Postulate of Quantum Mechanics - Measurement

Any observable entity (energy, momentum, etc.) corresponds to a Hermitian operator. (Hermitian $\leftrightarrow$ real eigenvalues)
Postulate of Quantum Mechanics - Measurement

Any observable entity (energy, momentum, etc.) corresponds to a [Hermitian] operator. (Hermitian $\leftrightarrow$ real eigenvalues)

$N$-dimensional quantum system:

Measure $\Rightarrow$ outcome must be in $\{\lambda_0, \ldots, \lambda_{N-1}\}$

(Assume for now non-degeneracy: $\lambda_i$’s are distinct and there are $N$ of them)
Postulate of Quantum Mechanics - Measurement

Any observable entity (energy, momentum, etc.) corresponds to a **Hermitian** operator. \((\text{Hermitian} \iff \text{real eigenvalues})\)

\(N\)-dimensional quantum system:

\[
\text{Measure} \implies \text{outcome must be in } \{\lambda_0, \ldots, \lambda_{N-1}\}
\]

(Assume for now non-degeneracy: \(\lambda_i\)'s are distinct and there are \(N\) of them)

After the measurement, system is in a state that is consistent with the outcome.

\[
\begin{align*}
\lambda_0 & \leftrightarrow \ket{v_0} \\
\lambda_1 & \leftrightarrow \ket{v_1} \\
& \quad \vdots \\
\lambda_{N-1} & \leftrightarrow \ket{v_{N-1}}
\end{align*}
\]

\(|v_0\rangle, \ldots, |v_{N-1}\rangle \text{ orthonormal basis.}\)
Postulate of Quantum Mechanics - Measurement

Any observable entity (energy, momentum, etc.) corresponds to a Hermitian operator. (Hermitian $\leftrightarrow$ real eigenvalues)

$N$-dimensional quantum system:

Measure $\Rightarrow$ outcome must be in $\{\lambda_0, \ldots, \lambda_{N-1}\}$

(Assume for now non-degeneracy: $\lambda_i$'s are distinct and there are $N$ of them)

After the measurement, system is in a state that is consistent with the outcome.

$$\begin{align*}
\lambda_0 & \leftrightarrow |v_0\rangle \\
\lambda_1 & \leftrightarrow |v_1\rangle \\
\cdots \\
\lambda_{N-1} & \leftrightarrow |v_{N-1}\rangle
\end{align*}$$

$|v_0\rangle, \ldots, |v_{N-1}\rangle$ orthonormal basis.

Hermitian Operator with:

Eigenvalues: $\lambda_0, \ldots, \lambda_{N-1}$

Eigenvectors: $|v_0\rangle, \ldots, |v_{N-1}\rangle$
Measurement, cont.

State: $|\Phi\rangle$

Measure quantity - operator $A = \sum_i \lambda_i |v_i\rangle \langle v_i|$
Measurement, cont.

State: $|\Phi\rangle$

Measure quantity - operator $A = \sum_i \lambda_i |v_i\rangle\langle v_i|$

$|\Phi\rangle = \alpha_0 |v_0\rangle + \cdots + \alpha_{N-1} |v_{N-1}\rangle$

Probability of outcome $\lambda_i$ is:

$|\alpha_i|^2 = |\langle v_i |\Phi \rangle|^2 = \langle \Phi |v_i \rangle \langle v_i |\Phi \rangle$
Measurement, cont.

State: $|\Phi\rangle$

Measure quantity - operator $A = \sum_i \lambda_i |v_i\rangle\langle v_i|$ 

$|\Phi\rangle = \alpha_0 |v_0\rangle + \cdots + \alpha_{N-1} |v_{N-1}\rangle$

Probability of outcome $\lambda_i$ is:

$$|\alpha_i|^2 = |\langle v_i | \Phi \rangle|^2 = \langle \Phi | v_i \rangle \langle v_i | \Phi \rangle$$

Expected outcome is:

$$\sum_i \text{Prob}[\text{Outcome is } \lambda_i] \cdot \lambda_i = \sum_i \langle \Phi | v_i \rangle \langle v_i | \Phi \rangle \lambda_i$$
Measurement, cont.

State: $|\Phi\rangle$

Measure quantity - operator $A = \sum_i \lambda_i |v_i\rangle\langle v_i|$

$|\Phi\rangle = \alpha_0 |v_0\rangle + \cdots + \alpha_{N-1} |v_{N-1}\rangle$

Probability of outcome $\lambda_i$ is:

$$|\alpha_i|^2 = |\langle v_i | \Phi \rangle|^2 = \langle \Phi | v_i \rangle \langle v_i | \Phi \rangle$$

Expected outcome is:

$$\sum_i \text{Prob}[\text{Outcome is } \lambda_i] \cdot \lambda_i = \sum_i \langle \Phi | v_i \rangle \langle v_i | \Phi \rangle \lambda_i$$

$$= \langle \Phi | \left( \sum_i \lambda_i |v_i\rangle\langle v_i| \right) |\Phi\rangle = \langle \Phi | A | \Phi \rangle$$
Measurement, cont.

State: $|\Phi\rangle$

Measure quantity - operator $A = \sum_i \lambda_i |v_i\rangle\langle v_i|$

$|\Phi\rangle = \alpha_0 |v_0\rangle + \cdots + \alpha_{N-1} |v_{N-1}\rangle$

Probability of outcome $\lambda_i$ is:

$$|\alpha_i|^2 = |\langle v_i|\Phi\rangle|^2 = \langle \Phi|v_i\rangle\langle v_i|\Phi\rangle$$

Expected outcome is:

$$\sum_i \text{Prob[Outcome is } \lambda_i\text{]} \cdot \lambda_i = \sum_i \langle \Phi|v_i\rangle\langle v_i|\Phi\rangle \lambda_i$$

$$= \langle \Phi| \left( \sum_i \lambda_i |v_i\rangle\langle v_i| \right) |\Phi\rangle = \langle \Phi|A|\Phi\rangle$$

$$= \left[ \begin{array}{c} \cdots \cdots \end{array} \right] \left[ \begin{array}{c} A \end{array} \right] \left[ \begin{array}{c} \langle \Phi| \end{array} \right] |\Phi\rangle$$
The Hamiltonian Operator - dynamics

The operator corresponding to energy is called the Hamiltonian, $H$. 
The Hamiltonian Operator - dynamics

The operator corresponding to energy is called the Hamiltonian, $H$.

The time evolution of a closed quantum system is described by Schroedinger’s Equation:

$$i\hbar \frac{d|\psi\rangle}{dt} = H|\psi\rangle.$$
The Hamiltonian Operator - dynamics

The operator corresponding to energy is called the Hamiltonian, $H$.

The time evolution of a closed quantum system is described by Schroedinger’s Equation:

$$i\hbar \frac{d|\psi\rangle}{dt} = H|\psi\rangle.$$  

Simulating the dynamics of quantum systems over time

$$|\psi(0)\rangle \rightarrow i\hbar \frac{d|\psi\rangle}{dt} = H|\psi\rangle \rightarrow |\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle$$
The Hamiltonian Operator - dynamics

The operator corresponding to energy is called the Hamiltonian, $H$.

The time evolution of a closed quantum system is described by Schroedinger’s Equation:

$$i\hbar \frac{d|\psi\rangle}{dt} = H|\psi\rangle.$$  

Simulating the dynamics of quantum systems over time

$$|\psi(0)\rangle \rightarrow i\hbar \frac{d|\psi\rangle}{dt} = H|\psi\rangle \rightarrow |\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle.$$
The Hamiltonian Operator - equilibrium

If a system $S$ interacts with its environment, $S$ will eventually reach an equilibrium state, called the *Gibbs state*.

The Gibbs state is also determined by Hamiltonian $H$.

$$H = \sum_i E_i |v_i\rangle \langle v_i|$$
The Hamiltonian Operator - equilibrium

If a system S interacts with its environment, S will eventually reach an equilibrium state, called the Gibbs state.

The Gibbs state is also determined by Hamiltonian H.

\[ H = \sum_i E_i |v_i\rangle \langle v_i| \]

\[ \rho_{eq} = \sum_i \frac{e^{-\beta E_i}}{Z} |v_i\rangle \langle v_i| \quad \text{where} \quad Z = \sum_i e^{-\beta E_i} \]

Parameter \( \beta \) scales inversely with temperature

Z is called the partition function
The Hamiltonian Operator - equilibrium

If a system $S$ interacts with its environment, $S$ will eventually reach an equilibrium state, called the *Gibbs state*.

The Gibbs state is also determined by Hamiltonian $H$.

$$ H = \sum_i E_i |v_i\rangle \langle v_i| $$

$$ \rho_{eq} = \sum_i \frac{e^{-\beta E_i}}{Z} |v_i\rangle \langle v_i| \quad \text{where} \quad Z = \sum_i e^{-\beta E_i} $$

Parameter $\beta$ scales inversely with temperature

$Z$ is called the *partition function*

$$ \rho_{eq} = \frac{e^{-\beta H}}{Z} \quad \text{where} \quad Z = \text{Tr} \left( e^{-\beta H} \right) $$

[Linden, Popescu, Short, Winter arXiv:0812.2385]
The Hamiltonian Operator - the ground state

As the temperature goes to 0,

the Gibbs state reaches the ground state.

\[
\lim_{\beta \to \infty} \rho_{eq} = \lim_{\beta \to \infty} \sum_i \frac{e^{-\beta E_i}}{Z} |v_i \rangle \langle v_i| = |v_0 \rangle \langle v_0|
\]

(assuming a unique ground state)
The Hamiltonian Operator - the ground state

As the temperature goes to 0, the Gibbs state reaches the ground state.

\[
\lim_{\beta \to \infty} \rho_{eq} = \lim_{\beta \to \infty} \sum_i \frac{e^{-\beta E_i}}{Z} |v_i \rangle \langle v_i| = |v_0 \rangle \langle v_0| 
\]

(assuming a unique ground state)

Given a Hamiltonian \( H \) for a quantum system \( S \):

- Compute the ground energy \( E_0 \) (lowest eigenvalue of \( H \))
- Compute some property of the ground state \( |v_0 \rangle \)
An Example of a Quantum System and Its Hamiltonian

The "state" is the position of the electron relative to the proton:

$$\psi(r, \theta, \phi)$$
An Example of a Quantum System and Its Hamiltonian

Hydrogen Atom

The "state" is the position of the electron relative to the proton:

\[ \psi(r, \theta, \phi) \]

The Hamiltonian describes the energy as a function of the electron location:

\[ \hat{H} = -\frac{\hbar^2}{2m_e} \Delta^2 - \frac{e^2}{4\pi\epsilon_0 r} \]

\[ \Delta^2 = \frac{1}{r^2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left( \sin \theta \frac{\partial}{\partial r} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial}{\partial \phi^2} \]
An Example of a Quantum System and Its Hamiltonian

The "state" is the position of the electron relative to the proton:

$$\psi(r, \theta, \phi)$$

The Hamiltonian describes the energy as a function of the electron location:

$$\hat{H} = -\frac{h^2}{2m_e} \Delta^2 - \frac{e^2}{4\pi\epsilon_0 r}$$

$$\Delta^2 = \frac{1}{r^2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left( \sin \theta \frac{\partial}{\partial r} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial}{\partial \phi^2}$$
Local Hamiltonians

Quantum system composed of $n$ interacting finite dimensional particles.
Local Hamiltonians

Quantum system composed of $n$ interacting finite dimensional particles.

Hilbert space for a particle: $\mathbb{C}^d$
Local Hamiltonians

Quantum system composed of \( n \) interacting finite dimensional particles.

Hilbert space for a particle: \( \mathbb{C}^d \)

Hilbert space for the whole system:

\[
(\mathbb{C}^d)^\otimes n
\]

Dimension = \( d^n \)
Local Hamiltonians
Local Hamiltonians
Local Hamiltonians

The Hamiltonian for a 3-qubit system is an $8 \times 8$ matrix $H_{1,2,3}$. 
The Hamiltonian for a 3-qubit system is an $8 \times 8$ matrix $H_{1,2,3}$.
The interaction between 3 qubits in an $n$-qubit system is $H_{1,2,3} \otimes I_{4,...,n}$. 
Local Hamiltonians

The Hamiltonian for a 3-qubit system is an $8 \times 8$ matrix $H_{1,2,3}$. The interaction between 3 qubits in an $n$-qubit system is $H_{1,2,3} \otimes I_4,\ldots,n$.

$$h = H_{n-2,n-1,n}$$

$$I_{1,\ldots,n-3} \otimes H_{n-2,n-1,n}$$
Local Hamiltonians

\[ H = \sum_a H_a \]

where each \( H_a \) acts on at most \( k \) qudits
Local Hamiltonians

\[ H = \sum_a H_a \]

where each \( H_a \) acts on at most \( k \) qudits

System consists of \( n \) \( d \)-dimensional particles
Local Hamiltonians

\[ H = \sum_a H_a \]
where each \( H_a \) acts on at most \( k \) qudits

System consists of \( n \) \( d \)-dimensional particles

Hilbert space has dimension \( d^n \)
Hamiltonian is a \( d^n \times d^n \) matrix.
Local Hamiltonians

\[ H = \sum_a H_a \]

where each \( H_a \) acts on at most \( k \) qudits

System consists of \( n \) \( d \)-dimensional particles

Hilbert space has dimension \( d^n \)

Hamiltonian is a \( d^n \times d^n \) matrix.

Succinct representation:
At most \( \binom{n}{k} = O(n^k) \) terms, each specified by \( d^{2k} \) entries.
Local Hamiltonians

\[ H = \sum_a H_a \]

where each \( H_a \) acts on at most \( k \) qudits

System consists of \( n \) \( d \)-dimensional particles

Hilbert space has dimension \( d^n \)

Hamiltonian is a \( d^n \times d^n \) matrix.

Succinct representation:

At most \( \binom{n}{k} = O(n^k) \) terms, each specified by \( d^{2k} \) entries.

What is the ground state of the quantum system?
Local Hamiltonians

\[ H = \sum_a H_a \]
where each \( H_a \) acts on at most \( k \) qudits

System consists of \( n \) \( d \)-dimensional particles

Hilbert space has dimension \( d^n \)
Hamiltonian is a \( d^n \times d^n \) matrix.

Succinct representation:
At most \( \binom{n}{k} = O(n^k) \) terms, each specified by \( d^{2k} \) entries.

Input: Hamiltonian \( H \), real numbers \( E \) and \( \Delta \)
Is the ground energy of \( H \leq E \) or \( \geq E + \Delta \)?
Local Hamiltonian Variations

Locality

$H = \sum_a H_a$
where each $H_a$ acts on at most $k$ qudits
Local Hamiltonian Variations

Locality

\[ H = \sum_a H_a \]
where each \( H_a \) acts on at most \( k \) qudits

Particle Dimension

\( \{ |j\rangle \} \)

\( \{ |0\rangle, |1\rangle, \ldots, |d - 1\rangle \} \)
Local Hamiltonian Variations

Locality

\[ H = \sum_a H_a \]
where each \( H_a \) acts on at most \( k \) qudits

Particle Dimension

\[ \{ |0\rangle, |1\rangle, \ldots, |d-1\rangle \} \]

Geometry
“Spin-Liquid Ground State of the $S = \frac{1}{2}$ Kagome Heisenberg Antiferromagnet”
Yan, Huse, White
Science, Vol 332, June 3, 2011

Kagome Lattice

Heisenberg Antiferromagnet Model

\[
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & -1 & -2 & 0 & 0 \\
0 & -2 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1
\end{pmatrix}
\]
“Spin-Liquid Ground State of the $S = \frac{1}{2}$ Kagome Heisenberg Antiferromagnet”

Yan, Huse, White

Science, Vol 332, June 3, 2011

Is the Ground State a Valence Bond Crystal?

Kagome Lattice
Is the Ground State a Valence Bond Crystal? or a Spin Liquid?

“Spin-Liquid Ground State of the $S = \frac{1}{2}$ Kagome Heisenberg Antiferromagnet”
Yan, Huse, White
Science, Vol 332, June 3, 2011
“A key problem in searching for spin liquids in 2D models is that there are no exact or nearly exact analytical or computational methods to solve infinite 2D quantum lattice systems.”

Yan, Huse, White
Science, Vol 332, June 3, 2011
“A key problem in searching for spin liquids in 2D models is that there are no exact or nearly exact analytical or computational methods to solve infinite 2D quantum lattice systems.”

Yan, Huse, White
Science, Vol 332, June 3, 2011

What is the complexity of the Local Hamiltonian problem?

• Set of local constraints

• Find a global state that minimizes cost
"Classical" Local Hamiltonian

$n d$-dimensional particles: $\mathcal{H} = (\mathbb{C}^d)^{\otimes n}$

Standard basis denoted by classical strings: $|x_1, x_2, \ldots, x_n\rangle$

Each $x_i \in \{0, \ldots, d - 1\}$
"Classical" Local Hamiltonian

$n d$-dimensional particles: $\mathcal{H} = (\mathbb{C}^d)^\otimes n$

Standard basis denoted by classical strings: $|x_1, x_2, \ldots, x_n\rangle$

Each $x_i \in \{0, \ldots, d - 1\}$

Special case of LH: $H = \sum_j H_j$

Each $H_j$ is diagonal in the standard basis.

$H$ is diagonal in the standard basis.
"Classical" Local Hamiltonian

$n d$-dimensional particles: $\mathcal{H} = (\mathbb{C}^d)^\otimes n$

Standard basis denoted by classical strings: $|x_1, x_2, \ldots, x_n\rangle$
Each $x_i \in \{0, \ldots, d - 1\}$

Special case of LH: $H = \sum_j H_j$
Each $H_j$ is diagonal in the standard basis. $H_j$ operates on particles $i_1, i_2, \ldots, i_k$
$H$ is diagonal in the standard basis.

Cost/Energy of setting: $x_{i_1} = a_1, \ldots, x_{i_k} = a_k$
"Classical" Local Hamiltonian

$n d$-dimensional particles: $\mathcal{H} = (\mathbb{C}^d)^{\otimes n}$

Standard basis denoted by classical strings: $|x_1, x_2, \ldots, x_n\rangle$

Each $x_i \in \{0, \ldots, d-1\}$

Special case of LH: $H = \sum_j H_j$

Each $H_j$ is diagonal in the standard basis.

$H$ is diagonal in the standard basis.

Cost/Energy of setting:

$x_{i_1} = a_1, \ldots, x_{i_k} = a_k$

Ground state is a standard basis state (i.e. a classical string)
"Classical" Local Hamiltonian

$n d$-dimensional particles: $\mathcal{H} = (\mathbb{C}^d)^\otimes n$

Standard basis denoted by classical strings: $|x_1, x_2, \ldots, x_n\rangle$

Each $x_i \in \{0, \ldots, d - 1\}$

Special case of LH: $H = \sum_j H_j$

Each $H_j$ is diagonal in the standard basis.

$H$ is diagonal in the standard basis.

Cost/Energy of setting: $x_{i_1} = a_1, \ldots, x_{i_k} = a_k$

Ground state is a standard basis state (i.e. a classical string)

Weighted Constraint Satisfaction Problem
Boolean Satisfiability and 3-SAT

Input: $n$ Boolean variables $x_1, \ldots, x_n$

$m$ clauses: $C_1, \ldots, C_m$.

$C_i$: disjunction of three literals. e.g., $(x_{i1} \lor \neg x_{i2} \lor x_{i3})$

Question: Is there a Boolean assignment to $x_1, \ldots, x_n$ such that

$$C_1 \land C_2 \land \cdots \land C_m = 1 ?$$
Local Hamiltonian is NP-hard

$3\text{SAT} \propto LH$
Local Hamiltonian is NP-hard

\[ 010 \rightarrow \begin{bmatrix}
0 & 0 & 1 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{bmatrix} \]

\[ \Leftrightarrow (x \lor \neg y \lor z) \]

3SAT \propto LH
Local Hamiltonian is NP-hard

$|010\rangle\langle 010| = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

$3\text{SAT} \propto LH$

$\equiv (x \lor \neg y \lor z)$
Local Hamiltonian is NP-hard

$$|010\rangle\langle 010| = \begin{bmatrix}
0 & 0 & 1 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}$$

$$H = \sum_j H_{C_j} \iff C_1 \land C_2 \land \cdots \land C_m$$

$$H$$ has a zero energy ground state

$$3SAT \propto LH \iff (x \lor \neg y \lor z)$$

$$\iff C_1 \land C_2 \land \cdots \land C_m$$

is satisfiable.
The class NP

**NP**

A problem is in NP if there is a polynomial time algorithm $A$ that takes two inputs, $x$ and $y$: 
The class NP

A problem is in NP if there is a polynomial time algorithm $A$ that takes two inputs, $x$ and $y$:

If $x \in L$, then there is a witness $y$ such that $A(x, y)$ accepts.
The class NP

**NP**

A problem is in NP if there is a polynomial time algorithm \( A \) that takes two inputs, \( x \) and \( y \):

If \( x \in L \), then there is a witness \( y \) such that \( A(x, y) \) accepts.

If \( x \notin L \), then for every \( y \), \( A(x, y) \) rejects.

\(|y| \leq \text{poly}(x)\)
The class NP

A problem is in NP if there is a polynomial time algorithm $A$ that takes two inputs, $x$ and $y$:

If $x \in L$, then there is a witness $y$ such that $A(x, y)$ accepts.

If $x \not\in L$, then for every $y$, $A(x, y)$ rejects.

$|y| \leq \text{poly}(x)$

$SAT \in \text{NP}$

$x$ encodes an instance of 3-SAT

Witness $y$:
satisfying assignment $y_i = 0/1$
The class NP

NP

A problem is in NP if there is a polynomial time algorithm $A$ that takes two inputs, $x$ and $y$:

If $x \in L$, then there is a witness $y$ such that $A(x, y)$ accepts.

If $x \notin L$, then for every $y$, $A(x, y)$ rejects.

$|y| \leq \text{poly}(x)$

Poly-sized circuit family $\{C_n\}$

If $|x| = n$, then

$A(x, y)$ accepts $\iff C_n(x, y) = 1$

$A(x, y)$ rejects $\iff C_n(x, y) = 0$
**The class NP**

**NP**

A problem is in NP if there is a polynomial time algorithm $A$ that takes two inputs, $x$ and $y$:

If $x \in L$, then there is a witness $y$ such that $A(x, y)$ accepts.

If $x \notin L$, then for every $y$, $A(x, y)$ rejects.

$|y| \leq \text{poly}(x)$

The circuit family $\{C_n\}$ must be **uniform**:
There is a polynomial time Turing Machine that computes $C_n$ on input $1^n$
Promise Problems

Decision Problems: answer is "Yes" or "No"

\[ L \subseteq \{0, 1\}^* \quad \text{and} \quad x \in L \implies "Yes" \]

\[ x \notin L \implies "No" \]

Promise Problems: input strings partitioned into 3 sets

\[ \text{Yes} \cup \text{No} \cup \text{Invalid} = \{0, 1\}^* \]
The class MA (Merlin-Arthur)

**MA**

A *promise* problem is in MA if there is a polynomial time *randomized* algorithm $R$ that takes two inputs, $x$ and $y$: 
The class MA (Merlin-Arthur)

A *promise* problem is in MA if there is a polynomial time randomized algorithm $R$ that takes two inputs, $x$ and $y$:

If $x \in \text{Yes}$, then there is a witness $y$ such that $R(x, y)$ accepts with prob $\geq \frac{2}{3}$. 

---

Quantum Hamiltonian Complexity - Sandy Irani
The class MA (Merlin-Arthur)

A *promise* problem is in MA if there is a polynomial time randomized algorithm $R$ that takes two inputs, $x$ and $y$:

If $x \in \text{Yes}$, then there is a witness $y$ such that $R(x, y)$ accepts with prob $\geq \frac{2}{3}$.

If $x \in \text{No}$, then for every $y$, $R(x, y)$ accepts with prob $\leq \frac{1}{3}$. 
The class MA (Merlin-Arthur)

**MA**

A *promise* problem is in MA if there is a polynomial time *randomized* algorithm $R$ that takes two inputs, $x$ and $y$:

If $x \in \text{Yes}$, then there is a witness $y$ such that $R(x, y)$ accepts with prob $\geq \frac{2}{3}$.

If $x \in \text{No}$, then for every $y$, $R(x, y)$ accepts with prob $\leq \frac{1}{3}$.

If $x \in \text{Invalid}$, then *no guarantees*!
The class MA (Merlin-Arthur)

MA

A promise problem is in MA if there is a polynomial time randomized algorithm $R$ that takes two inputs, $x$ and $y$:

If $x \in \text{Yes}$, then there is a witness $y$ such that $R(x, y)$ accepts with prob $\geq \frac{2}{3}$.

If $x \in \text{No}$, then for every $y$, $R(x, y)$ accepts with prob $\leq \frac{1}{3}$.

If $x \in \text{Invalid}$, then no guarantees!

$|y| \leq \text{poly}(|x|)$
The class MA (Merlin-Arthur)

A promise problem is in MA if there is a polynomial time randomized algorithm $R$ that takes two inputs, $x$ and $y$:

If $x \in \text{Yes}$, then there is a witness $y$ such that $R(x, y)$ accepts with prob $\geq \frac{2}{3}$.

If $x \in \text{No}$, then for every $y$, $R(x, y)$ accepts with prob $\leq \frac{1}{3}$.

If $x \in \text{Invalid}$, then no guarantees!

$|y| \leq \text{poly}(|x|)$

$R(x, y)$:
Uniform, polynomial-sized circuit family $\{C_n\}$: iff $|x| = n$, then

$x \in \text{Yes} \iff \exists y$ such that $\Pr_r[C_n(x, y, r) = 1] \geq \frac{2}{3}$

$x \in \text{No} \iff \forall y \Pr_r[C_n(x, y, r) = 1] \geq \frac{1}{3}$
The class QMA (Quantum Merlin Arthur)

QMA

A promise problem is in QMA if there is a poly-sized uniform quantum circuit family $\{C_n\}$ such that on input $x$, where $|x| = n$:
The class QMA (Quantum Merlin Arthur)

**QMA**

A *promise* problem is in QMA if there is a poly-sized uniform **quantum** circuit family \( \{ C_n \} \) such that on input \( x \), where \( |x| = n \):

If \( x \in \text{Yes} \), then there is a **quantum** witness \( |\phi\rangle \) such that
\[
\text{Prob}[C_n(x, |\phi\rangle) = 1] \geq 2/3.
\]

If \( x \notin \text{Yes} \), then there is a **quantum** witness \( |\phi\rangle \) such that
\[
\text{Prob}[C_n(x, |\phi\rangle) = 1] \geq 2/3.
\]
The class QMA (Quantum Merlin Arthur)

**QMA**

A *promise* problem is in QMA if there is a poly-sized uniform *quantum* circuit family \( \{C_n\} \) such that on input \( x \), where \( |x| = n \):

If \( x \in \text{Yes} \), then there is a *quantum* witness \( |\phi\rangle \) such that
\[
\text{Prob}[C_n(x, |\phi\rangle) = 1] \geq 2/3.
\]

If \( x \in \text{No} \), then for every \( |\phi\rangle \),
\[
\text{Prob}[C_n(x, |\phi\rangle) = 1] \leq 1/3.
\]
The class QMA (Quantum Merlin Arthur)

**QMA**

A *promise* problem is in QMA if there is a poly-sized uniform quantum circuit family \( \{ C_n \} \) such that on input \( x \), where \( |x| = n \):

- If \( x \in \text{Yes} \), then there is a quantum witness \( |\phi\rangle \) such that 
  \[ \Pr[C_n(x, |\phi\rangle) = 1] \geq \frac{2}{3}. \]

- If \( x \in \text{No} \), then for every \( |\phi\rangle \),
  \[ \Pr[C_n(x, |\phi\rangle) = 1] \leq \frac{1}{3}. \]

- If \( x \in \text{Invalid} \), then no guarantees!
The class QMA (Quantum Merlin Arthur)

QMA

A promise problem is in QMA if there is a poly-sized uniform quantum circuit family \( \{ C_n \} \) such that on input \( x \), where \( |x| = n \):

If \( x \in \text{Yes} \), then there is a quantum witness \( |\phi\rangle \) such that

\[
\text{Prob}[C_n(x, |\phi\rangle) = 1] \geq \frac{2}{3}.
\]

If \( x \in \text{No} \), then for every \( |\phi\rangle \),

\[
\text{Prob}[C_n(x, |\phi\rangle) = 1] \leq \frac{1}{3}.
\]

If \( x \in \text{Invalid} \), then no guarantees!

\( |\phi\rangle \) has poly(n) qubits.

\
\[
|\phi\rangle \quad \text{has poly(n) qubits.}
\]

\[
\text{Prob}[C_n(x, |\phi\rangle) = 1] \geq \frac{2}{3},
\]

\[
\text{Prob}[C_n(x, |\phi\rangle) = 1] \leq \frac{1}{3}.
\]

Quantum Hamiltonian Complexity - Sandy Irani
The class MA - amplification

\textbf{MA}(c, s)

A *promise* problem is in NP if there is a polynomial time *randomized* algorithm \( R \) that takes two inputs, \( x \) and \( y \):
The class MA - amplification

**MA(c, s)**

A *promise* problem is in NP if there is a polynomial time *randomized* algorithm $R$ that takes two inputs, $x$ and $y$:

If $x \in \text{Yes}$, then there is a witness $y$ such that $R(x, y)$ accepts with prob $\geq c$
The class MA - amplification

\textbf{MA}(c, s)

A \textit{promise} problem is in NP if there is a polynomial time \textit{randomized} algorithm \( R \) that takes two inputs, \( x \) and \( y \):

If \( x \in \text{Yes} \), then there is a witness \( y \) such that
\( R(x, y) \) accepts with prob \( \geq c \)

If \( x \in \text{No} \), then for every \( y \),
\( R(x, y) \) accepts with prob \( \leq s \)

\[ |x| = n \text{ and } |y| \leq \text{poly}(n) \]
The class MA - amplification

**MA(c, s)**

A *promise* problem is in NP if there is a polynomial time randomized algorithm $R$ that takes two inputs, $x$ and $y$:

If $x \in \text{Yes}$, then there is a witness $y$ such that $R(x, y)$ accepts with prob $\geq c$

If $x \in \text{No}$, then for every $y$, $R(x, y)$ accepts with prob $\leq s$

$|x| = n$ and $|y| \leq \text{poly}(n)$

If $c - s \geq \frac{1}{n^d}$, then

$$\text{MA}(c, s) = \text{MA} \left(1 - \frac{1}{2^n}, \frac{1}{2^n}\right)$$
The class MA - amplification

**MA(c, s)**

A *promise* problem is in NP if there is a polynomial time \textit{randomized} algorithm \( R \) that takes two inputs, \( x \) and \( y \):

If \( x \in \text{Yes} \), then there is a witness \( y \) such that \( R(x, y) \) accepts with prob \( \geq c \)

If \( x \in \text{No} \), then for every \( y \), \( R(x, y) \) accepts with prob \( \leq s \)

\(|x| = n\) and \(|y| \leq \text{poly}(n)\)

\[
\text{If } c - s \geq \frac{1}{n^d}, \text{ then } \\
\text{MA}(c, s) = \text{MA} \left( 1 - \frac{1}{2^n}, \frac{1}{2^n} \right)
\]
The class MA - amplification

MA(c, s)

A promise problem is in NP if there is a polynomial time randomized algorithm $R$ that takes two inputs, $x$ and $y$:

If $x \in \text{Yes}$, then there is a witness $y$ such that $R(x, y)$ accepts with prob $\geq c$

If $x \in \text{No}$, then for every $y$, $R(x, y)$ accepts with prob $\leq s$

$|x| = n$ and $|y| \leq \text{poly}(n)$

If $c - s \geq \frac{1}{n^d}$, then

$\text{MA}(c, s) = \text{MA} \left( 1 - \frac{1}{2^n}, \frac{1}{2^n} \right)$

Repeat $m$ times (with fresh random bits)

Threshold for acc $= \left( \frac{c+s}{2} \right)^m$
The class MA - amplification

**MA(c, s)**

A *promise* problem is in NP if there is a polynomial time randomized algorithm $R$ that takes two inputs, $x$ and $y$:

If $x \in \text{Yes}$, then there is a witness $y$ such that $R(x, y)$ accepts with prob $\geq c$

If $x \in \text{No}$, then for every $y$, $R(x, y)$ accepts with prob $\leq s$

$|x| = n$ and $|y| \leq \text{poly}(n)$

If $c - s \geq \frac{1}{nd}$, then

$\text{MA}(c, s) = \text{MA}\left(1 - \frac{1}{2^n}, \frac{1}{2^n}\right)$

**Repeat** $m$ times (with fresh random bits)

**Threshold for acc** $= \left(\frac{c+s}{2}\right)^m$

By Chernoff’s Inequality

For $m = \text{sufficiently large polynomial in } n$, Probability number of accepts deviates from the expectation by more than $\left(\frac{c-s}{2}\right)^m$ is exponentially small
The class QMA - amplification

**QMA(c, s)**

A *promise* problem is in QMA if there is a poly-sized uniform 
**quantum** circuit family \( \{ C_n \} \) such that on input \( x \), where \( |x| = n \):
The class QMA - amplification

**QMA(c, s)**

A *promise* problem is in QMA if there is a poly-sized uniform quantum circuit family \( \{ C_n \} \) such that on input \( x \), where \( |x| = n \):

If \( x \in \text{Yes} \), then there is a quantum witness \( |\phi\rangle \) such that
\[
\Pr[C_n(x, |\phi\rangle) = 1] \geq c
\]

If \( x \in \text{No} \), then for every \( |\phi\rangle \),
\[
\Pr[C_n(x, |\phi\rangle) = 1] \leq s
\]

\( |\phi\rangle \) has \( y = \text{poly}(n) \) qubits.
The class QMA - amplification

**QMA(c, s)**

A *promise* problem is in QMA if there is a poly-sized uniform quantum circuit family \( \{ C_n \} \) such that on input \( x \), where \( |x| = n \):

If \( x \in \text{Yes} \), then there is a quantum witness \( |\phi\rangle \) such that

\[
\text{Prob}[C_n(x, |\phi\rangle) = 1] \geq c
\]

If \( x \in \text{No} \), then for every \( |\phi\rangle \),

\[
\text{Prob}[C_n(x, |\phi\rangle) = 1] \leq s
\]

\( |\phi\rangle \) has \( y = \text{poly}(n) \) qubits.

If \( c - s \geq \frac{1}{n^d} \), then

\[
\text{QMA}(c, s) = \text{QMA} \left( 1 - \frac{1}{2^n}, \frac{1}{2^n} \right)
\]
The class QMA - amplification

QMA(c, s)

A promise problem is in QMA if there is a poly-sized uniform quantum circuit family \{C_n\} such that on input $x$, where $|x| = n$:

- If $x \in \text{Yes}$, then there is a quantum witness $|\phi\rangle$ such that
  $\Pr[C_n(x, |\phi\rangle) = 1] \geq c$

- If $x \in \text{No}$, then for every $|\phi\rangle$,
  $\Pr[C_n(x, |\phi\rangle) = 1] \leq s$

$|\phi\rangle$ has $y = \text{poly}(n)$ qubits.

If $c - s \geq \frac{1}{n^d}$, then

$\text{QMA}(c, s) = \text{QMA} \left(1 - \frac{1}{2^n}, \frac{1}{2^n}\right)$

Completeness:

$|\Phi\rangle |\Phi\rangle |\Phi\rangle \cdots |\Phi\rangle$

$m$ independent copies
The class QMA - amplification

**QMA(c, s)**

A *promise* problem is in QMA if there is a poly-sized uniform quantum circuit family \( \{ C_n \} \) such that on input \( x \), where \( |x| = n \):

- If \( x \in \text{Yes} \), then there is a quantum witness \( |\phi\rangle \) such that 
  \[
  \text{Prob}[C_n(x, |\phi\rangle) = 1] \geq c
  \]

- If \( x \in \text{No} \), then for every \( |\phi\rangle \), 
  \[
  \text{Prob}[C_n(x, |\phi\rangle) = 1] \leq s
  \]

|\phi\rangle \text{ has } y = \text{poly}(n) \text{ qubits.}

If \( c - s \geq \frac{1}{n^d} \), then

\[
\text{QMA}(c, s) = \text{QMA} \left( 1 - \frac{1}{2^n}, \frac{1}{2^n} \right)
\]

**Completeness:**

\[
|\Phi\rangle |\Phi\rangle |\Phi\rangle \ldots |\Phi\rangle
\]

\( m \) independent copies

**Soundness:**

\[
V_1 V_2 V_3 \ldots V_m
\]

\( m \cdot y \) qubits
The class QMA - amplification

**QMA(c, s)**

A *promise* problem is in QMA if there is a poly-sized uniform quantum circuit family \( \{ C_n \} \) such that on input \( x \), where \( |x| = n \):

If \( x \in \text{Yes} \), then there is a quantum witness \( |\phi\rangle \) such that
\[
\text{Prob}[C_n(x, |\phi\rangle) = 1] \geq c
\]

If \( x \in \text{No} \), then for every \( |\phi\rangle \),
\[
\text{Prob}[C_n(x, |\phi\rangle) = 1] \leq s
\]

\( |\phi\rangle \) has \( y = \text{poly}(n) \) qubits.

---

If \( c - s \geq \frac{1}{n^d} \), then
\[
\text{QMA}(c, s) = \text{QMA} \left( 1 - \frac{1}{2^n}, \frac{1}{2^n} \right)
\]

**Completeness:**

\[
|\Phi\rangle |\Phi\rangle |\Phi\rangle \cdots |\Phi\rangle
\]

\( m \) independent copies

**Soundness:**

\[
V_1 V_2 V_3 \cdots V_m
\]

\( m \cdot y \) qubits

\[
\text{Prob} \text{ACC} \leq s
\]
The class QMA - amplification

**QMA(c, s)**

A *promise* problem is in QMA if there is a poly-sized uniform quantum circuit family \(\{C_n\}\) such that on input \(x\), where \(|x| = n\):

- If \(x \in \text{Yes}\), then there is a quantum witness \(|\phi\rangle\) such that
  \[
  \text{Prob}[C_n(x, |\phi\rangle) = 1] \geq c
  \]
- If \(x \in \text{No}\), then for every \(|\phi\rangle\),
  \[
  \text{Prob}[C_n(x, |\phi\rangle) = 1] \leq s
  \]

\(|\phi\rangle\) has \(y = \text{poly}(n)\) qubits.

If \(c - s \geq \frac{1}{n^d}\), then

\[
\text{QMA}(c, s) = \text{QMA}\left(1 - \frac{1}{2^n}, \frac{1}{2^n}\right)
\]

**Completeness:**

\[
|\Phi\rangle |\Phi\rangle |\Phi\rangle \cdots |\Phi\rangle
\]

\(m\) independent copies

**Soundness:**

\[
V_1 \quad V_2 \quad V_3 \quad \cdots \quad V_m
\]

\(m \cdot y\) qubits

\[
\begin{align*}
\text{Prob} & \leq s \\
\text{ACC} & \leq s
\end{align*}
\]

Even when conditioned on outcome of \(V_1\)’s measurement
The class QMA - amplification

**QMA(c, s)**

A *promise* problem is in QMA if there is a poly-sized uniform quantum circuit family \( \{ C_n \} \) such that on input \( x \), where \( |x| = n \):

If \( x \in \text{Yes} \), then there is a quantum witness \( |\phi\rangle \) such that
\[
\text{Prob}[C_n(x, |\phi\rangle) = 1] \geq c
\]

If \( x \in \text{No} \), then for every \( |\phi\rangle \)
\[
\text{Prob}[C_n(x, |\phi\rangle) = 1] \leq s
\]

\( |\phi\rangle \) has \( y = \text{poly}(n) \) qubits.

If \( c - s \geq \frac{1}{n^d} \), then
\[
\text{QMA}(c, s) = \text{QMA} \left( 1 - \frac{1}{2^n}, \frac{1}{2^n} \right)
\]

**Completeness:**

\[
|\Phi\rangle |\Phi\rangle |\Phi\rangle \cdots |\Phi\rangle
\]

\( m \) independent copies

**Soundness:**

\[
V_1 \ V_2 \ V_3 \cdots \ V_m
\]

\( m \cdot y \) qubits

\[
\text{Prob} \leq s \quad \text{and} \quad \text{Prob} \leq s
\]

Even when conditioned on outcome of \( V_1 \)'s measurement
The Marriott-Watrous "Trick"

**QMA\_y(c, s)**

A *promise* problem is in QMA if there is a poly-sized uniform quantum circuit family \( \{ C_n \} \) such that on input \( x \), where \( |x| = n \):

If \( x \in \text{Yes} \), then there is a quantum witness \( |\phi\rangle \) such that
\[
\text{Prob}[C_n(x, |\phi\rangle) = 1] \geq c
\]

If \( x \in \text{No} \), then for every \( |\phi\rangle \),
\[
\text{Prob}[C_n(x, |\phi\rangle) = 1] \leq s
\]

\( |\phi\rangle \) has \( y(n) \) qubits.
The Marriott-Watrous “Trick”

$\text{QMA}_y(c, s)$

A promise problem is in QMA if there is a poly-sized uniform quantum circuit family $\{C_n\}$ such that on input $x$, where $|x| = n$:

If $x \in \text{Yes}$, then there is a quantum witness $|\phi\rangle$ such that
\[ \text{Prob}[C_n(x, |\phi\rangle) = 1] \geq c \]

If $x \in \text{No}$, then for every $|\phi\rangle$,
\[ \text{Prob}[C_n(x, |\phi\rangle) = 1] \leq s \]

$|\phi\rangle$ has $y(n)$ qubits.

If $c - s \geq \frac{1}{n^c}$, then

$\text{QMA}_y(c, s) = \text{QMA}_y \left(1 - \frac{1}{2^n}, \frac{1}{2^n}\right)$

for every polynomial $y$.
The Marriott-Watrous “Trick”

**QMA\_y(c, s)**

A *promise* problem is in QMA if there is a poly-sized uniform quantum circuit family \( \{ C_n \} \) such that on input \( x \), where \(|x| = n\):

If \( x \in \text{Yes} \), then there is a quantum witness \( |\phi\rangle \) such that
\[
\text{Prob}[C_n(x, |\phi\rangle) = 1] \geq c
\]

If \( x \in \text{No} \), then for every \( |\phi\rangle \),
\[
\text{Prob}[C_n(x, |\phi\rangle) = 1] \leq s
\]

\(|\phi\rangle\) has \( y(n) \) qubits.

If \( c - s \geq \frac{1}{nc} \), then
\[
\text{QMA}_y(c, s) = \text{QMA}_y \left( 1 - \frac{1}{2^n}, \frac{1}{2^n} \right)
\]
for every polynomial \( y \)

Probabilistically try and back up after a measurement.

Measure for a successful back up.

Principle of deferred measurements.
Complexity Classes and Complete Problems

\[ \text{NP} \subseteq \text{MA} \subseteq \text{QMA} \]
Complexity Classes and Complete Problems

\[ \text{NP} \subseteq \text{MA} \subseteq \text{QMA} \subseteq \text{PP} \subseteq \text{PSPACE} \]
Complexity Classes and Complete Problems

Boolean satisfiability is complete for NP

[Cook-Levin]

$\text{NP} \subseteq \text{MA} \subseteq \text{QMA} \subseteq \text{PP} \subseteq \text{PSPACE}$
Complexity Classes and Complete Problems

- Boolean satisfiability is complete for NP
  [Cook-Levin]
- Local Hamiltonian is complete for QMA
  [Kitaev]

\[
\text{NP} \subseteq \text{MA} \subseteq \text{QMA} \subseteq \text{PP} \subseteq \text{PSPACE}
\]
Complexity Classes and Complete Problems

$NP \subseteq MA \subseteq QMA \subseteq PP \subseteq PSPACE$

Boolean satisfiability is complete for NP
[Cook-Levin]

Local Hamiltonian is complete for QMA
[Kitaev]

$P \subseteq BPP \subseteq BQP$
Complexity Classes and Complete Problems

NP ⊆ MA ⊆ QMA ⊆ PP ⊆ PSPACE

Boolean satisfiability is complete for NP
[Cook-Levin]

Local Hamiltonian is complete for QMA
[Kitaev]

P ⊆ BPP ⊆ BQP
The Local Hamiltonian Problem

Input:

\[ H_1, \ldots, H_r, \text{ set of Hermitian positive semi-definite matrices operating on } k \text{ qudits of dimension } d, \text{ with bounded norm } \| H_i \| \leq 1. \]
The Local Hamiltonian Problem

Input:

$H_1, \ldots, H_r$, set of Hermitian positive semi-definite matrices operating on $k$ qudits of dimension $d$, with bounded norm $\|H_i\| \leq 1$. 
The Local Hamiltonian Problem

Input:

$H_1, \ldots, H_r$, set of Hermitian positive semi-definite matrices operating on $k$ qudits of dimension $d$, with bounded norm $\|H_i\| \leq 1$.

Each matrix indicates the set of $k$ qudits (out of the set of $n$ qudits in the system) on which it operates. Each matrix is given with poly$(n)$ bits.
The Local Hamiltonian Problem

Input:

$H_1, \ldots, H_r$, set of Hermitian positive semi-definite matrices operating on $k$ qudits of dimension $d$, with bounded norm $\|H_i\| \leq 1$.

Each matrix indicates the set of $k$ qudits (out of the set of $n$ qudits in the system) on which it operates. Each matrix is given with $\text{poly}(n)$ bits.

Two real numbers $E$ and $\Delta \geq 1/\text{poly}(n)$
The Local Hamiltonian Problem

Input:

\[ H_1, \ldots, H_r, \text{ set of Hermitian positive semi-definite matrices operating on } k \text{ qudits of dimension } d, \text{ with bounded norm } \|H_i\| \leq 1. \]

Each matrix indicates the set of \( k \) qudits (out of the set of \( n \) qudits in the system) on which it operates. Each matrix is given with \( \text{poly}(n) \) bits.

Two real numbers \( E \) and \( \Delta \geq 1/\text{poly}(n) \)

Output:

Is the smallest eigenvalue of \( H = H_1 + \cdots + H_r \leq E \) or are all eigenvalues \( \geq E + \Delta \)?
The Local Hamiltonian Problem

**Input:**

$H_1, \ldots, H_r$, set of Hermitian positive semi-definite matrices operating on $k$ qudits of dimension $d$, with bounded norm $\|H_i\| \leq 1$.

Eigenvalues of each $H_i$ in $[0, 1]$.

$H_i + \alpha I \rightarrow$ eigenvalues of $H$ shift by $\alpha$

$\alpha H \rightarrow$ eigenvalues of $H$ scale by factor of $\alpha$

Each matrix indicates the set of $k$ qudits (out of the set of $n$ qudits in the system) on which it operates. Each matrix is given with poly($n$) bits.

Two real numbers $E$ and $\Delta \geq 1/poly(n)$

**Output:**

Is the smallest eigenvalue of $H = H_1 + \cdots + H_r \leq E$ or are all eigenvalues $\geq E + \Delta$?
Local Hamiltonian is in QMA

Boolean Satisfiability $\in \text{NP}$
Local Hamiltonian is in QMA

Boolean Satisfiability ∈ NP

Is $\Phi(y)$ satisfiable?

Witness: Satisfying assignment $y$
Local Hamiltonian is in QMA

Boolean Satisfiability $\in$ NP

Is $\Phi(y)$ satisfiable?
Witness: Satisfying assignment $y$

Local Hamiltonian $\in$ QMA
Local Hamiltonian is in QMA

Boolean Satisfiability $\in$ NP

Local Hamiltonian $\in$ QMA

Is $\Phi(y)$ satisfiable?
Witness: Satisfying assignment $y$

Is there a state whose energy (according to $H$) is less than $E$?
$\langle \Phi | H | \Phi \rangle \leq E$?
Local Hamiltonian is in QMA

Boolean Satisfiability ∈ NP

Local Hamiltonian ∈ QMA

Is $\Phi(y)$ satisfiable? 
Witness: Satisfying assignment $y$

Is there a state whose energy (according to $H$) is less than $E$? 
$\langle \Phi | H | \Phi \rangle \leq E$?
Witness: $|\Phi\rangle$
Local Hamiltonian is in QMA

Boolean Satisfiability $\in \text{NP}$

Is $\Phi(y)$ satisfiable?
Witness: Satisfying assignment $y$

Is there a state whose energy (according to $H$) is less than $E$?
$\langle \Phi|H|\Phi \rangle \leq E$?
Witness: $|\Phi\rangle$

Guarantee:
There exists $|\Phi\rangle$ such that $\langle \Phi|H|\Phi \rangle \leq E$
OR
For all $|\Phi\rangle$, $\langle \Phi|H|\Phi \rangle \geq E + \Delta$
Local Hamiltonian is in QMA

Boolean Satisfiability \(\in\) NP

Is \(\Phi(y)\) satisfiable?
Witness: Satisfying assignment \(y\)

Local Hamiltonian \(\in\) QMA

Is there a state whose energy (according to \(H\)) is less than \(E\)?
\(\langle \Phi | H | \Phi \rangle \leq E?\)
Witness: \(|\Phi\rangle\)

Guarantee:
There exists \(|\Phi\rangle\) such that \(\langle \Phi | H | \Phi \rangle \leq E\)
OR
For all \(|\Phi\rangle\), \(\langle \Phi | H | \Phi \rangle \geq E + \Delta\)

Need a measurement whose outcome = 1 with probability \(\propto \langle \Phi | H | \Phi \rangle\).
Local Hamiltonian is in QMA

\[ H = H_1 + H_2 + \cdots + H_r \]  Each \( H_i \) is \( k \)-local
Local Hamiltonian is in QMA

\[ H = H_1 + H_2 + \cdots + H_r \]  Each \( H_i \) is \( k \)-local

Pick \( H_a \) at random where \( H_a = \sum_j \lambda_{aj} |v_{aj}\rangle\langle v_{aj}| \)

(recall \( 0 \leq \lambda_{aj} \leq 1 \))
Local Hamiltonian is in QMA

\[ H = H_1 + H_2 + \cdots + H_r \]

Each \( H_i \) is \( k \)-local

Pick \( H_a \) at random where

\[ H_a = \sum_j \lambda_{aj} |v_{aj}\rangle \langle v_{aj}| \]

(recall \( 0 \leq \lambda_{aj} \leq 1 \))

Add auxiliary bit and implement unitary:
For every \( j \):

\[
|v_{aj}\rangle |0\rangle \Rightarrow |v_{aj}\rangle \left( \sqrt{1 - \lambda_{aj}} |0\rangle + \sqrt{\lambda_{aj}} |1\rangle \right)
\]

Measure last qubit
Local Hamiltonian is in QMA

\[ H = H_1 + H_2 + \cdots + H_r \quad \text{Each } H_i \text{ is } k\text{-local} \]

Pick \( H_a \) at random where

\[ H_a = \sum_j \lambda_{aj} |v_{aj}\rangle \langle v_{aj}| \quad (\text{recall } 0 \leq \lambda_{aj} \leq 1) \]

Add auxiliary bit and implement unitary:

For every \( j \):

\[ |v_{aj}\rangle |0\rangle \Rightarrow |v_{aj}\rangle \left( \sqrt{1 - \lambda_{aj}} |0\rangle + \sqrt{\lambda_{aj}} |1\rangle \right) \]

Measure last qubit

\[ |\Phi\rangle |0\rangle = \sum_j \alpha_{aj} |v_{aj}\rangle |\beta_{aj}\rangle |0\rangle \]

\( k \) qubits

\( H_a \) acts on

the rest of the qubits
Local Hamiltonian is in QMA

\[ H = H_1 + H_2 + \cdots + H_r \quad \text{Each } H_i \text{ is } k\text{-local} \]

Pick \( H_a \) at random where

\[ H_a = \sum_j \lambda_{aj} |v_{aj}\rangle \langle v_{aj}| \]

(recall \( 0 \leq \lambda_{aj} \leq 1 \))

Add auxiliary bit and implement unitary:

For every \( j \):

\[ |v_{aj}\rangle|0\rangle \Rightarrow |v_{aj}\rangle \left( \sqrt{1 - \lambda_{aj}} |0\rangle + \sqrt{\lambda_{aj}} |1\rangle \right) \]

Measure last qubit

\[ |\Phi\rangle|0\rangle = \sum_j \alpha_{aj} |v_{aj}\rangle|\beta_{aj}\rangle|0\rangle \Rightarrow \]

\[ \sum_j \alpha_{aj} |v_{aj}\rangle|\beta_{aj}\rangle \left( \sqrt{1 - \lambda_{aj}} |0\rangle + \sqrt{\lambda_{aj}} |1\rangle \right) \]
Local Hamiltonian is in QMA

\[ H = H_1 + H_2 + \cdots + H_r \quad \text{Each } H_i \text{ is } k\text{-local} \]

Pick \( H_a \) at random where \( H_a = \sum_j \lambda_{aj} |v_{aj} \rangle \langle v_{aj}| \)

(recall \( 0 \leq \lambda_{aj} \leq 1 \))

Add auxiliary bit and implement unitary:

For every \( j \):

\[ |v_{aj} \rangle |0\rangle \Rightarrow |v_{aj} \rangle \left( \sqrt{1 - \lambda_{aj}} |0\rangle + \sqrt{\lambda_{aj}} |1\rangle \right) \]

Measure last qubit

\[ |\Phi\rangle |0\rangle = \sum_j \alpha_{aj} |v_{aj} \rangle |\beta_{aj}\rangle |0\rangle \Rightarrow \]

\[ \sum_j \alpha_{aj} |v_{aj} \rangle |\beta_{aj}\rangle \left( \sqrt{1 - \lambda_{aj}} |0\rangle + \sqrt{\lambda_{aj}} |1\rangle \right) \]

Prob of measuring 1:
Local Hamiltonian is in QMA

\[ H = H_1 + H_2 + \cdots + H_r \quad \text{Each } H_i \text{ is } k\text{-local} \]

Pick \( H_a \) at random where

\[ H_a = \sum_j \lambda_{aj} |v_{aj}\rangle \langle v_{aj}| \]

(recall \( 0 \leq \lambda_{aj} \leq 1 \))

Add auxiliary bit and implement unitary:

For every \( j \):

\[ |v_{aj}\rangle |0\rangle \Rightarrow |v_{aj}\rangle (\sqrt{1 - \lambda_{aj}} |0\rangle + \sqrt{\lambda_{aj}} |1\rangle) \]

Measure last qubit

\[ |\Phi\rangle |0\rangle = \sum_j \alpha_{aj} |v_{aj}\rangle |\beta_{aj}\rangle |0\rangle \Rightarrow \]

\[ \sum_j \alpha_{aj} |v_{aj}\rangle |\beta_{aj}\rangle (\sqrt{1 - \lambda_{aj}} |0\rangle + \sqrt{\lambda_{aj}} |1\rangle) \]

Prob of measuring 1: \( \sum_j |\alpha_{aj}|^2 \lambda_{aj} \)
Local Hamiltonian is in QMA

\[ H = H_1 + H_2 + \cdots + H_r \quad \text{Each } H_i \text{ is } k\text{-local} \]

Pick \( H_a \) at random where \( H_a = \sum_j \lambda_{aj} |v_{aj}\rangle \langle v_{aj}| \) 
(recall \( 0 \leq \lambda_{aj} \leq 1 \))

Add auxiliary bit and implement unitary:
For every \( j \):
\[ |v_{aj}\rangle |0\rangle \implies |v_{aj}\rangle \left( \sqrt{1 - \lambda_{aj}} |0\rangle + \sqrt{\lambda_{aj}} |1\rangle \right) \]

Measure last qubit
\[ |\Phi\rangle |0\rangle = \sum_j \alpha_{aj} |v_{aj}\rangle |\beta_{aj}\rangle |0\rangle \implies \]
\[ \sum_j \alpha_{aj} |v_{aj}\rangle |\beta_{aj}\rangle \left( \sqrt{1 - \lambda_{aj}} |0\rangle + \sqrt{\lambda_{aj}} |1\rangle \right) \]

Prob of measuring 1: \[ \sum_j |\alpha_{aj}|^2 \lambda_{aj} = \langle \Phi | H_a | \Phi \rangle \]
Local Hamiltonian is in QMA

\[ H = H_1 + H_2 + \cdots + H_r \quad \text{Each } H_i \text{ is } k\text{-local} \]
Local Hamiltonian is in QMA

\[ H = H_1 + H_2 + \cdots + H_r \quad \text{Each } H_i \text{ is } k\text{-local} \]

Pick \( H_a \) at random where \( H_a = \sum_j \lambda_{aj} |v_{aj}\rangle \langle v_{aj}| \)

(recall \( 0 \leq \lambda_{aj} \leq 1 \))
Local Hamiltonian is in QMA

\[ H = H_1 + H_2 + \cdots + H_r \]  
Each \( H_i \) is \( k \)-local

Pick \( H_a \) at random where \( H_a = \sum_j \lambda_{aj} |v_{aj} \rangle \langle v_{aj}| \)  
(recall \( 0 \leq \lambda_{aj} \leq 1 \))

If \( H_a \) is picked, prob of measuring 1:  
\[ = \langle \Phi | H_a | \Phi \rangle \]
Local Hamiltonian is in QMA

\[ H = H_1 + H_2 + \cdots + H_r \quad \text{Each } H_i \text{ is } k\text{-local} \]

Pick \( H_a \) at random where \( H_a = \sum_j \lambda_{aj} |v_{aj}\rangle\langle v_{aj}| \)

(recall \( 0 \leq \lambda_{aj} \leq 1 \))

If \( H_a \) is picked, prob of measuring 1:

\[ = \langle \Phi | H_a | \Phi \rangle \]

Probability of measuring 1 (overall):

\[ = \frac{1}{r} \sum_{a=1}^{r} \langle \Phi | H_a | \Phi \rangle = \frac{1}{r} \langle \Phi | H | \Phi \rangle \]
Local Hamiltonian is in QMA

\[ H = H_1 + H_2 + \cdots + H_r \quad \text{Each } H_i \text{ is } k\text{-local} \]

Pick \( H_a \) at random where

\[ H_a = \sum_j \lambda_{aj} |v_{aj}\rangle \langle v_{aj}| \]

(recall \( 0 \leq \lambda_{aj} \leq 1 \))

If \( H_a \) is picked, prob of measuring 1:

\[ = \langle \Phi | H_a | \Phi \rangle \]

Probability of measuring 1 (overall):

\[ = \frac{1}{r} \sum_{a=1}^{r} \langle \Phi | H_a | \Phi \rangle = \frac{1}{r} \langle \Phi | H | \Phi \rangle \]

either \( \leq E/r \quad \text{OR} \quad \geq (E + \Delta)/r \)
Boolean Satisfiability is NP-hard  [Cook-Levin]

Start with a generic language $L$ in NP

Is $x \in L$?
Boolean Satisfiability is NP-hard \[\text{[Cook-Levin]}\]

Start with a generic language L in NP

Is \(x \in L\)?

Is there a string y that causes this circuit to output 1?
Boolean Satisfiability is NP-hard [Cook-Levin]

Start with a generic language $L$ in NP

Is $x \in L$?

\[
\begin{array}{c}
\text{Is there a string } y \text{ that causes this circuit to output 1?}
\end{array}
\]

Boolean Formula:

$\Phi_x(y)$
Boolean Satisfiability is NP-hard \text{[Cook-Levin]}

Start with a generic language $L$ in NP

Is $x \in L$?

\[ x \in L \]

\[ \Phi_x(y) \]

Is there a string $y$ that causes this circuit to output 1? \iff Is $\Phi_x(y)$ satisfiable?
Local Hamiltonian is QMA-hard

Start with a generic promise problem in QMA

Is $x \in \text{Yes}$? or is $x \in \text{NO}$?
Local Hamiltonian is QMA-hard

Start with a generic promise problem in QMA

Is $x \in \text{Yes?}$ or is $x \in \text{NO?}$

Is there a quantum state $\phi$ that causes this quantum circuit to output 1 with high probability?
Local Hamiltonian is QMA-hard

Start with a generic promise problem in QMA

Is \( x \in \text{Yes?} \) or is \( x \in \text{NO?} \)

\[
\begin{bmatrix}
|0\rangle/|1\rangle \\
|0\rangle/|1\rangle \\
|0\rangle/|1\rangle \\
|0\rangle/|1\rangle \\
|0\rangle/|1\rangle \\
|0\rangle/|1\rangle
\end{bmatrix}

\]

\[
C_n 
\]

\[
M \xrightarrow{0/1}
\]

\[
5\text{-Local Hamiltonian:} \\
(H_x, E, \Delta)
\]

Is there a quantum state \( \phi \rangle \) that causes this quantum circuit to output 1 with high probability?
Local Hamiltonian is QMA-hard [Kitaev 1995]

Start with a generic promise problem in QMA

Is $x \in \text{Yes?}$ or is $x \in \text{NO?}$

\[
\begin{align*}
|\phi\rangle & \quad |\phi\rangle \\
|0\rangle & \quad |1\rangle
\end{align*}
\]

Is there a quantum state $\phi\rangle$ that causes this quantum circuit to output 1 with high probability?

\[
\begin{align*}
\mathcal{C}_n & \quad 0/1 \\
(\mathcal{H}_x, E, \Delta) & \quad \Rightarrow \\
\Rightarrow & \quad \text{Ground energy of } H_x \leq E
\end{align*}
\]
Local Hamiltonian is QMA-hard \cite[Kitaev 1995]{1995Kitaev}

Start with a generic promise problem in QMA

Is \( x \in \text{Yes?} \) or is \( x \in \text{NO?} \)

\[
\begin{array}{c}
\left| x \right\rangle \\
\left| 0 \right\rangle / \left| 1 \right\rangle \\
\left| 0 \right\rangle / \left| 1 \right\rangle \\
\left| 0 \right\rangle / \left| 1 \right\rangle \\
\left| 0 \right\rangle / \left| 1 \right\rangle \\
\left| 0 \right\rangle / \left| 1 \right\rangle \\
\end{array}
\]

\[
\begin{array}{c}
\left| \phi \right\rangle \\
\left| 0 \right\rangle / \left| 1 \right\rangle \\
\left| 0 \right\rangle / \left| 1 \right\rangle \\
\left| 0 \right\rangle / \left| 1 \right\rangle \\
\left| 0 \right\rangle / \left| 1 \right\rangle \\
\left| 0 \right\rangle / \left| 1 \right\rangle \\
\end{array}
\]

5-Local Hamiltonian:
\((H_x, E, \Delta)\)

Is there a quantum state \( \left| \phi \right\rangle \)
that causes this quantum circuit
to output 1 with high probability?

For every \( \left| \phi \right\rangle \), circuit outputs 0 w.h.p.

\[
\Rightarrow \text{Ground energy of } H_x \leq E
\]

\[
\Rightarrow \text{Ground energy of } H_x \geq E + \Delta
\]
Boolean Satisfiability is NP-hard

Start with a generic language $L$ in NP

Is $x \in L$?
Boolean Satisfiability is NP-hard

Start with a generic language $L$ in NP

Is $x \in L$?

Is there a string $y$ that causes this circuit to output 1?
Boolean Satisfiability is NP-hard

Start with a generic language \( L \) in NP

Is \( x \in L \)?

**Reduction:** input \( x \)

- Use \( |x| = n \) to compute \( C_n \) (uniformity)

Is there a string \( y \) that causes this circuit to output 1?
Boolean Satisfiability is NP-hard

Start with a generic language $L$ in NP

Is $x \in L$?

Reduction: input $x$

- Use $|x| = n$ to compute $C_n$ (uniformity)
- Convert $C_n$ to a Boolean formula

Is there a string $y$ that causes this circuit to output 1?
Boolean Satisfiability is NP-hard

Start with a generic language $L$ in NP

Is $x \in L$?

Reduction: input $x$

- Use $|x| = n$ to compute $C_n$ (uniformity)
- Convert $C_n$ to a Boolean formula
- Add terms to hard-code input $x$ and enforce output $= 1$.

Is there a string $y$ that causes this circuit to output 1?
Circuit to Boolean formula

Circuit $C_n$ has gates $G_1, \ldots, G_m$, where $m = poly(n)$.

Add variables $g_1, \ldots, g_m$, one for each gate.
Circuit to Boolean formula

Circuit $C_n$ has gates $G_1, \ldots, G_m$, where $m = \text{poly}(n)$.

Add variables $g_1, \ldots, g_m$, one for each gate.

Add a clause for each gate:

1. OR gate:
   
   $a \lor b \Rightarrow (a \lor b) \leftrightarrow g_i$

2. AND gate:
   
   $a \land b \Rightarrow (a \land b) \leftrightarrow g_i$

3. NOT gate:
   
   $\neg a \Rightarrow (\neg a) \leftrightarrow g_i$
Circuit to Boolean formula

Circuit $C_n$ has gates $G_1, \ldots, G_m$, where $m = poly(n)$.

Add variables $g_1, \ldots, g_m$, one for each gate.

Add a clause for each gate:

\[
\begin{align*}
&\text{OR:} \quad a \rightarrow g_i \Rightarrow (a \lor b \iff g_i) \\
&\text{AND:} \quad a \rightarrow g_i \Rightarrow (a \land b \iff g_i) \\
&\text{NOT:} \quad a \rightarrow g_i \Rightarrow (\neg a \iff g_i)
\end{align*}
\]

Hard-code $x$:

\[
\begin{align*}
x_i = 0 & \rightarrow \text{add clause } (\neg x_i) \\
x_i = 1 & \rightarrow \text{add clause } (x_i)
\end{align*}
\]
Circuit to Boolean formula

Circuit $C_n$ has gates $G_1, \ldots, G_m$, where $m = \text{poly}(n)$.

Add variables $g_1, \ldots, g_m$, one for each gate.

Add a clause for each gate:

- For OR gate:
  \[
  g_i \Rightarrow ((a \lor b) \leftrightarrow g_i)
  \]

- For AND gate:
  \[
  g_i \Rightarrow ((a \land b) \leftrightarrow g_i)
  \]

- For NOT gate:
  \[
  g_i \Rightarrow (\neg a \leftrightarrow g_i)
  \]

Hard-code $x$:
- $x_i = 0 \rightarrow$ add clause $(\neg x_i)$
- $x_i = 1 \rightarrow$ add clause $(x_i)$

Output of $G_m = \text{output of circuit}$:
Add clause $(g_m)$
Circuit to Boolean formula

Circuit $C_n$ has gates $G_1, \ldots, G_m$, where $m = poly(n)$.

Add variables $g_1, \ldots, g_m$, one for each gate.

Add a clause for each gate:

- $a \lor b \implies ((a \lor b) \leftrightarrow g_i)$
- $a \land b \implies ((a \land b) \leftrightarrow g_i)$
- $\neg a \implies (\neg a \leftrightarrow g_i)$

Boolean formula is the conjunction of all the clauses.

Hard-code $x$:
- $x_i = 0 \implies \text{add clause } (\neg x_i)$
- $x_i = 1 \implies \text{add clause } (x_i)$

Output of $G_m = \text{output of circuit}$:
- Add clause $(g_m)$
Circuit to Boolean formula

Circuit $C_n$ has gates $G_1, \ldots, G_m$, where $m = poly(n)$.

Add variables $g_1, \ldots, g_m$, one for each gate.

Add a clause for each gate:

$$\begin{align*}
\text{OR:} & \quad a \quad b \quad g_i \quad \Rightarrow \quad ((a \lor b) \leftrightarrow g_i) \\
\text{AND:} & \quad a \quad b \quad g_i \quad \Rightarrow \quad ((a \land b) \leftrightarrow g_i) \\
\text{NOT:} & \quad a \quad g_i \quad \Rightarrow \quad (\neg a \leftrightarrow g_i)
\end{align*}$$

Boolean formula is the conjunction of all the clauses. Can reduce to CNF or 3SAT form.

Hard-code $x$:

- $x_i = 0 \rightarrow$ add clause $(\neg x_i)$
- $x_i = 1 \rightarrow$ add clause $(x_i)$

Output of $G_m =$ output of circuit:
Add clause $(g_m)$
The class \( \text{NP} \) and Turing Machine Tableaus

**NP**

A problem is in \( \text{NP} \) if there is a polynomial time algorithm \( A \) that takes two inputs, \( x \) and \( y \):

If \( x \in L \), then there is a witness \( y \) such that \( A(x, y) \) accepts.

If \( x \notin L \), then for every \( y \), \( A(x, y) \) rejects.

\[ |y| \leq \text{poly}(x) \]
The class NP and Turing Machine Tableaus

**NP**

A problem is in NP if there is a polynomial time algorithm $A$ that takes two inputs, $x$ and $y$:

If $x \in L$, then there is a witness $y$ such that $A(x, y)$ accepts.

If $x \notin L$, then for every $y$, $A(x, y)$ rejects.

$|y| \leq \text{poly}(x)$
Boolean Satisfiability is NP-hard

Start with a generic language $L$ in NP

Is $x \in L$?
Boolean Satisfiability is NP-hard

Start with a generic language $L$ in NP

Is $x \in L$?

Is there a string $y$ that causes this Turing Machine to accept?
Boolean Satisfiability is NP-hard

Start with a generic language $L$ in NP

Is $x \in L$?

$$(x, y) \xrightarrow{\text{Turing Machine } M} \text{Accept/Reject} \quad \implies \quad \text{Boolean Formula: } \Phi_x(y)$$

Is there a string $y$ that causes this Turing Machine to accept?
Boolean Satisfiability is NP-hard

Start with a generic language \( L \) in NP

Is \( x \in L \)?

\[ (x, y) \quad \text{Turing Machine} \quad M \quad \begin{array}{c} \text{Accept/} \\
\text{Reject} \end{array} \quad \Rightarrow \]

Boolean Formula: \( \Phi_x(y) \)

Is there a string \( y \) that causes this Turing Machine to accept?

\( \Leftrightarrow \quad \text{Is } \Phi_x(y) \text{ satisfiable?} \)
Turing Machine Tableau

\[(x, y) \rightarrow \text{Turing Machine } M \rightarrow \text{Accept/Reject}\]
Turing Machine Tableau

TM Tableau:
Configurations written in a 2D array
for a Turing Machine $M$ in input $(x, y)$

<table>
<thead>
<tr>
<th>$q_0/x_1$</th>
<th>$x_2$</th>
<th>$\cdots$</th>
<th>$y_m$</th>
<th>$\cdots$</th>
<th>$-$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>$q_1/x_2$</td>
<td>$\cdots$</td>
<td>$y_m$</td>
<td>$\cdots$</td>
<td>$-$</td>
</tr>
<tr>
<td>$q_3/a$</td>
<td>$b$</td>
<td>$\cdots$</td>
<td>$y_m$</td>
<td>$\cdots$</td>
<td>$-$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$q_A/-$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
<td>$\cdots$</td>
</tr>
</tbody>
</table>

Height: running time $= n^c$.

Width: space used $\leq n^c$.
Turing Machine to Boolean formula

Contents of a cell determined by the three cells above it.

<table>
<thead>
<tr>
<th>$q_1/a$</th>
<th>$b$</th>
<th>$c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_2/b$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$b$</th>
<th>$q_1/a$</th>
<th>$c$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$b$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$b$</th>
<th>$a$</th>
<th>$c$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Turing Machine to Boolean formula

Contents of a cell determined by the three cells above it.

Can build Boolean circuit STEP
- input (binary encoding of) 3 cells
- output (binary encoding of) 1 cell

Circuit is constant size
Turing Machine to Boolean formula

Contents of a cell determined by the three cells above it.

Can build Boolean circuit STEP

- input (binary encoding of) 3 cells
- output (binary encoding of) 1 cell

Circuit is constant size

Each circuit can be converted to a Boolean formula
(set of Boolean constraints)
Turing Machine to Boolean formula

\[
\begin{array}{cccccc}
q_0/x_1 & x_2 & \cdots & y_m & \cdots & - \\
\text{STEP} & \text{STEP} & \text{STEP} & \text{STEP} & \text{STEP} & \\
\text{STEP} & \text{STEP} & \text{STEP} & \text{STEP} & \text{STEP} & \\
\text{STEP} & \text{STEP} & \text{STEP} & \text{STEP} & \text{STEP} & \\
\end{array}
\]
Turing Machine to Boolean formula

Output 1 iff cell contains $q_{ACC}$
Turing Machine to Boolean formula

Features we will keep for QMA

Output 1 iff cell contains $q_{ACC}$
Turing Machine to Boolean formula

Features we will keep for QMA

Hard code input $x$ into circuit/constraints

Output 1 iff cell contains $q_{ACC}$
Turing Machine to Boolean formula

Features we will keep for QMA

- Hard code input $x$ into circuit/constraints
- Input $y$ (witness) is variable

Output 1 iff cell contains $q_{ACC}$
Turing Machine to Boolean formula

Features we will keep for QMA

- Hard code input $x$ into circuit/constraints
- Input $y$ (witness) is variable
- Satisfied constraints $\Rightarrow$ State represents entire history of the computation.
  (Configuration after each step)

Output 1 iff cell contains $q_{ACC}$
Turing Machine to Boolean formula

Features we will keep for QMA

- Hard code input $x$ into circuit/constraints
- Input $y$ (witness) is variable
- Satisfied constraints $\Rightarrow$
  - State represents entire history of the computation.
  - (Configuration after each step)
- Additional term to test if computation accepts

Output 1 iff cell contains $q_{ACC}$
Turing Machine to Boolean formula

<table>
<thead>
<tr>
<th>q₀/x₁</th>
<th>x₂</th>
<th>⋯</th>
<th>yₘ</th>
<th>⋯</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEP</td>
<td>STEP</td>
<td>STEP</td>
<td>STEP</td>
<td>STEP</td>
</tr>
</tbody>
</table>

Output 1 iff cell contains q_{ACC}

Features we will keep for QMA

- Hard code input x into circuit/constraints
- Input y (witness) is variable
- Satisfied constraints ⇒ State represents entire history of the computation.
  (Configuration after each step)
- Additional term to test if computation accepts
On to Part II...