantum Computers (Apps II: Search)

(**Subtrack of** Quantum Computing: An App-Oriented Approach)

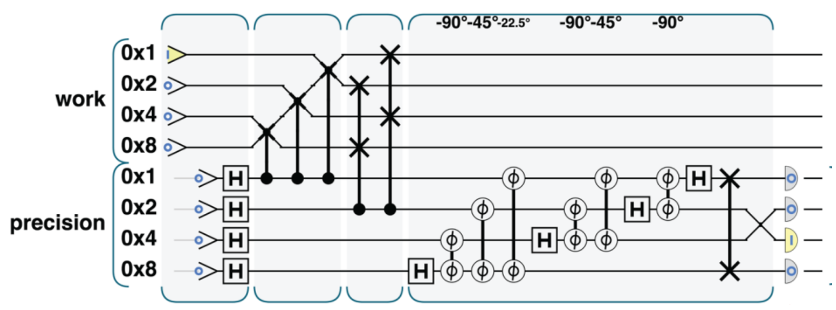
Moez A. AbdelGawad

moez@{cs.rice.edu, alexu.edu.eg, srtacity.sci.eg}

Sat., Dec. 21^{th}, 2019

Quantum Computers are Real

- What are they useful for?
  - Let's discover, by programming them!
  - And seeing *examples* of how others programmed them.
- A hands-on approach to programming QCs/QPUs.
  - By doing; i.e., by writing code & building programs.
  - Using simulators, since real QCs are harder-to-access (so far).
- Goals: Read, understand, write, and *debug* quantum programs.
  - Ones like this program.



Topics Covered So Far

- Introduction:
  - Qubit, Superposition, and Entanglement.
  - Single-Qubit Ops: H, NOT and Phase.
  - Multi-Qubit Ops: Conditional Ops (e.g., CNOT).
  - Teleportation.

- Modules:
  - Quantum Arithmetic and Logic.
  - (Quantum) Amplitude Amplification.
    - Converting phase info into magnitude info.
  - Quantum Fourier Transform.
    - Revealing patterns (frequencies).
  - (Quantum) Phase Estimation.
    - Characterization of quantum operations.
  - Quantum Simulation and Real Data.
    - QRAM, Quantum Vector & Matrix Encodings.

Quantum Apps

- Quantum Simulation.
  - Using quantum operations to approximate unitary matrices that describe quantum operations representing Hermitian matrices (the Hamiltonians).
- **Quantum Search (Grover's algorithm).**
  - Using quantum phase logic and amplitude amplification to check the satisfiability of logical formulas.
- Quantum Graphics (Quantum Supersampling).
- Quantum Cryptography (Shor's algorithm).
- Quantum Machine Learning (QML).

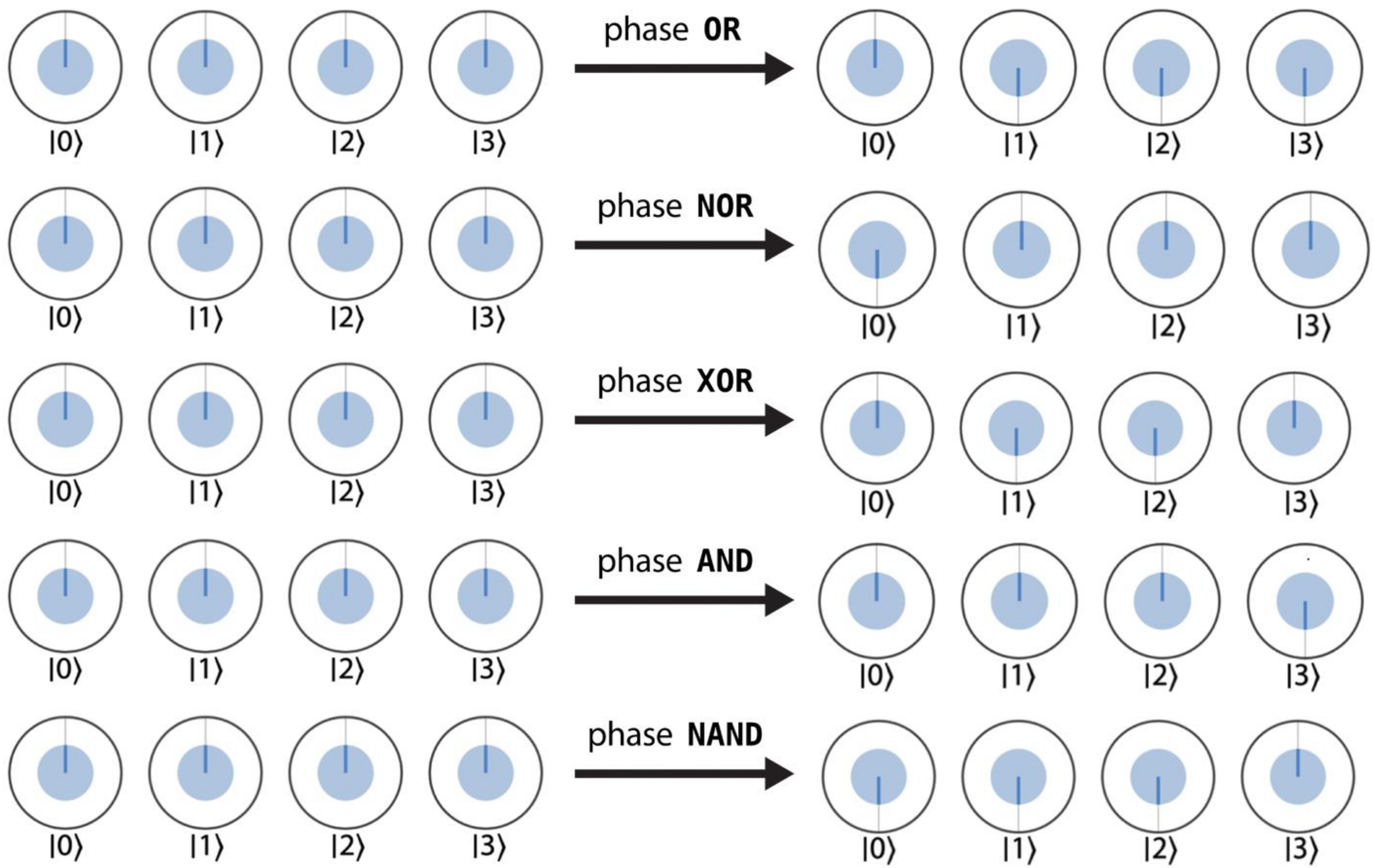
QUANTUM APPLICATIONS

QUANTUM SEARCH

Lecture Outline

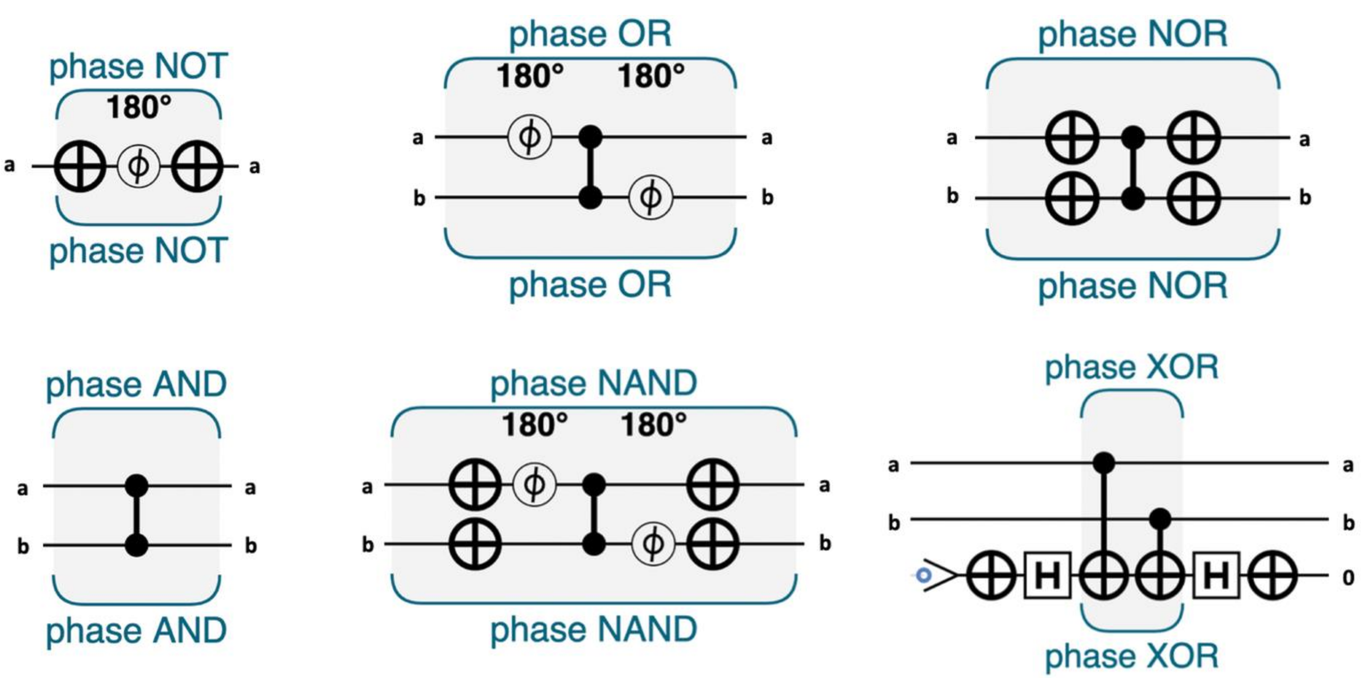
- Phase Logic.
- Logic Puzzles.
  - Kitten and Tigers.
- Boolean Satisfiability (3-SAT).
  - And Unsatisfiability.
- Discussion:
  - Database and online searching.
- Research Discussion:
  - Searching for proofs in proof assistants.

Phase Logic



Phase Logic

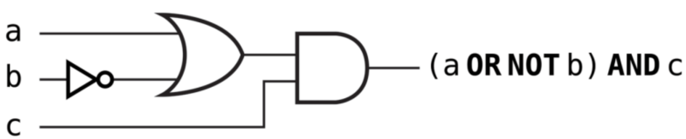
- Fundamental Boolean Logic Ops.

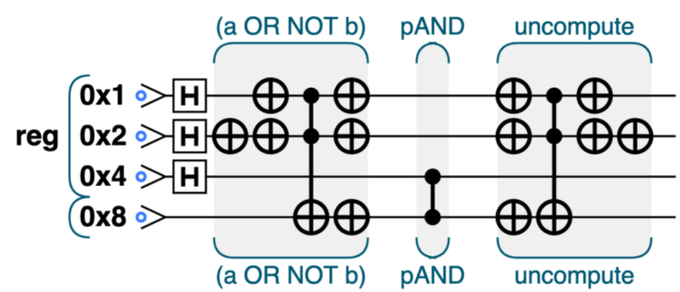


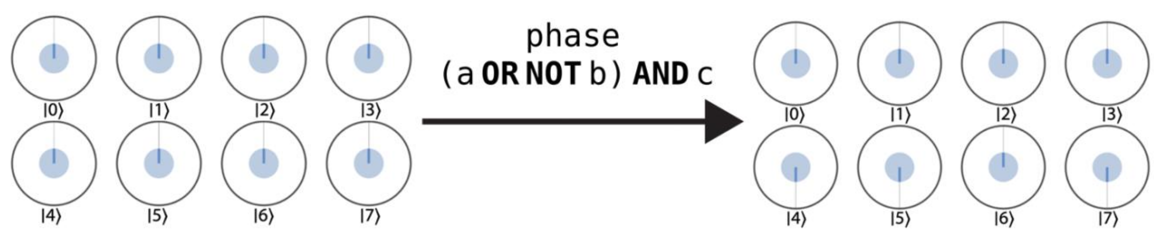
(scratch init. to $|-\rangle$, unentangled)

Quantum Boolean Circuits

- Input to phase logic is encoded in state *values*, but produces output encoded in state *phases*.
  - “Type mismatch.”
- All-but-last op: Use magnitude logic (Ch.5: AL).
  - Last Op: Use phase logic.

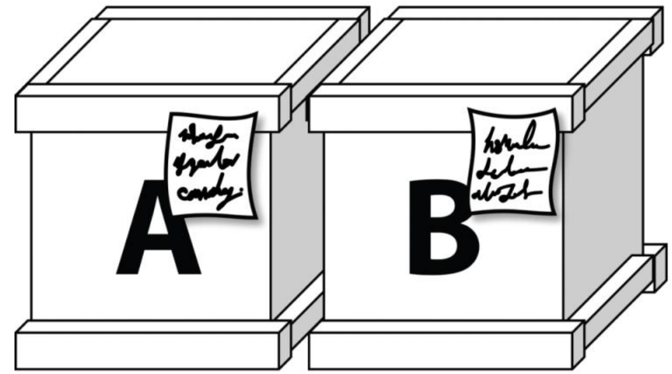


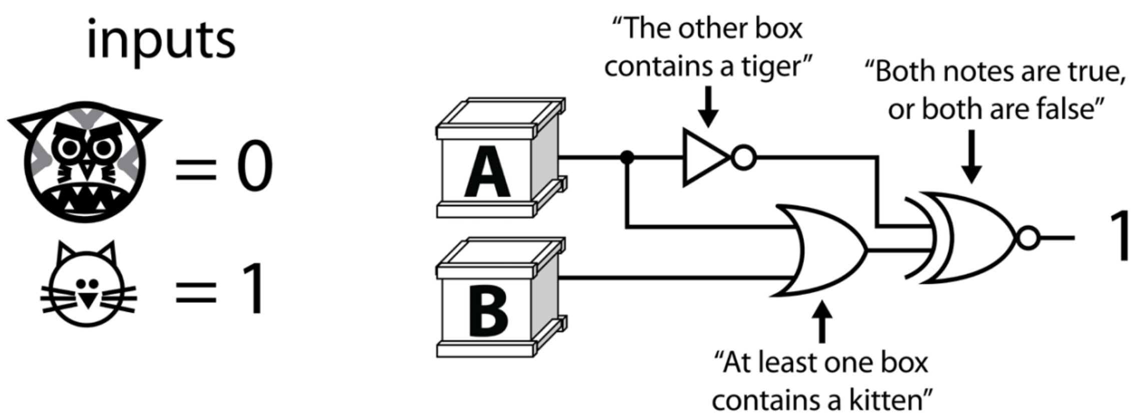




Solving Logic Puzzles

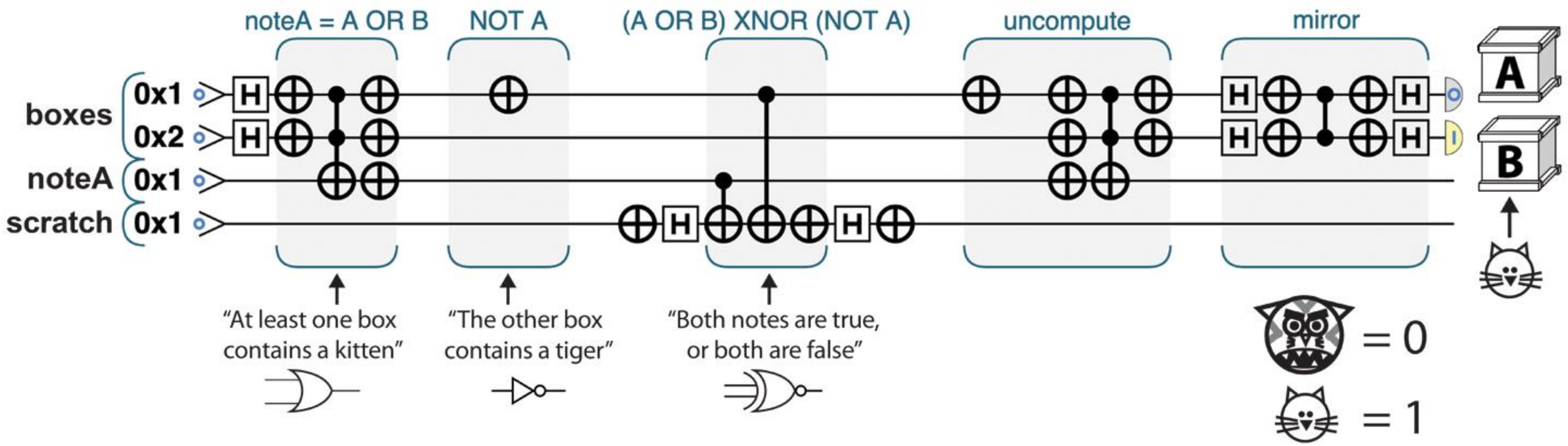
- Of Kittens and Tigers.
  - Open one box only.
  - Note A (OR) and Note B (NOT).
  - Notes are both true, or both false (XOR).





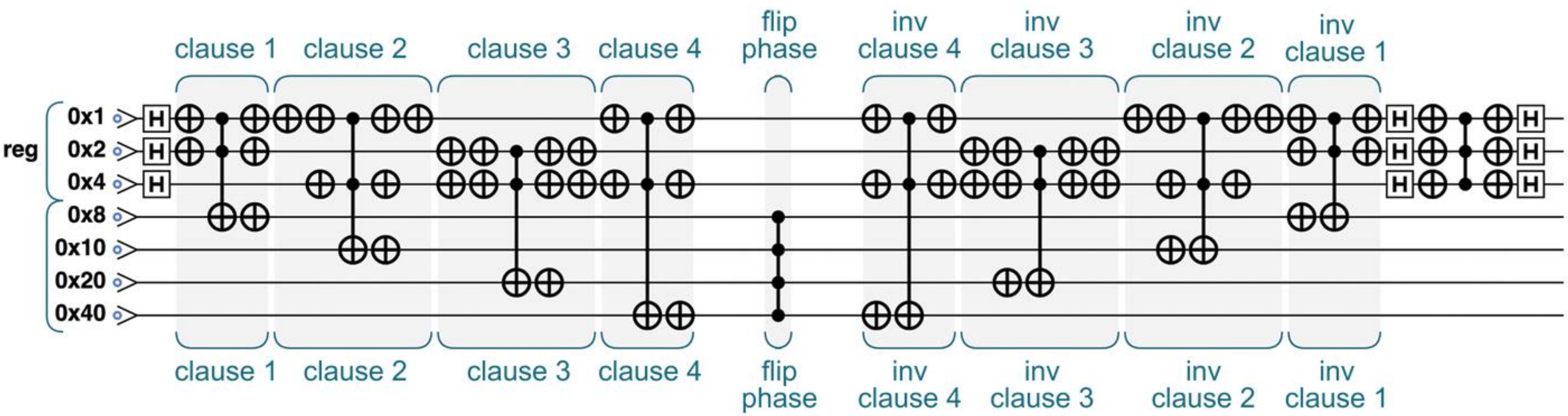
Kittens and Tigers

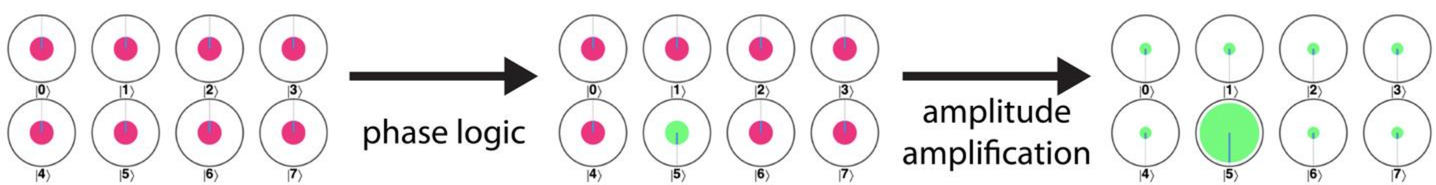
- Run only *once*.



Satisfiability (3-SAT)

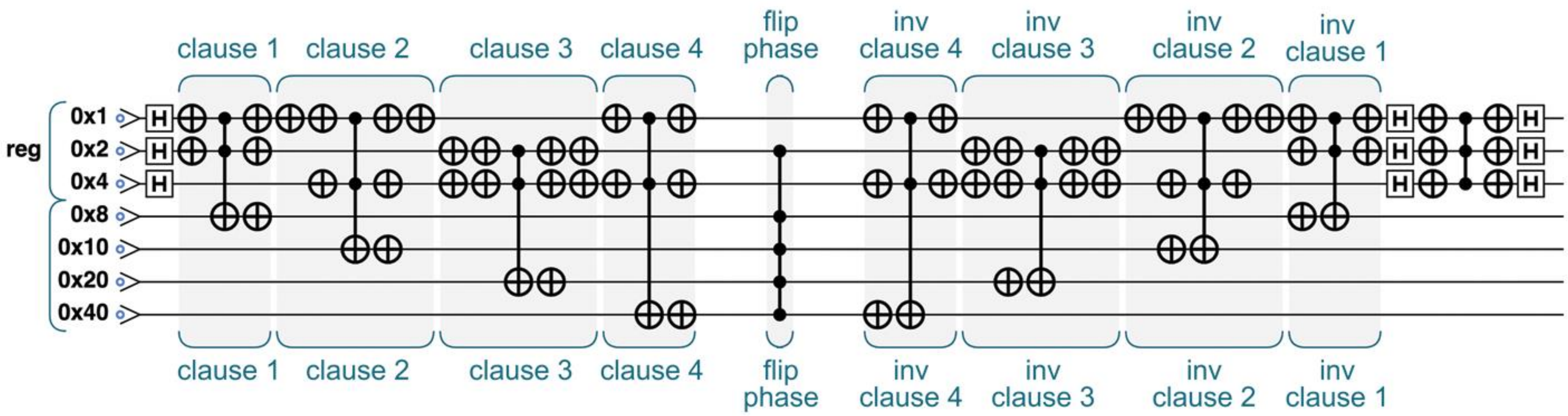
- (a OR b) AND (NOT a OR c) AND (NOT b OR NOT c) AND (a OR c)

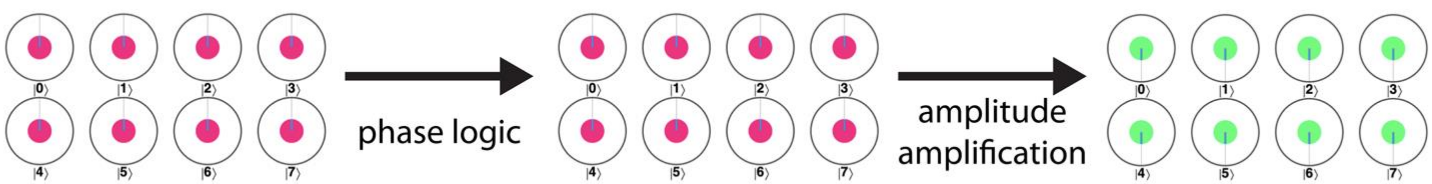




Unsatisfiable 3-SAT

- (a OR b) AND (NOT a OR c) AND (NOT b OR NOT c) AND (a OR c) AND b





- Sol^{n}: Check result *conventionally* (in polynomial time, for NP problems).

Speeding Up Conventional Algorithms

- AA can speed up algorithms with *one-sided error*.
  - If the answer to the problem is “no” the algorithm always outputs “no”.
  - If the answer to the problem is “yes” algorithm outputs the answer “yes” with probability $p > 0$.
  - For QPU speedup, substitute the repeated probabilistic subroutine with an amplitude amplification (AA) step.
  - Useful in finding global minima/maxima (non-convex optimization algorithms).
    - For a function f: $\text{integer} \rightarrow \text{integer}$, finding the index i of a quantum register such that $f(i)$ has the lowest/highest value.

Discussion

"That's NOT what we were seeking, expecting or looking for!!
Where's *the* searching!?? …
Where's what we've been looking for!?"

- Q: How can quantum <u>search</u> get used to implement and speedup customary searches?
  - Offline searches, such as database SQL queries, and file searches (e.g., using file indexing).
  - Online searches, such as Google search.

- A: More quantum data encodings (e.g., for textual data), very loooooong (Boolean, qubit-by-qubit) logical formulas, and, for varying values sought, conditional searching.

Research Discussion: Proof Assistants

- See earlier research discussion in Lect. 5 (QFT slides).

- Coq, Isabelle, and Proof Designer.
  - Coq domain theory formalization (in 2010, 2015).
    - Recent post on [Coq-Club], and quantum searching for proofs.
  - Lean, ACL^2 (ACL2), Little Prover (Scheme/Lisp), … .
    - Functional programming.
  - Proof Designer:
    - For developing elementary set-theoretic proofs. User-friendly.
    - ‘How to Prove It: A Structured Approach’, 3rd Ed., Velleman, 2019.
    - Proof Designer 2, and APM (Android Proof Maker).
      - STDF proposal.

Projects

- Quantum Simulation:
  - Represent the Hamiltonian of some physical or chemical system of your choice on a quantum computer.

- Quantum Search:
  - Offline Search: Given a database, implement a quantum search algorithm for finding elements in the database that satisfy a variable (non-constant) input condition.
  - Online Search: Implement a quantum search algorithm for finding the most relevant webpages to a variable set of search keywords.

- More graduation project ideas after finishing Ch.13.

Next Lecture Appetizer!

- In next lecture (isA):
  - Quantum Graphics.
    - Quantum Supersampling (QSS).
  - Quantum Cryptography.
    - Shor’s Algorithm.

Course Webpage

http://eng.staff.alexu.edu.eg/staff/moez/teaching/pqc-f19

- Where you can:
  - Download lecture slides (incl. exercises and homework).
  - Check links to other useful material.

Thank You