GHZ

Entanglement - Beyond Bell Pairs

- Entanglement Reminder: We've seen the Bell pair and its significance.
- Next Step: Introducing the GHZ (Greenberger-Horne-Zeilinger) state, a 3-qubit entangled state:
- Why GHZ Matters: Used for illustrating key entanglement concepts and has unique applications we'll explore later.

Classical vs. Quantum Correlations

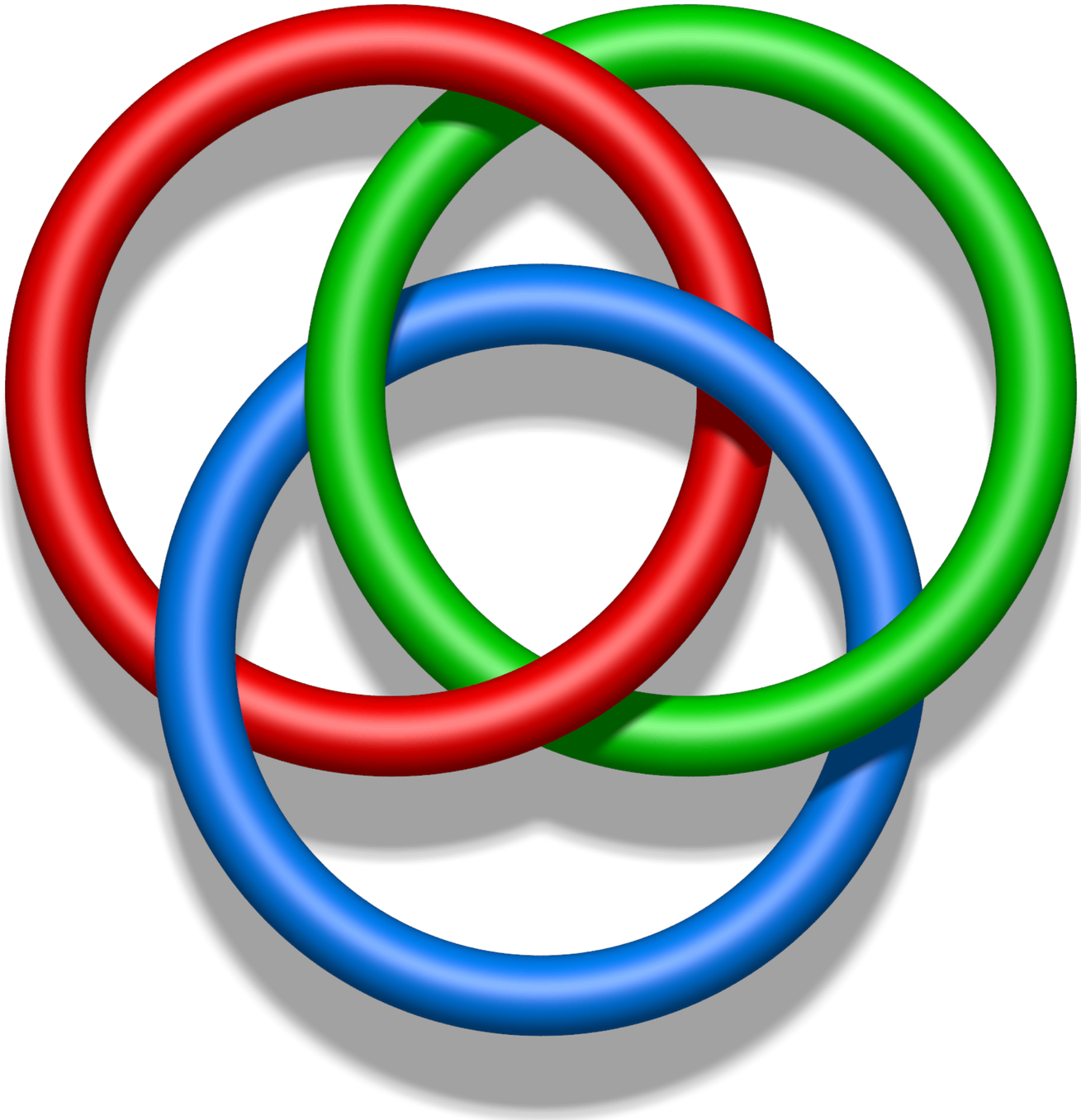
- Classical Scenario: Alice, Bob, and Charlie hold bits that are either all 0s or all 1s, showing classical correlation.
- Quantum Scenario (GHZ State):
  - All three can observe entanglement when together.
  - What if Charlie leaves? Alice and Bob only see classical correlation without knowing if Charlie measured.

The No-Communication Theorem

- If Alice and Bob share an entangled state, nothing Alice chooses to do will have any effect on Bob’s density matrix.
- In other words, there’s no observable effect on Bob’s end. Which is the fundamental reason that quantum mechanics is compatible with the limitations of relativity.
- Key Insight: If Charlie measured his qubit, Alice and Bob wouldn’t detect it.
- Outcome:
  - Without Charlie, Alice and Bob see classical correlations only.
  - This highlights that entanglement among three qubits can "disappear" when one qubit is removed.

Understanding the GHZ State through Density Matrices

- Density Matrix for Alice and Bob: (partial representation of the shared state)
  - Different from the density matrix of a Bell pair.
  - Implication: Shows only classical correlation without all three qubits.
- Principle: If Alice is maximally entangled with Bob, she can’t be maximally entangled with Charlie.
- GHZ State Insight: All three qubits need to be together to observe entanglement.
- Analogy: Similar to Borromean Rings — three rings are linked together, but no two are directly linked.



Collision Problem

The Collision Problem

- Given black-box access to $F\colon \{1,\ldots,N\} \to \{1,\ldots,N\}$.
- Promised that F is two-to-one.
- Task: Find x and y such that $x \neq y$ and $f(x) = f(y)$
- Decision version:
  - Promised that F is either one-to-one or two-to-one
  - Task: decide which.
- If solve the “search” version, then also solve the “decision” version.
- Lower bound for “decision” implies the same lower bound for “search”.

Classical randomized algorithm

- $\sqrt{N}$ queries are necessary and sufficient to solve the collision problem.
- Why?
  - Upper bound: birthday paradox.
  - Lower bound.

Quantumly…

- Use Grover's algorithm to get $\sqrt{N}$.
- Question: Can we combine the two approaches to do better than $\sqrt{N}$?
- Brassard, Hoyer, Tapp 1997
  - Pick $\sqrt[3]{N}$ random inputs, query them classically, and sort the results for fast lookup
  - Run Grover's algorithm on $N^{2/3}$ more random inputs
  - In Grover search, count each input x as "marked" if and only if $f(x) = f(y)$ for one of the $\sqrt[3]{N}$ inputs that was already queried in the first step.

Element Distinctness

Element Distinctness Problem

- Given black-box access to $F\colon \{1,\ldots,N\} \to \{1,\ldots,N\}$, with no promises.
- Task: Determine whether F is one-to-one.
- Classically, hash the elements, which would take N queries.

Quantumly…

- Takes only $N^{3/4}$ queries [Buhrman et al. 2000]
- Partition the values of F into $\sqrt{N}$ blocks of length $\sqrt{N}$ each.
- Doing Grover search through $\sqrt{N}$ blocks, counting each block as “marked” if and only if the “inner algorithm” finds a collision involving that block.

- “Inner algorithm”:
  - Pick a block at random and query all elements in it.
  - If we find a collision in the block, we are done.
  - Otherwise, sort the elements in the block for fast lookup, and Grover search on the other values, counting an item as “marked” if and only if it equals an element from the collision block.
- “Inner algorithm” succeeds as long as we picked a block that contains at least one element of a collision pair.

Parity

The Parity Problem

- Given $x \in \{0,1\}^n$, determine $x_1 \oplus \ldots \oplus x_n$.
- Classically, requires ___ queries.
- Quantumly, we can do it in $n/2$ queries.
  - Think about the case where $n = 2$…
- Fact 1: Quantum query complexity for the parity problem is at least $n/2$.
- Fact 2: Given $x \in \{0,1\}^n$, determine whether it has more 0's or 1's. Quantum query complexity for the majority problem is at least n.

Quantum Complexity Theory

Classical Complexity Theory

- P (Polynomial-Time)
  - Decision problems solvable in polynomial time on a deterministic digital computer.
  - Examples: Linear programming, connectivity of graphs.
- NP (Nondeterministic Polynomial-Time)
  - Problems with a deterministic polynomial-time verification for “yes” answers.
  - Example: Factoring, as a yes-or-no decision problem.
- NP-hard
  - Problems to which every NP problem can be reduced in polynomial time.
  - Solving any NP-hard problem in polynomial time would solve all NP problems in polynomial time.
- NP-complete
  - Problems that are both in NP and NP-hard, the “hardest” in NP.
  - Examples: Traveling Salesman, 3SAT, Max Clique, Sudoku, and more.

Quantum Complexity Class BQP

- BQP (Bounded-Error Quantum Polynomial-Time)
  - Defined by Bernstein and Vazirani (1993) as a quantum generalization of P.
  - Contains decision problems solvable in polynomial time on a quantum computer.
- Relation to Classical Classes:
  - $P \subseteq BQP$
    - Quantum computers can simulate classical ones (e.g., via Toffoli gates for AND, OR, NOT).
  - Factoring in BQP
    - Shor's algorithm shows Factoring is in BQP, but it's unknown if Factoring is in P.
  - $NP \subseteq BQP$?
    - We don't know if quantum computers can solve all NP problems (including NP-complete) in polynomial time.
  - BBBV Theorem: No easy proof that $NP \subseteq BQP$ exists by treating NP problems as a "black box."

Big Open Questions in Quantum Complexity

- Can Quantum Computers Solve NP-Complete Problems in Polynomial Time?
- Is BQP $\subseteq$ NP?
  - For every quantum-solvable problem, can we find a short, classically verifiable proof?
- Is BQP Contained in Another Classical Class?
  - Bernstein and Vazirani showed BQP $\subseteq$ PSPACE
  - (PSPACE: Problems solvable with polynomial memory but possibly exponential time).

The Challenge of Proving P ≠ BQP

- Proving P ≠ BQP would imply P ≠ PSPACE, a major unsolved problem in itself.
- Hence, proving P ≠ BQP might be as challenging as proving P ≠ NP.