Computationally Hard Problems in Translationally-Invariant Spin Systems

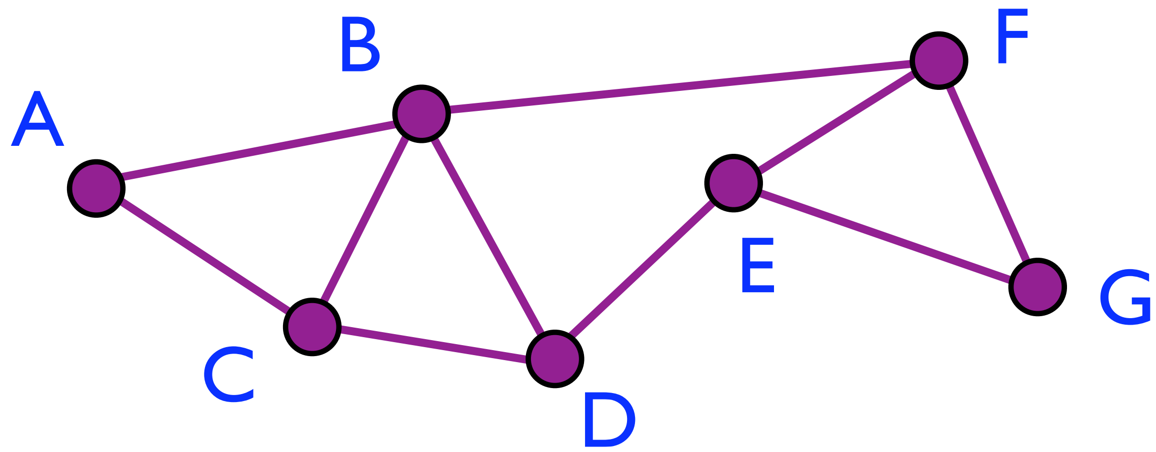
Daniel Gottesman
Perimeter Institute

Work w/ Sandy Irani

arXiv:0905.2419 [quant-ph], Proc. FOCS 2009

Computing Properties of Spin Systems

A spin system consists of particles with an internal Hilbert space which interact via a Hamiltonian, usually via terms acting on a limited number of particles at once.



$$H = X_A Z_B + Z_A Z_C + Y_B Z_C + X_B X_D + Y_C Z_D + Z_B X_F + Y_D Z_E + X_E X_F + Z_F X_G + Z_E Z_G$$

We would like to be able to learn about spin systems. E.g., we would like to be able to find the spectrum, or even just the ground state energy, of a Hamiltonian H.

Can we do this with reasonable expenditure of computational resources?

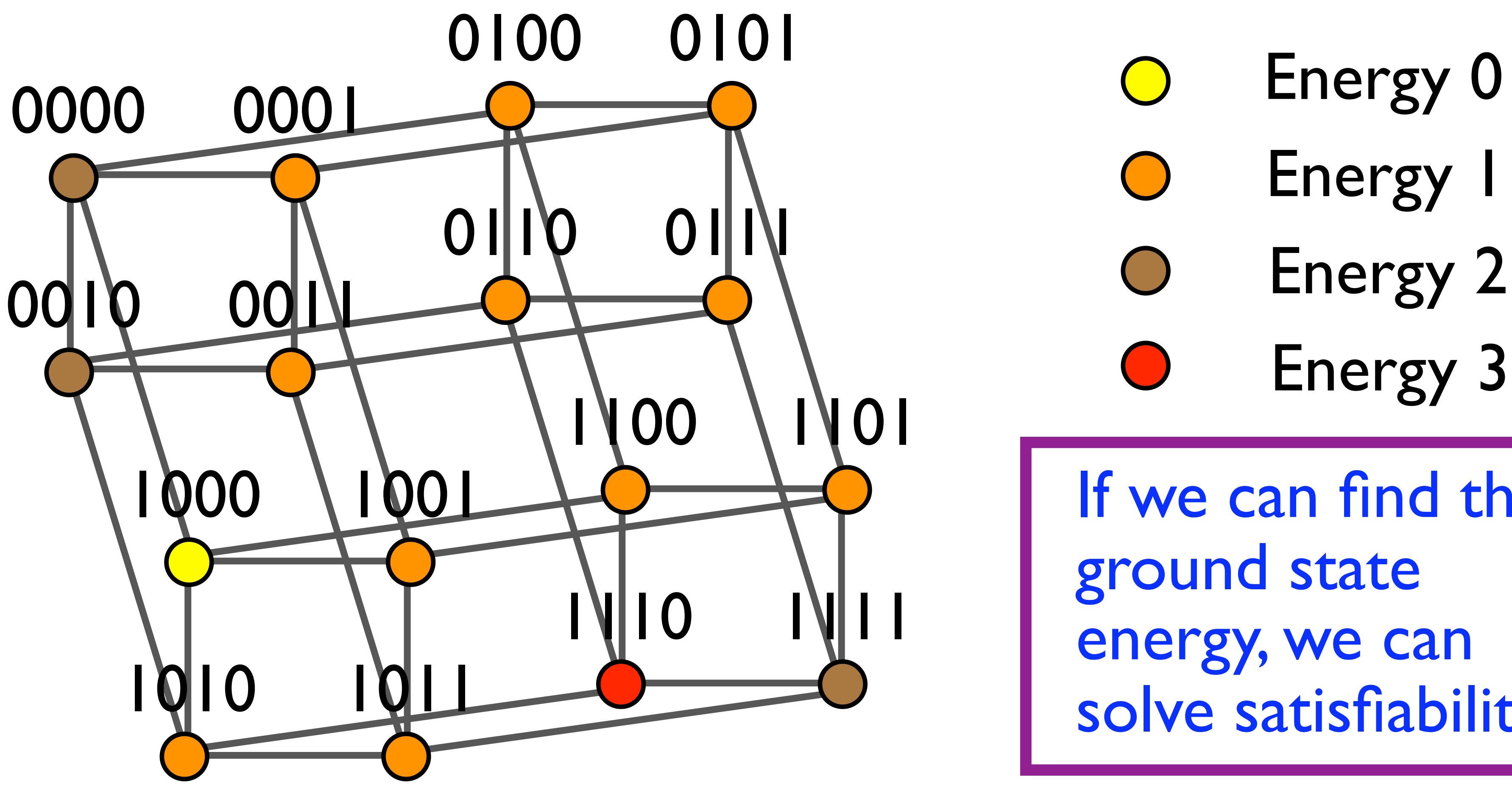
Note that it is not automatic that we can find the ground state energy efficiently even on a quantum computer: Some systems - spin glasses - take a very long time to relax to a thermal state.

Satisfiability as a Spin System

We can imagine encoding Boolean satisfiability - a hard problem - into a spin system, with energy = # satisfied clauses.

(not A or not B or not C) and (A or D) and (B or C or not D) and (A or B or D) and (not B or C or not D) and (not C or not D) and (not A or not C or D) and (not A or not B or D)

$$H = ABC + (1-A)(1-D) + (1-B)(1-C)D + (1-A)(1-B)(1-D) + CD + B(1-C)D + AC(1-D) + AB(1-D)$$



If we can find the ground state energy, we can solve satisfiability!

Who Cares if Spin Systems are Hard?

We consider that a problem can be efficiently solved if it takes time polynomial in the input size. Satisfiability probably has no efficient solution (unless P=NP). Thus, spin systems can be hard.

- Hardness results serve as no-go theorems for efficient simulation.
- When it is hard to find the ground state, the system cannot relax efficiently - a spin-glass-like property. Spin glasses are poorly understood, and perhaps rigorous computational hardness results can help shed some light on them.
- Hardness results can be related to constructions to implement quantum computation in spin systems, e.g. adiabatic and quantum walk-based quantum computation. The techniques can be useful for designing other interesting spin systems.
- Hardness results in spin systems can serve as a jumping-off point for hardness results about other computational problems, and in general they can help us understand quantum complexity classes.

Complexity Classes and NP

Definition: A language L is a set of bit strings of arbitrary length. An instance x is a bit string which we consider as example of L; a yes instance corresponds to $x \in L$, while a no instance corresponds to $x \notin L$. Sometimes we are concerned about languages which involve a promise, which means we do not care about arbitrary bit instances, only ones with some property. A complexity class is a set of languages.

Definition: A language L is in NP if there exists a polynomial-time algorithm f(x,w), and for each $x \in L$, there exists a "witness" w_x such that $f(x,w_x)=1$, while if $x \notin L$, $f(x,w) = 0\ \forall w$.

Some problems, such as satisfiability or the Traveling Salesman problem, are as difficult as any problem in NP. We formalize this through the notion of reduction and completeness.

Definition: A language M reduces to a language L if there exists a polynomial-time algorithm f(x) such that $f(x) \in L$ iff $x \in M$. A language L is complete for a complexity class C if $L \in C$ and any language $M \in C$ can be reduced to L.

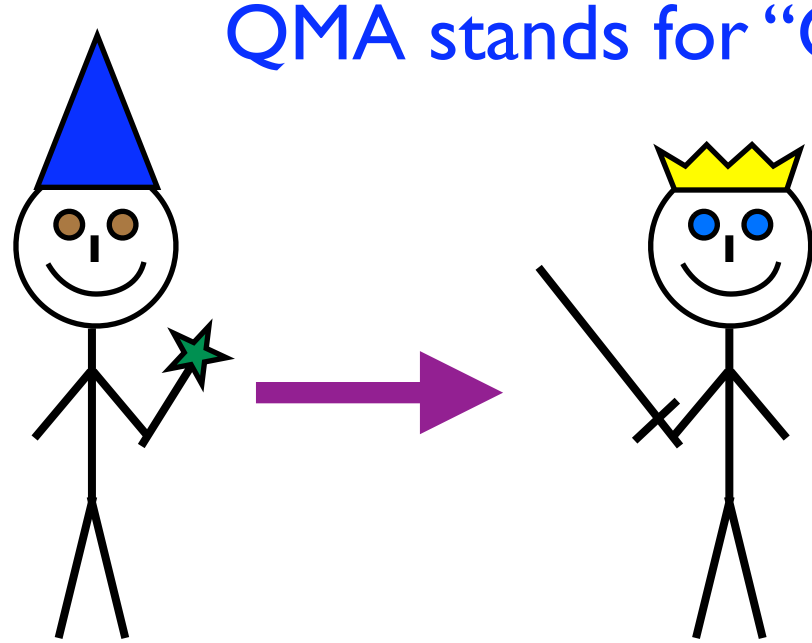
QMA

NP is a classical complexity class composed of problems which may be hard to solve but are easy to check on a classical computer. QMA is the best quantum analogue, a complexity class composed of problems which may be hard to solve, but are easy to check on a quantum computer.

Definition: A language L is in QMA if there exists a polynomial-time *quantum* algorithm $f(x, |\psi\rangle)$ and for each $x \in L$ there exists a *quantum* witness $|\psi_x\rangle$ such that $f(x, |\psi_x\rangle)=1$ with probability at least 2/3, while if $x \notin L$, $f(x, |\psi\rangle) = 0$ with probability at least 2/3 $\forall\ |\psi\rangle$. We are given a promise that one of these two cases is true for x.

QMA stands for "Quantum Merlin-Arthur"

Merlin is very powerful, but not very trustworthy.



He wishes to prove something to Arthur, who has limited computational power.

k-Local Hamiltonians

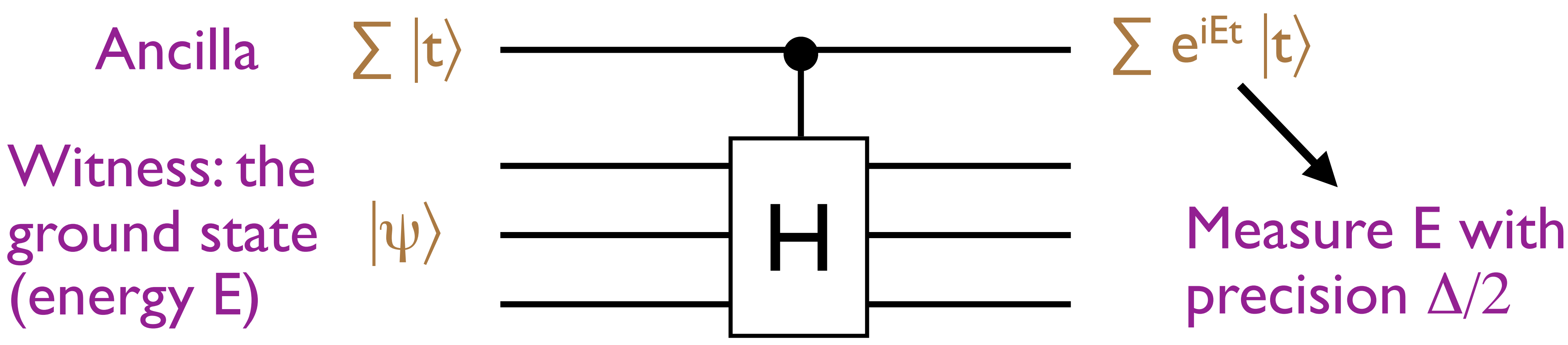
Definition: A Hamiltonian is k-local if it is the sum of terms that act on at most k qubits (or more generally, k particles) at a time.

E.g., $H = X_1X_2X_3 + Z_1Z_4 + X_3X_4X_5$ is 3-local.

Note that "local" does not imply anything about geometry.

Definition: The language k-LOCAL HAMILTONIAN consists of triplets (H, E, Δ). H is a k-local Hamiltonian, and we have the promise that the ground state energy is either at most E (the "yes" instances) or above $E+\Delta$ (the "no" instances).

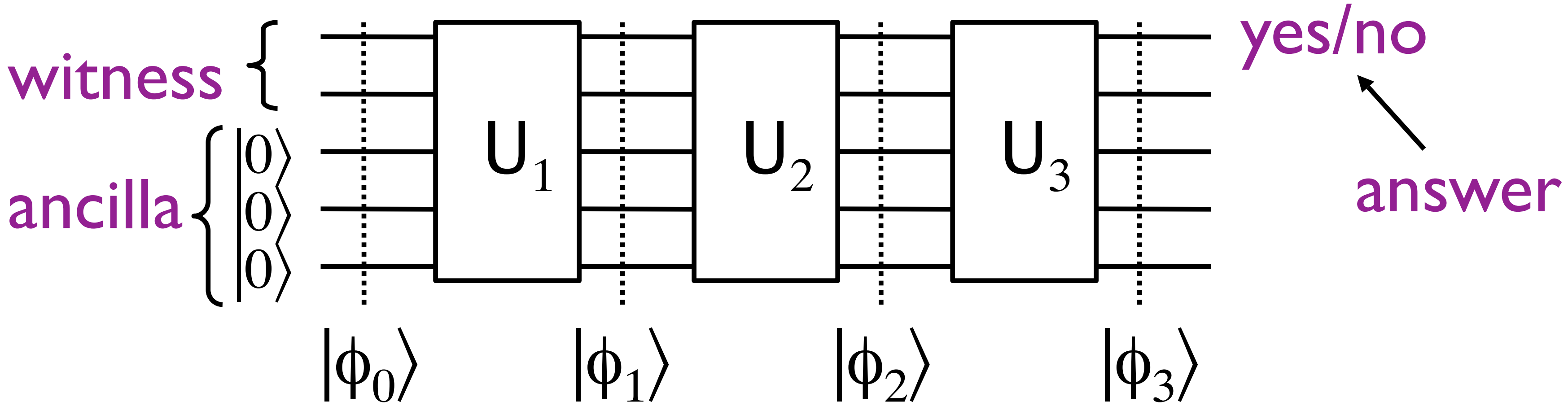
Theorem: k-LOCAL HAMILTONIAN is in QMA.



5-Local Hamiltonians

Theorem: (Kitaev 2002) 5-LOCAL HAMILTONIAN is QMA-complete.

We wish to take an instance x of an arbitrary QMA language and find a Hamiltonian H such that the ground state of H is ~0 if x is a "yes" instance and is ~Δ if x is a "no" instance.



Checking circuit for the instance x

We will write down a Hamiltonian whose ground state corresponds to the "history" of the checking circuit.

The History State

The history state:

$$|\psi\rangle = \sum |\phi_t\rangle |t\rangle$$

Circuit's state at time t → $|\phi_t\rangle$; clock → $|t\rangle$

The clock is encoded in unary: e.g., $|1110000\rangle$ is time 3.

We need 3 types of terms in our Hamiltonian:

- Terms to initialize $|\phi_0\rangle$ correctly: $|1\rangle\langle 1|_{ancilla} \otimes |0\rangle\langle 0|_{clock}$
- Terms to check the output qubit: $|0\rangle\langle 0|_{answer} \otimes |T\rangle\langle T|_{clock}$
  (T is the length of the checking circuit.)
- Terms to ensure each time step in the circuit is correct:
  $\left(|a\rangle \otimes |t\rangle - U_t |a\rangle \otimes |t+1\rangle \right) \left(\langle a| \otimes \langle t| - \langle a| (U_t)^\dagger \otimes \langle t+1| \right)$

In the last type of term, we write $|t\rangle = |100\rangle$ (with the 1 in the t-th position in the clock), and $|t+1\rangle = |110\rangle$. U can be a 2-qubit gate.

The third type of term generates a walk Hamiltonian, equivalent to a particle hopping on a line, which has gap $\sim 1/T^2$. (This sets Δ.)

Hard Vs. Easy Quantum Spin Systems

Some Hamiltonians are thus hard to simulate, even on a quantum computer, but of course some systems we can simulate efficiently on a classical computer. What distinguishes easy and hard?

5-Local Hamiltonians	Hard (QMA-complete)
Specific 1D Systems	Easy (P)

Hard Vs. Easy Quantum Spin Systems

Some Hamiltonians are thus hard to simulate, even on a quantum computer, but of course some systems we can simulate efficiently on a classical computer. What distinguishes easy and hard?

5-Local Hamiltonians	Hard (QMA-complete)
2-Local Hamiltonians	
Specific 1D Systems	Easy (P)

Hard Vs. Easy Quantum Spin Systems

Some Hamiltonians are thus hard to simulate, even on a quantum computer, but of course some systems we can simulate efficiently on a classical computer. What distinguishes easy and hard?

5-Local Hamiltonians	Hard (QMA-complete)
2-Local Hamiltonians	Hard (QMA-complete)
Specific 1D Systems	Easy (P)

Hard Vs. Easy Quantum Spin Systems

Some Hamiltonians are thus hard to simulate, even on a quantum computer, but of course some systems we can simulate efficiently on a classical computer. What distinguishes easy and hard?

5-Local Hamiltonians	Hard (QMA-complete)
2-Local Hamiltonians	Hard (QMA-complete)
Hamiltonians in 2D	
Specific 1D Systems	Easy (P)

Hard Vs. Easy Quantum Spin Systems

Some Hamiltonians are thus hard to simulate, even on a quantum computer, but of course some systems we can simulate efficiently on a classical computer. What distinguishes easy and hard?

5-Local Hamiltonians	Hard (QMA-complete)
2-Local Hamiltonians	Hard (QMA-complete)
Hamiltonians in 2D	Hard (QMA-complete)
Specific 1D Systems	Easy (P)

Hard Vs. Easy Quantum Spin Systems

Some Hamiltonians are thus hard to simulate, even on a quantum computer, but of course some systems we can simulate efficiently on a classical computer. What distinguishes easy and hard?

5-Local Hamiltonians	Hard (QMA-complete)
2-Local Hamiltonians	Hard (QMA-complete)
Hamiltonians in 2D	Hard (QMA-complete)
Hamiltonians in 1D	
Specific 1D Systems	Easy (P)

Hard Vs. Easy Quantum Spin Systems

Some Hamiltonians are thus hard to simulate, even on a quantum computer, but of course some systems we can simulate efficiently on a classical computer. What distinguishes easy and hard?

5-Local Hamiltonians	Hard (QMA-complete)
2-Local Hamiltonians	Hard (QMA-complete)
Hamiltonians in 2D	Hard (QMA-complete)
Hamiltonians in 1D	Hard (QMA-complete)
Specific 1D Systems	Easy (P)

Hard Vs. Easy Quantum Spin Systems

Some Hamiltonians are thus hard to simulate, even on a quantum computer, but of course some systems we can simulate efficiently on a classical computer. What distinguishes easy and hard?

5-Local Hamiltonians	Hard (QMA-complete)
2-Local Hamiltonians	Hard (QMA-complete)
Hamiltonians in 2D	Hard (QMA-complete)
Hamiltonians in 1D	Hard (QMA-complete)
Hamiltonians in 1D with qubits	
Specific 1D Systems	Easy (P)

Hard Vs. Easy Quantum Spin Systems

Some Hamiltonians are thus hard to simulate, even on a quantum computer, but of course some systems we can simulate efficiently on a classical computer. What distinguishes easy and hard?

5-Local Hamiltonians	Hard (QMA-complete)
2-Local Hamiltonians	Hard (QMA-complete)
Hamiltonians in 2D	Hard (QMA-complete)
Hamiltonians in 1D	Hard (QMA-complete)
Hamiltonians in 1D with qubits	?
Specific 1D Systems	Easy (P)

Hard Vs. Easy Quantum Spin Systems

Some Hamiltonians are thus hard to simulate, even on a quantum computer, but of course some systems we can simulate efficiently on a classical computer. What distinguishes easy and hard?

5-Local Hamiltonians	Hard (QMA-complete)
2-Local Hamiltonians	Hard (QMA-complete)
Hamiltonians in 2D	Hard (QMA-complete)
Hamiltonians in 1D	Hard (QMA-complete)
Hamiltonians in 1D with qubits	?
Translationally-Invariant Ham.	
Specific 1D Systems	Easy (P)

Hard Vs. Easy Quantum Spin Systems

Some Hamiltonians are thus hard to simulate, even on a quantum computer, but of course some systems we can simulate efficiently on a classical computer. What distinguishes easy and hard?

5-Local Hamiltonians	Hard (QMA-complete)
2-Local Hamiltonians	Hard (QMA-complete)
Hamiltonians in 2D	Hard (QMA-complete)
Hamiltonians in 1D	Hard (QMA-complete)
Hamiltonians in 1D with qubits	?
Translationally-Invariant Ham.	Hard (QMA_{EXP}-complete)
Specific 1D Systems	Easy (P)

Hard Vs. Easy Quantum Spin Systems

Some Hamiltonians are thus hard to simulate, even on a quantum computer, but of course some systems we can simulate efficiently on a classical computer. What distinguishes easy and hard?

5-Local Hamiltonians	Hard (QMA-complete)
2-Local Hamiltonians	Hard (QMA-complete)
Hamiltonians in 2D	Hard (QMA-complete)
Hamiltonians in 1D	Hard (QMA-complete)
Hamiltonians in 1D with qubits	?
Translationally-Invariant Ham.	Hard (QMA_{EXP}-complete)
Trans. & Reflection-Inv. Ham.	Hard (QMA_{EXP}-complete)
Specific 1D Systems	Easy (P)

Translation Invariance

Theorem: (Gottesman, Irani, 2009) 2-LOCAL HAMILTONIAN is QMA_{EXP}-complete when the Hamiltonians involve *translation-invariant* nearest-neighbor interactions in 1D. The input is N (the number of spins) written in binary.

QMA_{EXP} is like QMA, but with exponential size (in the input) witness and verification circuit. However, since in this case, the input is log N in length, this just means the problem probably cannot be solved in a time less than exp(N).

Proof Idea:

Use a 1D translationally-invariant quantum system to simulate a quantum Turing machine which first counts N, the number of particles available, and writes it in binary on one end of the computer. Then it simulates a universal quantum Turing machine on the input N.

QMA vs. QMA_{EXP}

QMA is the class of problems which can be checked in polynomial time on a quantum computer, given a polynomial-sized witness.

QMA_{EXP} is the class of problems which can be checked in exponential time on a quantum computer, given an exponential-size quantum witness.

"Polynomial" and "exponential" are given as functions of the size of the input.

We believe that QMA-hard problems will take more than poly time to solve on a quantum computer and that QMA_{EXP}-hard problems will take more than exp time.

- In the non-translationally-invariant 1D Hamiltonian problem, the input size is poly(N), and the problem is QMA-complete, so the difficulty is likely exp(N).
- In the translationally-invariant 1D Hamiltonian problem, the input size is log(N), and the problem is QMA_{EXP}-complete, so the difficulty is likely exp(N) again.

Overview of the Construction

We wish to create transition rules that cause the 1D system to perform the following steps. Then use the usual tricks to make a Hamiltonian whose ground state is a history state for this machine.

1. Initialization: Check the set-up configuration at all sites.
2. Counting: Simulate a Turing machine (TM) to count N.
3. Computation: Simulate a universal quantum Turing machine.

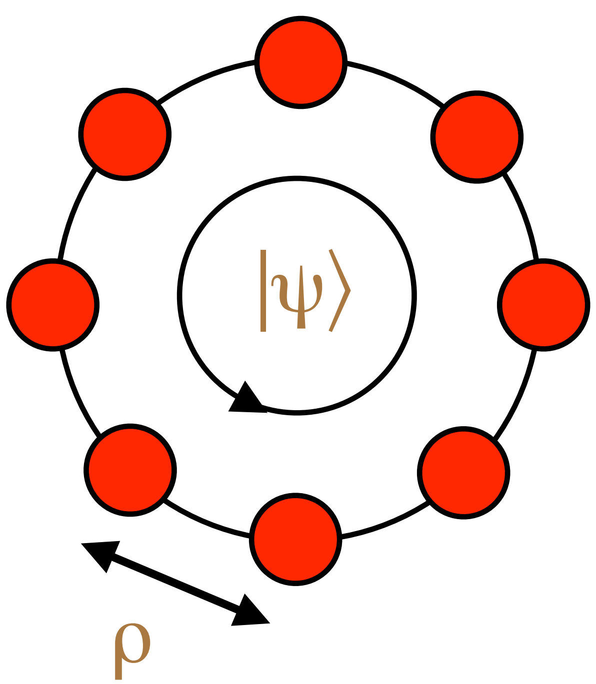
We break up the system into six tracks. Each site has a state for each of the tracks.

1. Second Hand: Sweeps left and right. Each cycle causes one step of a Turing machine. (The first cycle does initialization.)
2. Minute Hand: Sweeps left and right. Counts the steps of each Turing machine. Advances one step for each second hand sweep.
3. Work Tape: Both Turing machines use this track to write on. After the counting phase, the work tape contains a function of N, which is used as the input for the computation phase.
4. Counting Head: Contains the counting TM's head (phase 2).
5. Computation Head: Contains the universal TM head (phase 3).
6. Witness: Contains the witness for phase 3.

N-Representability

We can use the hardness of translation-invariant Hamiltonians to show other computational problems are hard.

N-Representability: Given a 2-particle density matrix ρ, determine whether there exists an N-particle symmetric pure state $|\psi\rangle$ on the circle such that $\mathrm{Tr}_{N-2} |\psi\rangle\langle\psi| = \rho$.



Suppose we can solve N-Representability up to some fixed accuracy ε. Then we can solve Trans-Inv. Hamiltonian on a circle: try a dense network of possible ρs (within ε of every state), test if each can be extended to a global state and calculate its energy. We determine the minimum and that gives us the minimum energy of H.

N-Representability is QMA_{EXP}-complete.

Hardness of Classical Spin Systems

What about classical spin systems? (I.e., systems where all Hamiltonian terms are ZZ-type terms, in the standard basis.)

3-Satisfiability (3-Local Ham.) Hard (NP-complete)

2-Satisfiability (w/ bits) Easy (P)

Hardness of Classical Spin Systems

What about classical spin systems? (I.e., systems where all Hamiltonian terms are ZZ-type terms, in the standard basis.)

3-Satisfiability (3-Local Ham.)	Hard (NP-complete)
Max-2-SAT (2-Local Ham.)	Hard (NP-complete)
2-Satisfiability (w/ bits)	Easy (P)

Hardness of Classical Spin Systems

What about classical spin systems? (I.e., systems where all Hamiltonian terms are ZZ-type terms, in the standard basis.)

3-Satisfiability (3-Local Ham.)	Hard (NP-complete)
Max-2-SAT (2-Local Ham.)	Hard (NP-complete)
Max-2-SAT in 2D (Tiling)	
2-Satisfiability (w/ bits)	Easy (P)

Hardness of Classical Spin Systems

What about classical spin systems? (I.e., systems where all Hamiltonian terms are ZZ-type terms, in the standard basis.)

3-Satisfiability (3-Local Ham.)	Hard (NP-complete)
Max-2-SAT (2-Local Ham.)	Hard (NP-complete)
Max-2-SAT in 2D (Tiling)	Hard (NP-complete)
2-Satisfiability (w/ bits)	Easy (P)

Hardness of Classical Spin Systems

What about classical spin systems? (I.e., systems where all Hamiltonian terms are ZZ-type terms, in the standard basis.)

3-Satisfiability (3-Local Ham.)	Hard (NP-complete)
Max-2-SAT (2-Local Ham.)	Hard (NP-complete)
Max-2-SAT in 2D (Tiling)	Hard (NP-complete)
Max-2-SAT in 1D	
2-Satisfiability (w/ bits)	Easy (P)

Hardness of Classical Spin Systems

What about classical spin systems? (I.e., systems where all Hamiltonian terms are ZZ-type terms, in the standard basis.)

3-Satisfiability (3-Local Ham.)	Hard (NP-complete)
Max-2-SAT (2-Local Ham.)	Hard (NP-complete)
Max-2-SAT in 2D (Tiling)	Hard (NP-complete)
Max-2-SAT in 1D	Easy (P)
2-Satisfiability (w/ bits)	Easy (P)

Hardness of Classical Spin Systems

What about classical spin systems? (I.e., systems where all Hamiltonian terms are ZZ-type terms, in the standard basis.)

3-Satisfiability (3-Local Ham.)	Hard (NP-complete)
Max-2-SAT (2-Local Ham.)	Hard (NP-complete)
Max-2-SAT in 2D (Tiling)	Hard (NP-complete)
Translationally-Invariant Tiling	
Max-2-SAT in 1D	Easy (P)
2-Satisfiability (w/ bits)	Easy (P)

Hardness of Classical Spin Systems

What about classical spin systems? (I.e., systems where all Hamiltonian terms are ZZ-type terms, in the standard basis.)

3-Satisfiability (3-Local Ham.)	Hard (NP-complete)
Max-2-SAT (2-Local Ham.)	Hard (NP-complete)
Max-2-SAT in 2D (Tiling)	Hard (NP-complete)
Translationally-Invariant Tiling	Hard (NEXP-complete)
Max-2-SAT in 1D	Easy (P)
2-Satisfiability (w/ bits)	Easy (P)

Hardness of Classical Spin Systems

What about classical spin systems? (I.e., systems where all Hamiltonian terms are ZZ-type terms, in the standard basis.)

3-Satisfiability (3-Local Ham.)	Hard (NP-complete)
Max-2-SAT (2-Local Ham.)	Hard (NP-complete)
Max-2-SAT in 2D (Tiling)	Hard (NP-complete)
Translationally-Invariant Tiling	Hard (NEXP-complete)
Max-2-SAT in 1D	Easy (P)
Trans. Inv. 1D Max-2-SAT	
2-Satisfiability (w/ bits)	Easy (P)

Hardness of Classical Spin Systems

What about classical spin systems? (I.e., systems where all Hamiltonian terms are ZZ-type terms, in the standard basis.)

3-Satisfiability (3-Local Ham.)	Hard (NP-complete)
Max-2-SAT (2-Local Ham.)	Hard (NP-complete)
Max-2-SAT in 2D (Tiling)	Hard (NP-complete)
Translationally-Invariant Tiling	Hard (NEXP-complete)
Max-2-SAT in 1D	Easy (P)
Trans. Inv. 1D Max-2-SAT	Easy (P, as a function of log N)
2-Satisfiability (w/ bits)	Easy (P)

Other Variations

	2D, Trans. Inv. Only	2D, Reflection	2D, Rotation	1D
4-Corners BC				
Unweighted	NEXP-complete	P‡	P	P
Weighted	NEXP-complete	NEXP-complete	P	P
Open BC				
Unweighted	P†	P	P	P
Weighted	NEXP-complete	NEXP-complete	P	P
Periodic BC				
Unweighted	NEXP-complete*	P‡	P	P
Weighted	NEXP-complete	NEXP-complete+	P	P

Periodic Boundary Conditions: Can we tile a torus of size N?

Weighted TILING: Each possible pair of tiles has a cost (+ or -).

Reflection Symmetry: Rules don't depend on left/right or up/down.

Rotation Symmetry: Rules don't depend on direction at all.

† But with an uncomputable parameter.

* Under a randomized reduction.

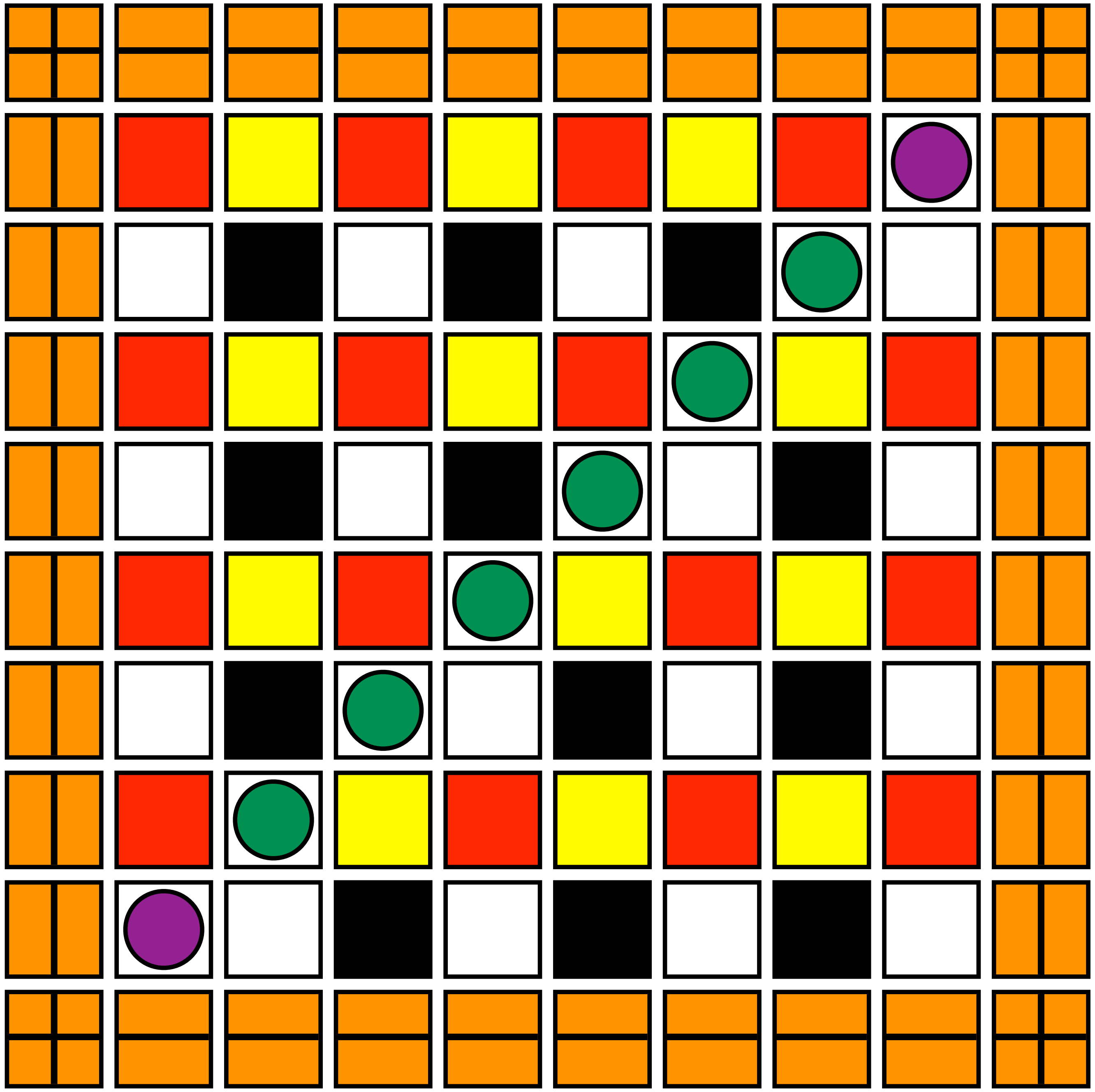
‡ With a possibly uncomputable parameter.

+ P‡ for constant total cost.

Hardness with Reflection Symmetry

By choosing weights appropriately, we can force a layer of classical tiles to arrange themselves like this. We need weights to create a small penalty for the circle tiles; otherwise, there could be multiple ones ruining the pattern.

The point is that in the vicinity of the circle tiles, parity constraints break the reflection symmetry.



We use additional layers of tiles to break the reflection symmetry everywhere. A similar trick works in the quantum case.

Summary

- Finding the ground states of a general translationally-invariant Hamiltonian in 1D is likely to be too hard for a quantum computer.
- Determining if it is possible to tile an N x N grid in 2D with classical tiles is likely to be too hard for a classical computer.
- These spin systems are likely spin glasses with no quenched disorder. The disorder is emergent.
- Note: 1D quantum problem is exp(N) difficulty vs. 1D classical, which is poly(N). But TI 1D quantum is exp(N) vs. TI classical, which is poly(log N).

Some open questions:

- Can we find more natural Hamiltonians which lead to hard problems (say in 2D)?
- What is the border between hard and easy problems? 1D Hamiltonians with qubits?
- Spin systems with constant gaps? (In 1D, is in NP.)