Quantum Hamiltonian Complexity

Part I

Sandy Irani
Computer Science Department
UC Irvine

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 ↔ real eigenvalues)

N-dimensional quantum system:

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

(Assume for now non-degeneracy: λ_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. (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: λ_i's are distinct and there are N of them)

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

$$\lambda_0 \leftrightarrow |v_0\rangle$$
$$\lambda_1 \leftrightarrow |v_1\rangle$$
$$\ldots$$
$$\lambda_{N-1} \leftrightarrow |v_{N-1}\rangle$$

$|v_0\rangle, \ldots, |v_{N-1}\rangle$ 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: λ_i's are distinct and there are N of them)

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

$$\lambda_0 \leftrightarrow |v_0\rangle$$
$$\lambda_1 \leftrightarrow |v_1\rangle$$
$$\ldots$$
$$\lambda_{N-1} \leftrightarrow |v_{N-1}\rangle$$

$|v_0\rangle, \ldots, |v_{N-1}\rangle$ 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 λ_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 λ_i is:

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

Expected outcome is:

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

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 λ_i is:

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

Expected outcome is:

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

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

Measurement, cont.

State: $|\Phi\rangle$

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

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

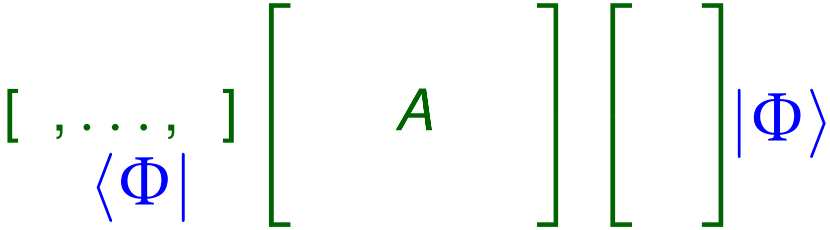
Probability of outcome λ_i is:

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

Expected outcome is:

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

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



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

$$|\psi(0)\rangle \longrightarrow \boxed{i\hbar \frac{d|\psi\rangle}{dt} = H|\psi\rangle.} \longrightarrow |\psi(t)\rangle$$

$$= e^{-iHt}|\psi(0)\rangle$$

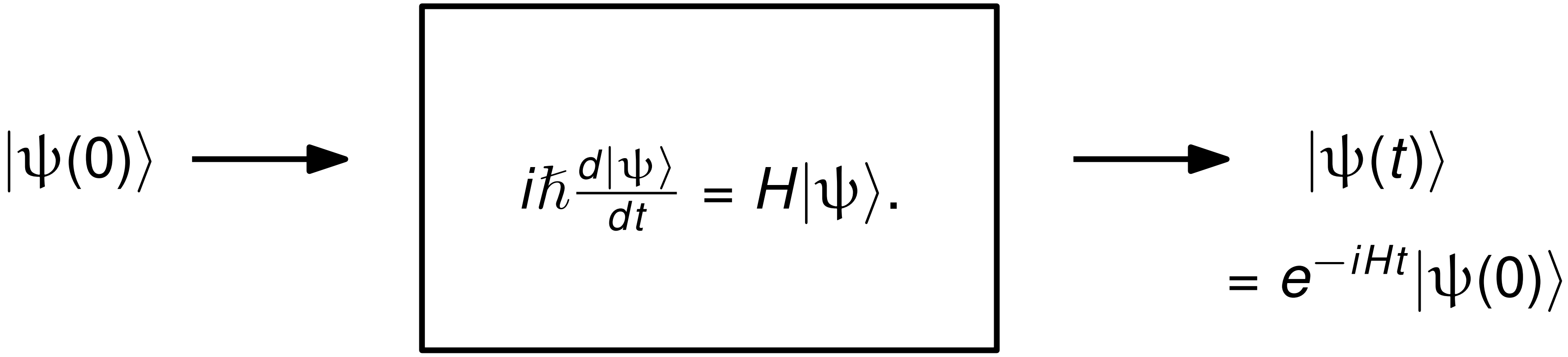
The Hamiltonian Operator - dynamics

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



$$|\psi(0)\rangle \longrightarrow \boxed{i\hbar \frac{d|\psi\rangle}{dt} = H|\psi\rangle.} \longrightarrow |\psi(t)\rangle$$

$$= e^{-iHt}|\psi(0)\rangle$$

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 β 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.

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

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

Parameter β scales inversely with temperature

Z is called the *partition function*

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

[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.

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

(assuming a unique ground state)

The Hamiltonian Operator - the ground state

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

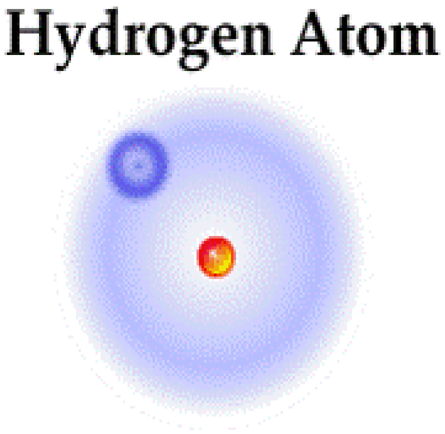
$$\lim_{\beta \to \infty} \rho_{eq} = \lim_{\beta \to \infty} \sum_i \frac{e^{-\beta E_i}}{Z} |v_i\rangle\langle v_i| = |v_0\rangle\langle v_0|$$

(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 $|v_0\rangle$

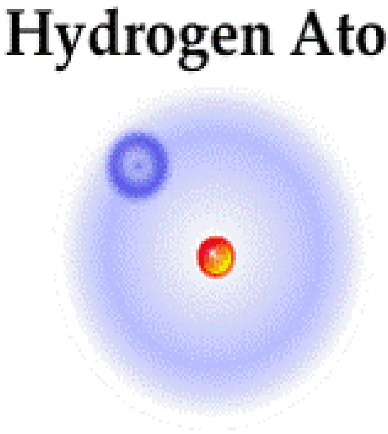
An Example of a Quantum System and Its Hamiltonian



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



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

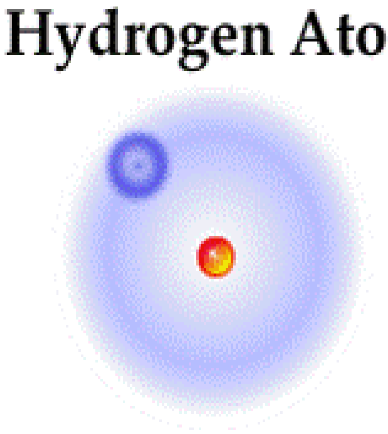
$$\psi(r, \theta, \phi)$$

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

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

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

An Example of a Quantum System and Its Hamiltonian



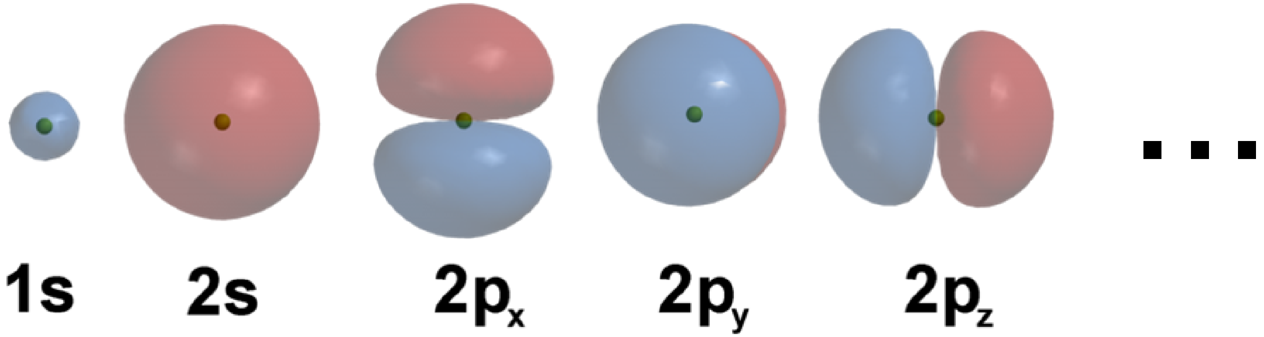
The ”state” is the position of the electron relative to the proton:

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

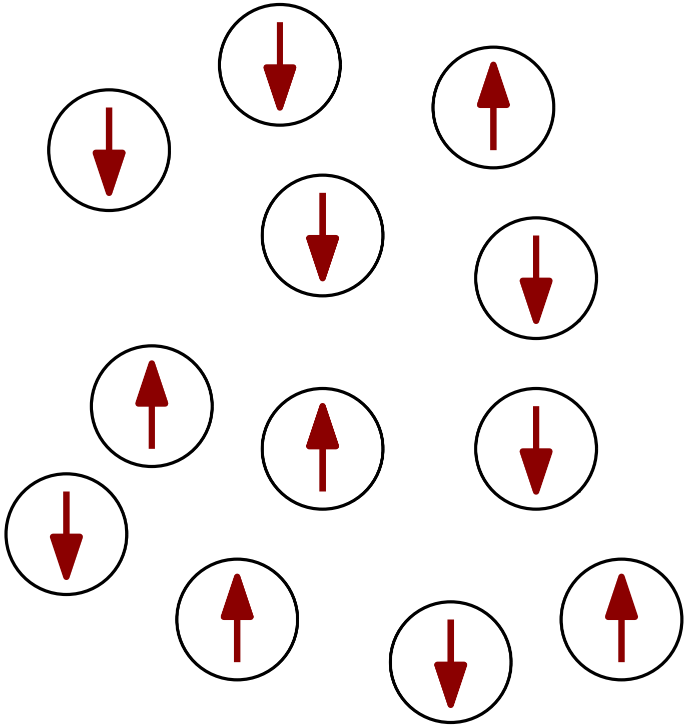
The Hamiltonian describes the energy as a function of the electron location:

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

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

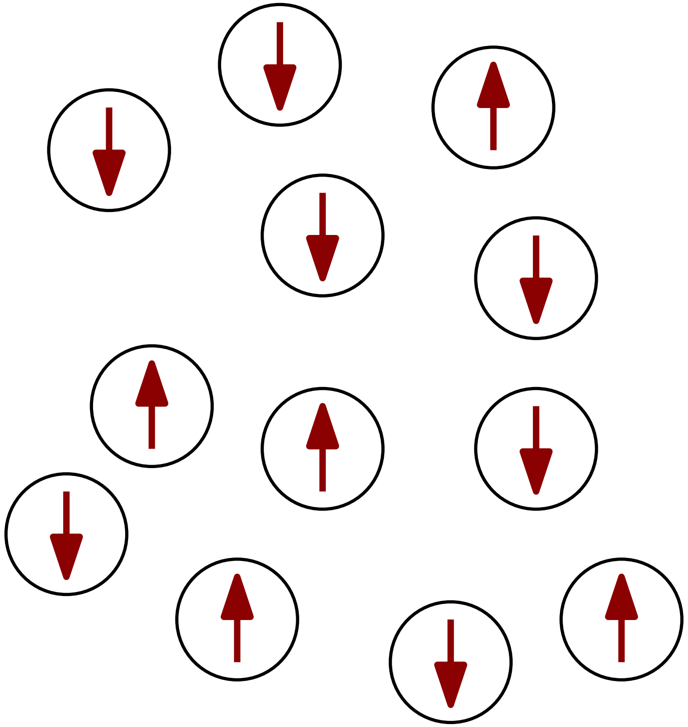


Local Hamiltonians



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

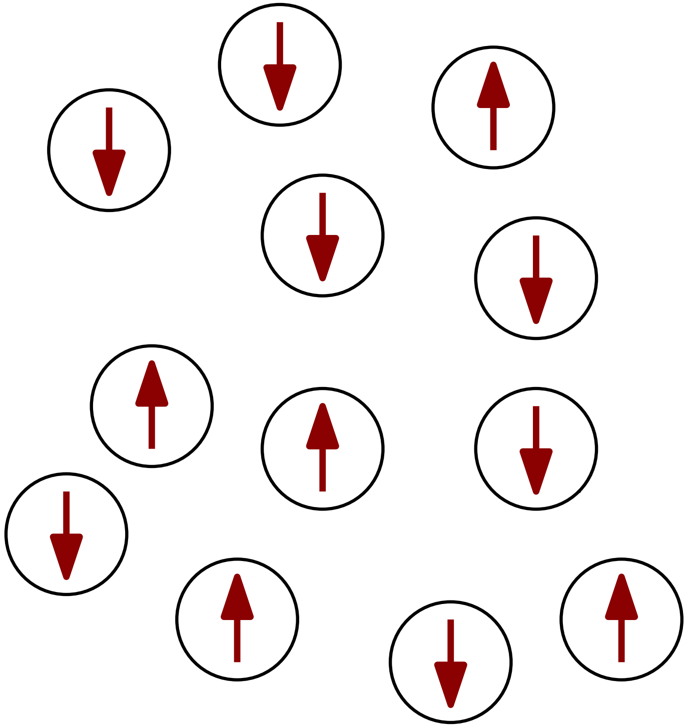
Local Hamiltonians



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.

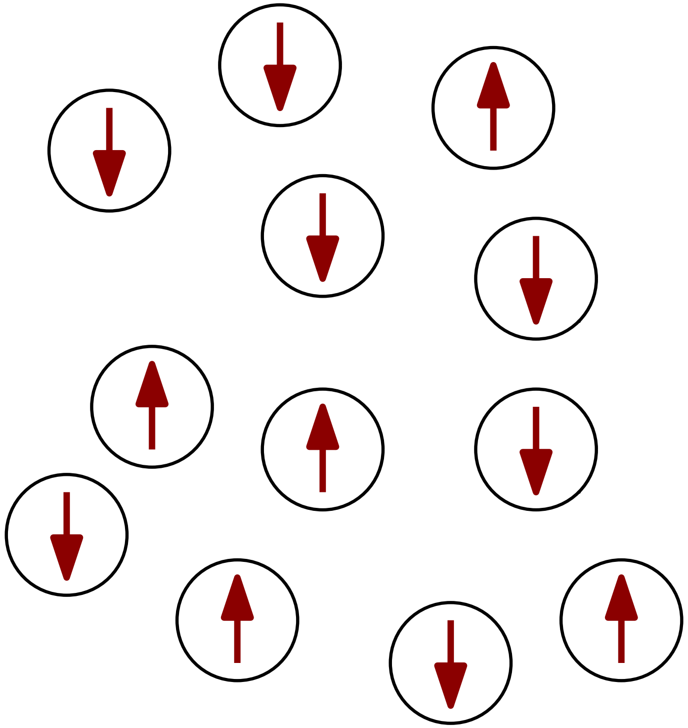
Hilbert space for a particle: $\mathbb{C}^d$

Hilbert space for the whole system:

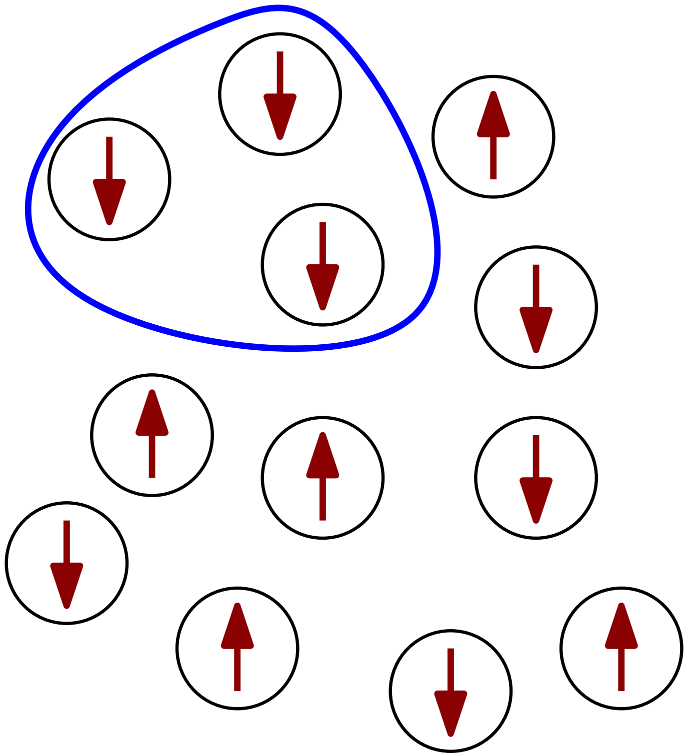
$$\left(\mathbb{C}^d\right)^{\otimes n}$$

Dimension = d^n

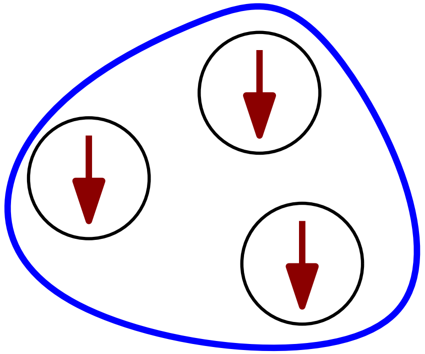
Local Hamiltonians



Local Hamiltonians

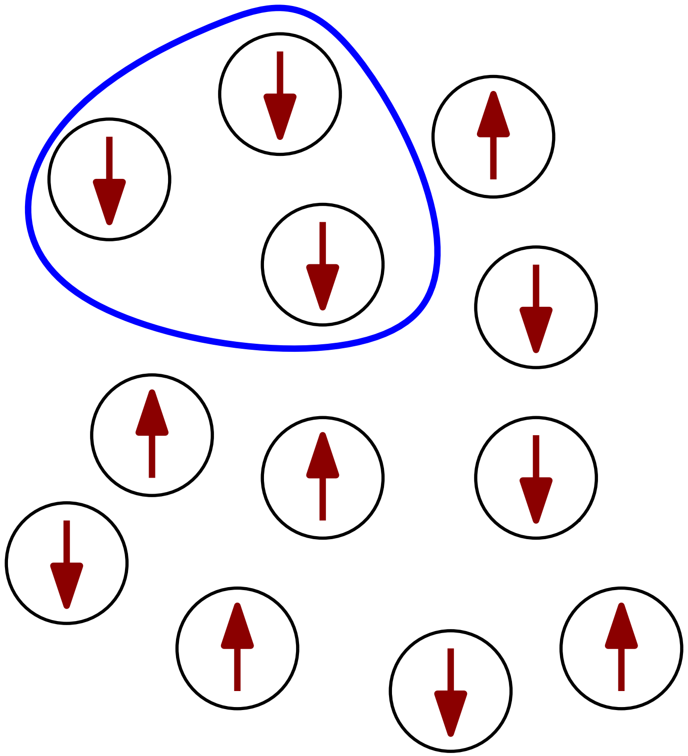


Local Hamiltonians



The Hamiltonian for a 3-qubit system is an 8 × 8 matrix $H_{1,2,3}$.

Local Hamiltonians



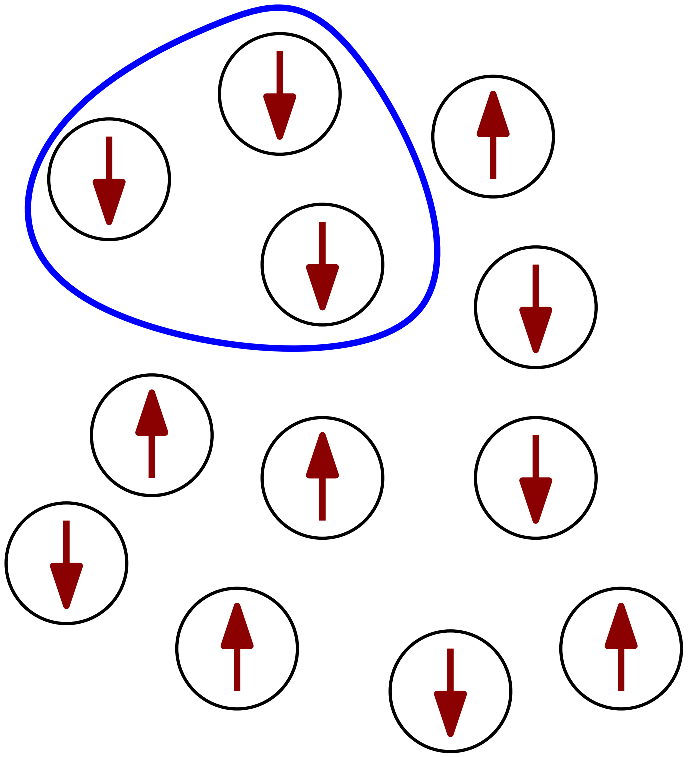
The Hamiltonian for a 3-qubit system is an 8×8 matrix $H_{1,2,3}$.

The interaction between 3 qubits in an n-qubit system is $H_{1,2,3} \otimes I_{4,\ldots,n}$.

Local Hamiltonians

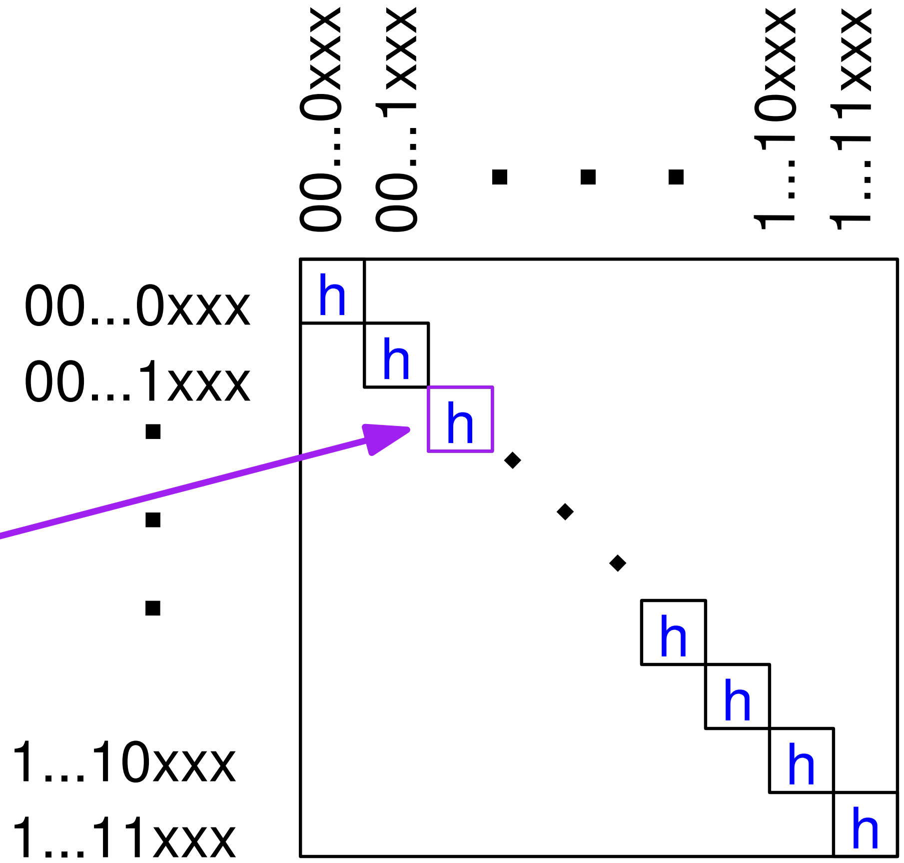
The Hamiltonian for a 3-qubit system is an 8×8 matrix $H_{1,2,3}$.

The interaction between 3 qubits in an n-qubit system is $H_{1,2,3} \otimes I_{4,\ldots,n}$.

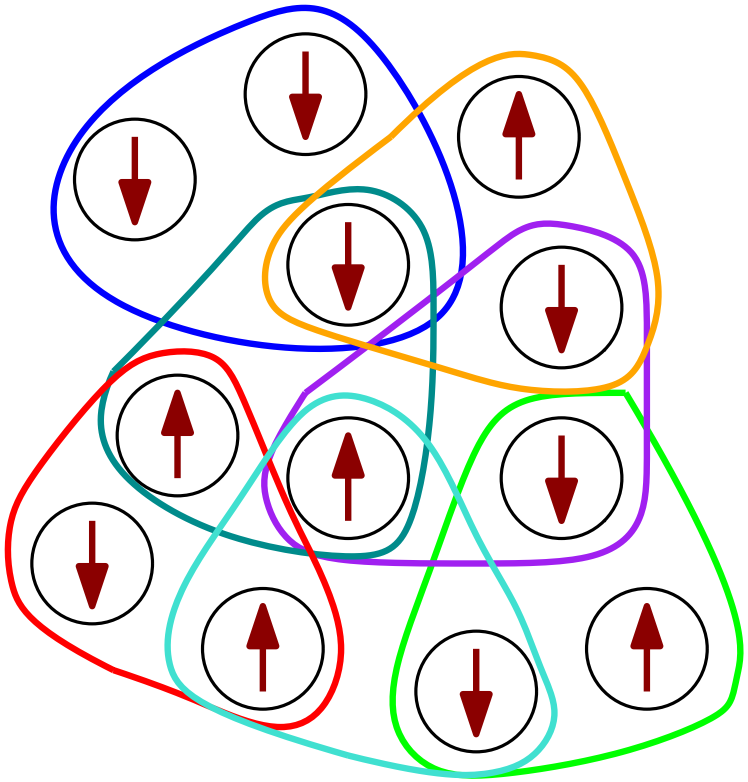


$h = H_{n-2,n-1,n}$

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



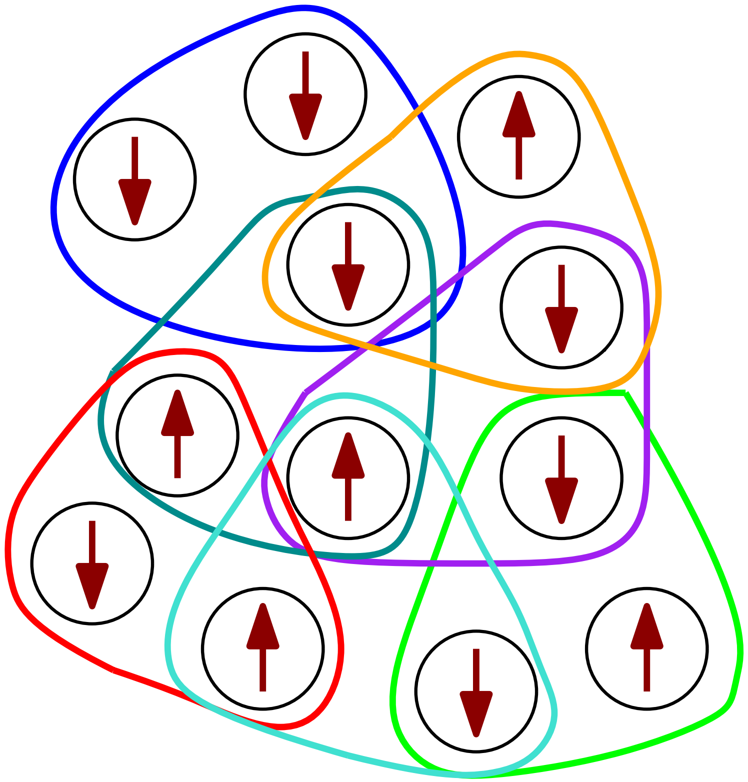
Local Hamiltonians



$H = \sum_a H_a$

where each H_a acts on at most k qudits

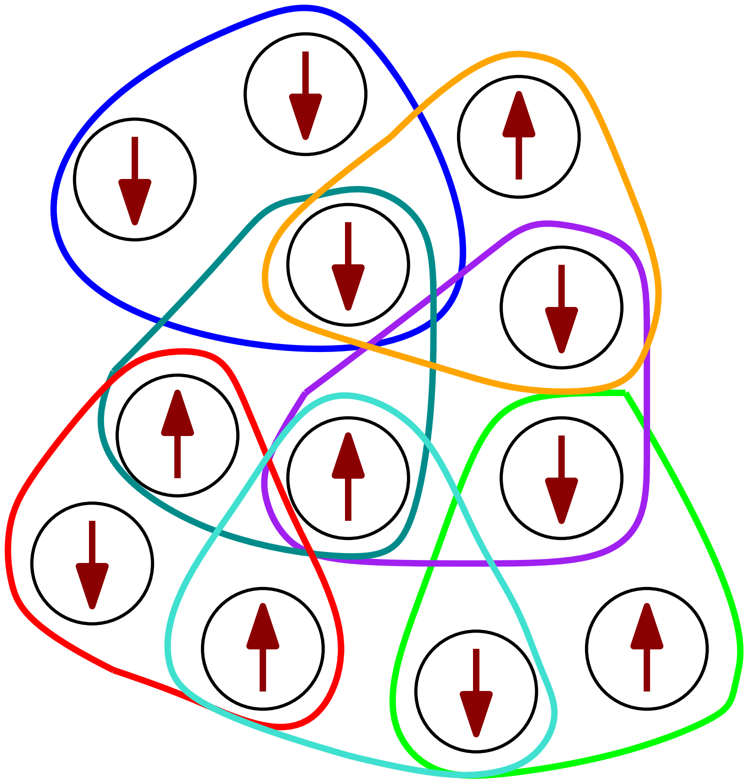
Local Hamiltonians



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

System consists of n d-dimensional particles

Local Hamiltonians

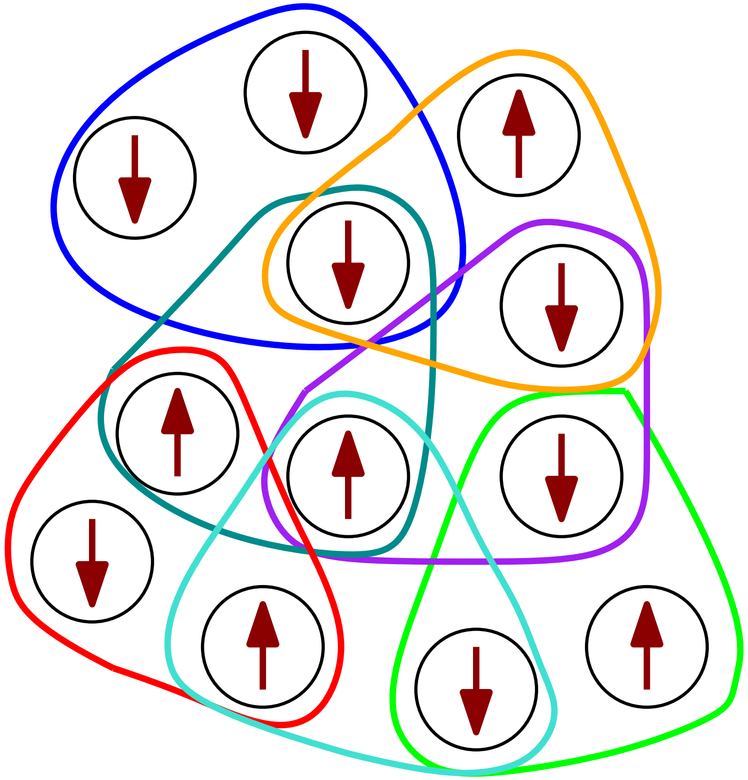


$H = \sum_a H_a$
where each H_a acts on at most k qudits

System consists of n d-dimensional particles

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

Local Hamiltonians



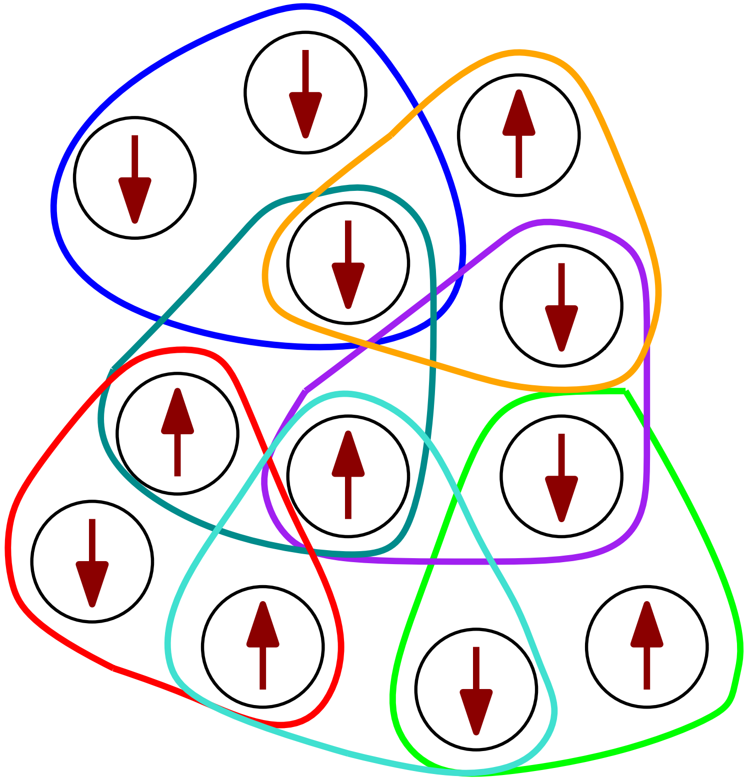
$H = \sum_a H_a$
where each H_a acts on at most k qudits

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

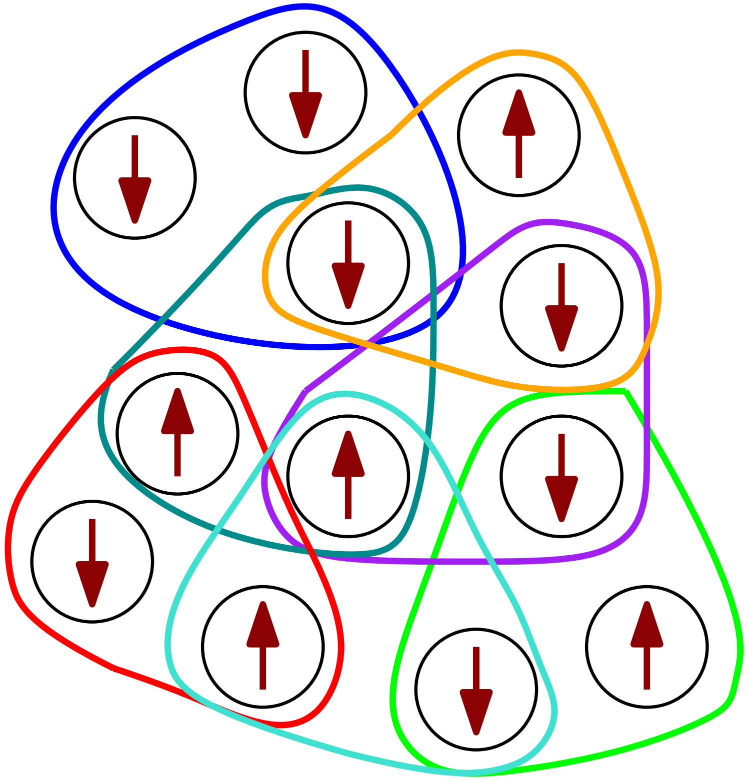
System consists of n d-dimensional particles

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

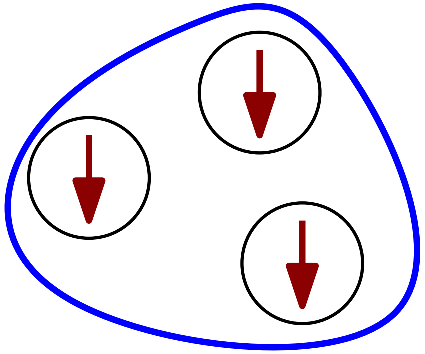
System consists of n d-dimensional particles

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 Δ
Is the ground energy of $H \leq E$ or $\geq E + \Delta$?

Local Hamiltonian Variations

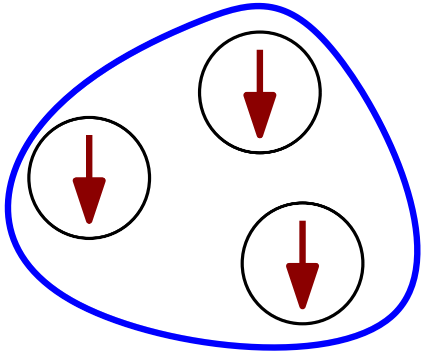


Locality

$H = \sum_a H_a$

where each H_a acts on at most k qudits

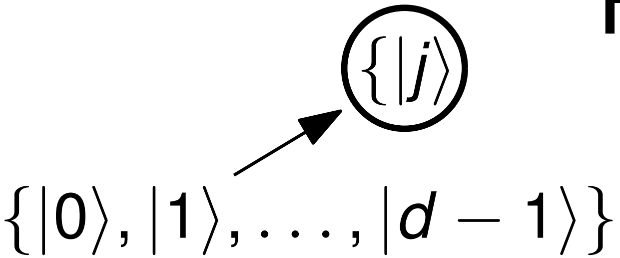
Local Hamiltonian Variations



Locality

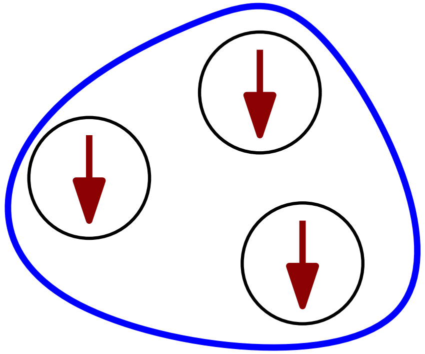
$H = \sum_a H_a$
where each H_a acts on at most k qudits

Particle Dimension



$\{|0\rangle, |1\rangle, \ldots, |d-1\rangle\}$

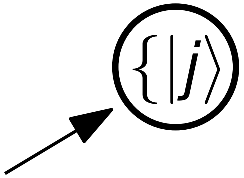
Local Hamiltonian Variations



Locality

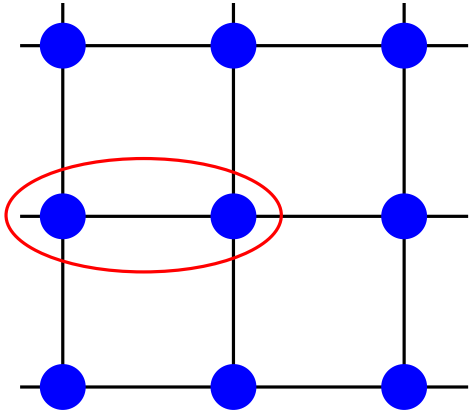
$H = \sum_a H_a$
where each H_a acts on at most k qudits

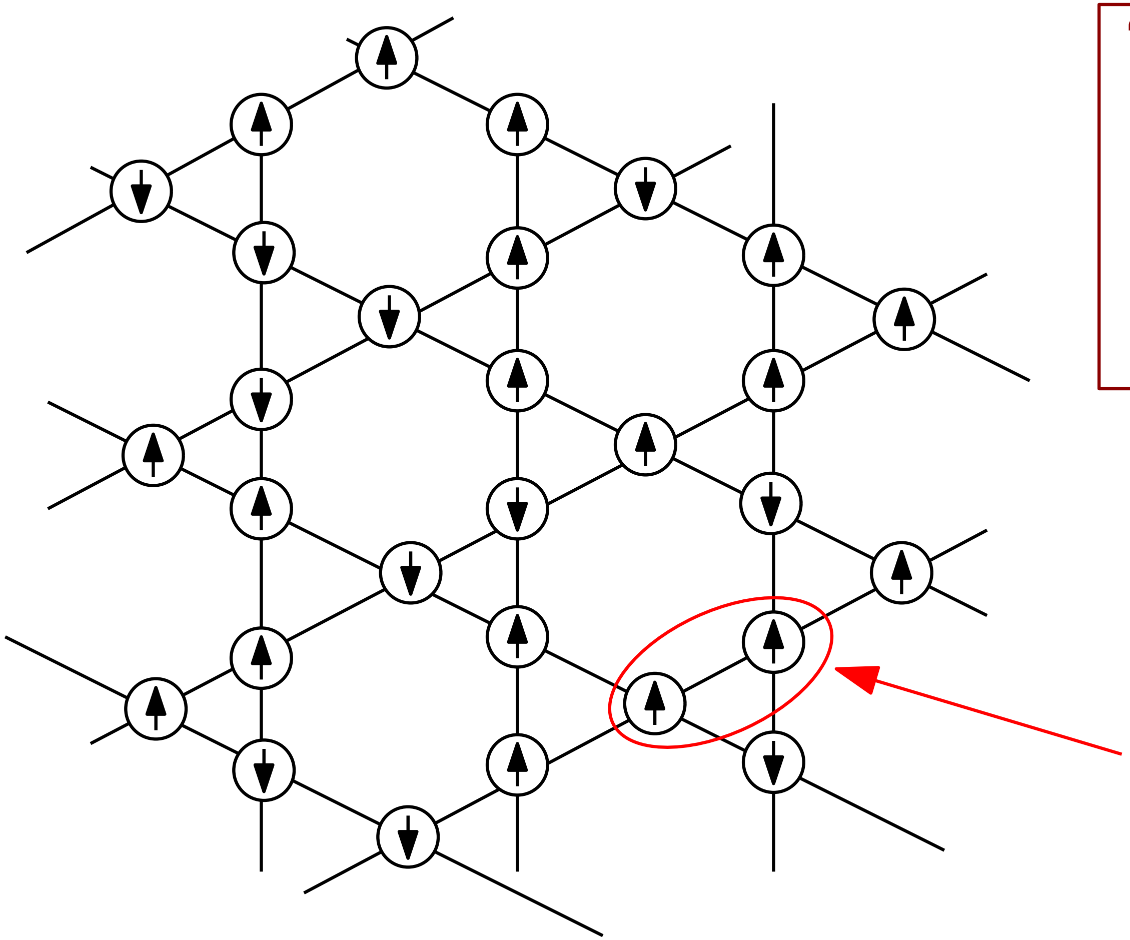
Particle Dimension



$\{|0\rangle, |1\rangle, \ldots, |d-1\rangle\}$

Geometry





Kagome Lattice

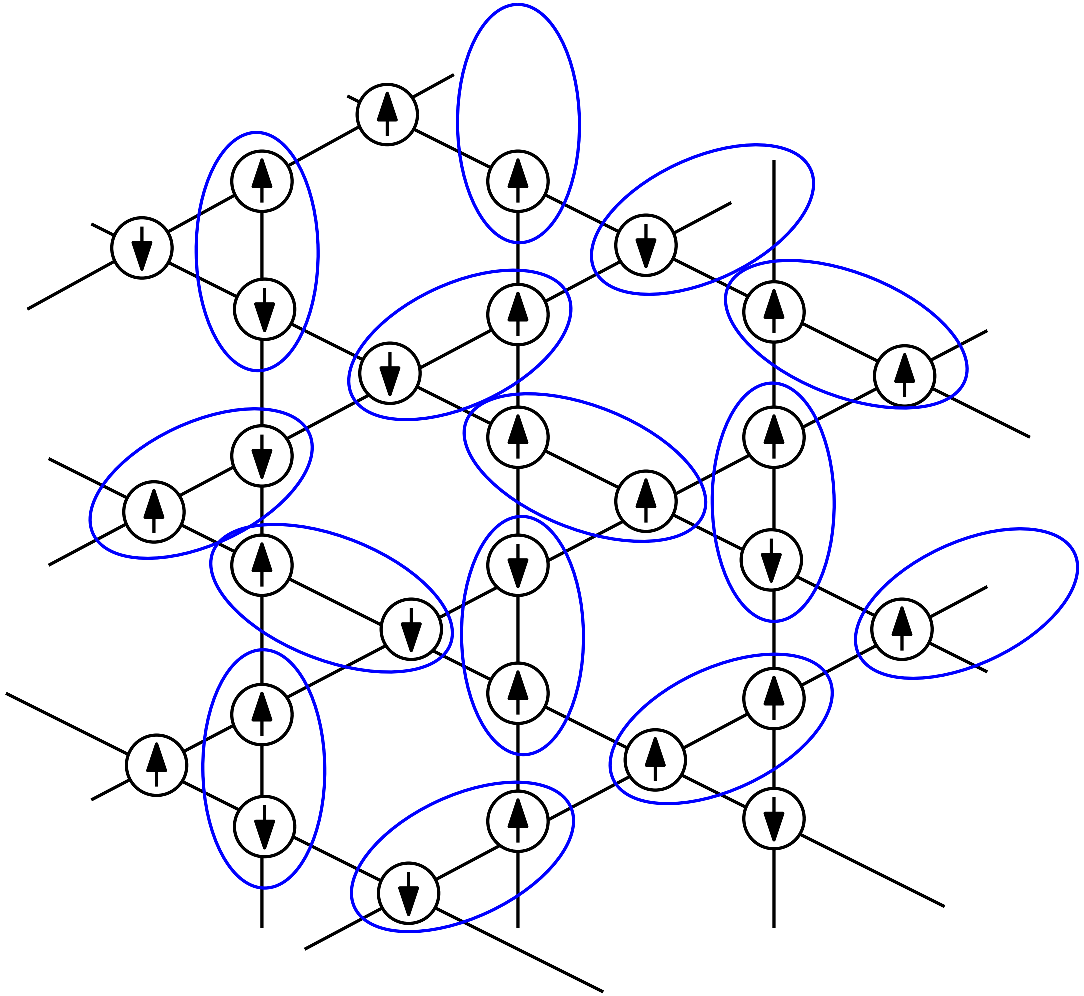
“Spin-Liquid Ground State of the $S = \frac{1}{2}$ Kagome Heisenberg Antiferromagnet”

Yan, Huse, White

Science, Vol 332, June 3, 2011

$$\begin{array}{c|cccc} & \mathbf{00} & \mathbf{01} & \mathbf{10} & \mathbf{11} \\ \hline \mathbf{00} & 1 & 0 & 0 & 0 \\ \mathbf{01} & 0 & -1 & -2 & 0 \\ \mathbf{10} & 0 & -2 & -1 & 0 \\ \mathbf{11} & 0 & 0 & 0 & 1 \end{array}$$

Heisenberg
- Antiferromagnet
Model



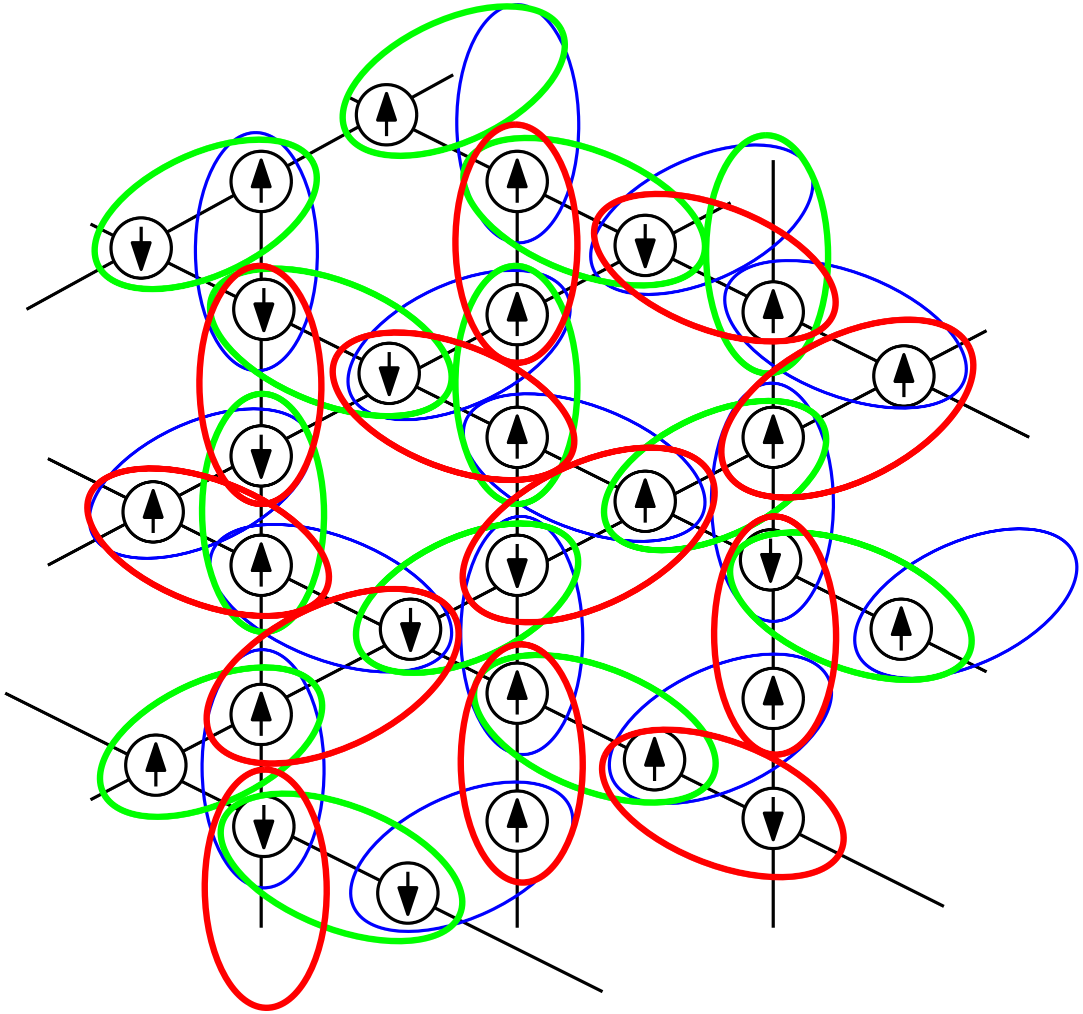
Kagome Lattice

“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

“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?

or a Spin Liquid?

“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: $\mathcal{H} = \left(\mathbb{C}^d\right)^{\otimes n}$

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

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

”Classical” Local Hamiltonian

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

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.

$$\begin{bmatrix} * & & & & & & & \\ & * & & & & & & \\ & & * & & & & & \\ & & & * & & & & \\ & & & & * & & & \\ & & & & & * & & \\ & & & & & & * & \\ & & & & & & & * \end{bmatrix}$$

”Classical” Local Hamiltonian

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

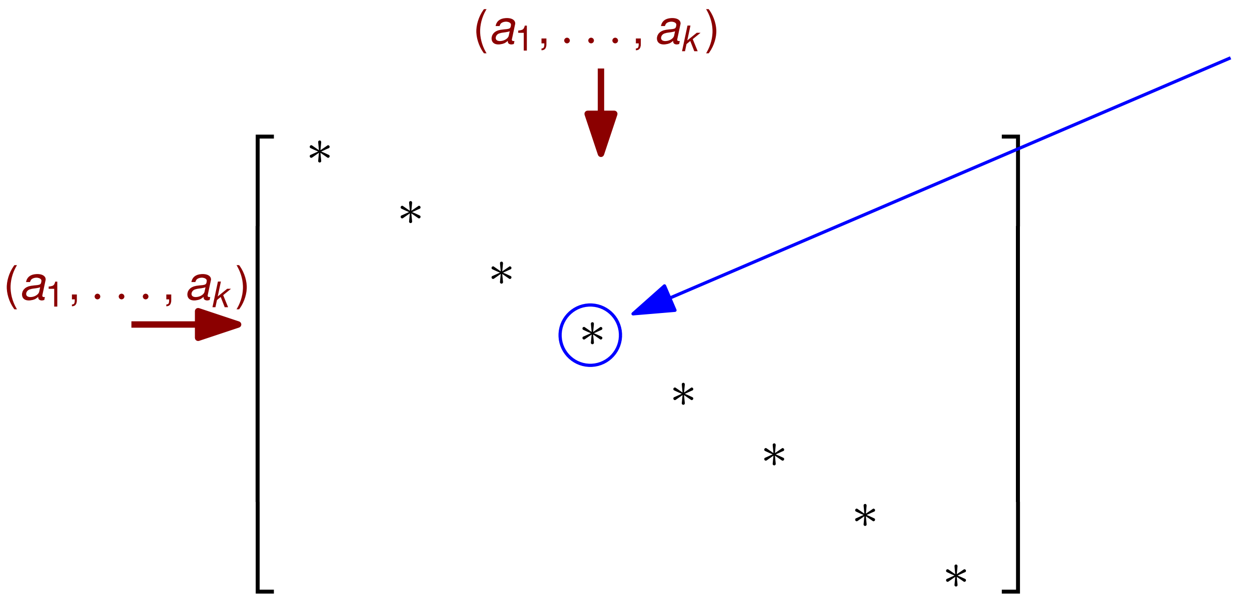
Standard basis denoted by classical strings: $|x_1, x_2, \ldots, x_n\rangle$

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.

H_j operates on particles $i_1, i_2, \ldots, i_k$



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

”Classical” Local Hamiltonian

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

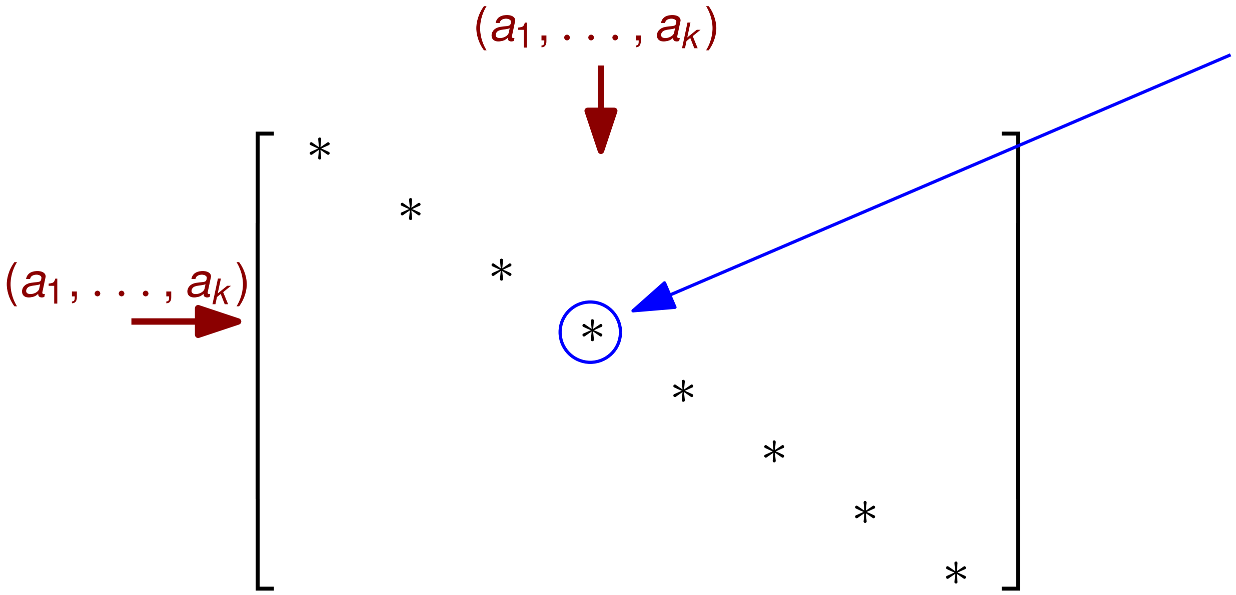
Standard basis denoted by classical strings: $|x_1, x_2, \ldots, x_n\rangle$

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.

H_j operates on particles $i_1, i_2, \ldots, i_k$



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: $\mathcal{H} = \left(\mathbb{C}^d\right)^{\otimes n}$

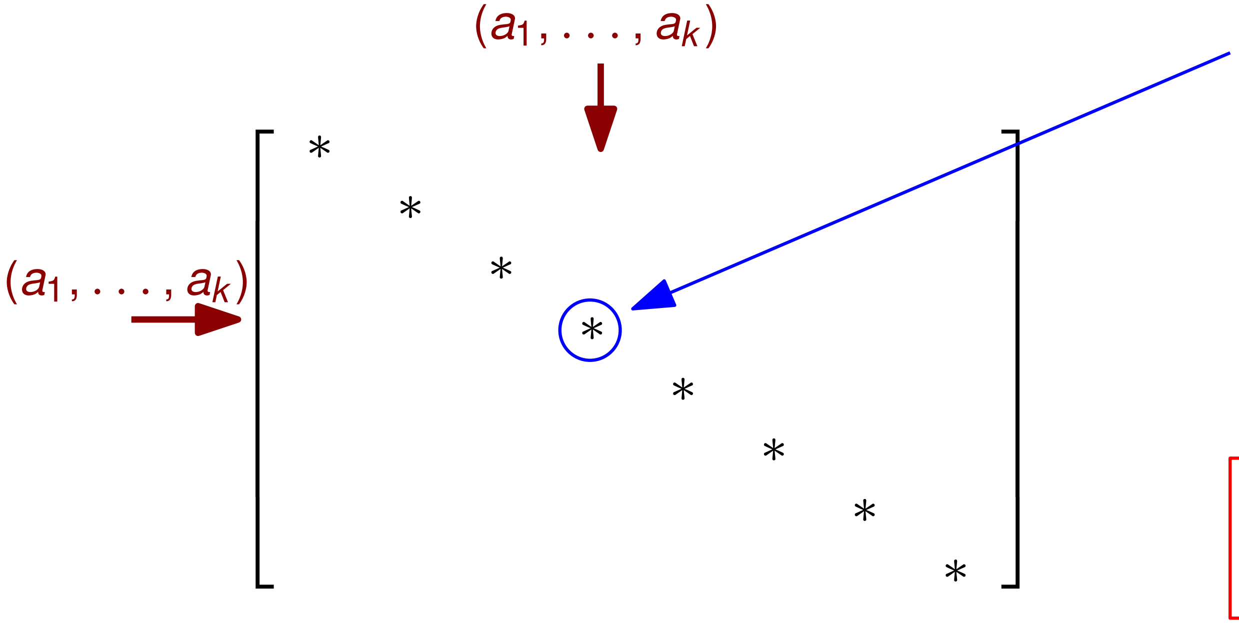
Standard basis denoted by classical strings: $|x_1, x_2, \ldots, x_n\rangle$

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.

H_j operates on particles $i_1, i_2, \ldots, i_k$



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., $(x_{i1} \vee \neg x_{i2} \vee x_{i3})$

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

$$C_1 \wedge C_2 \wedge \cdots \wedge C_m = 1\ ?$$

Local Hamiltonian is NP-hard

$\text{3SAT} \propto \text{LH}$

Local Hamiltonian is NP-hard

$$\begin{array}{c|cccccccc}
 & & & 010 \downarrow & & & & & \\
\hline
 & 0 & & & & & & & \\
 & & 0 & & & & & & \\
010 \rightarrow & & & 1 & & & & & \\
 & & & & 0 & & & & \\
 & & & & & 0 & & & \\
 & & & & & & 0 & & \\
 & & & & & & & 0 & \\
 & & & & & & & & 0
\end{array}$$

3SAT $\propto$ LH

$$\Leftrightarrow (x \vee \neg y \vee z)$$

Local Hamiltonian is NP-hard

3SAT $\propto$ LH

$$|010\rangle\langle 010| = \begin{array}{r c} & \begin{array}{cccccccc} & & 010 & & & & & \\ & & \downarrow & & & & & \end{array} \\ 010 \rightarrow & \begin{bmatrix} 0 & & & & & & & \\ & 0 & & & & & & \\ & & 1 & & & & & \\ & & & 0 & & & & \\ & & & & 0 & & & \\ & & & & & 0 & & \\ & & & & & & 0 & \\ & & & & & & & 0 \end{bmatrix} \end{array} \Leftrightarrow (x \vee \neg y \vee z)$$

Local Hamiltonian is NP-hard

$$|010\rangle\langle 010| = \begin{array}{cc} & \begin{array}{cccccccc} & & \overset{010}{\downarrow} & & & & & \end{array} \\ 010 \rightarrow & \begin{bmatrix} 0 & & & & & & & \\ & 0 & & & & & & \\ & & 1 & & & & & \\ & & & 0 & & & & \\ & & & & 0 & & & \\ & & & & & 0 & & \\ & & & & & & 0 & \\ & & & & & & & 0 \end{bmatrix} \end{array}$$

$$\Leftrightarrow (x \vee \neg y \vee z)$$

$3SAT \propto LH$

$$H = \sum_j H_{C_j} \quad \Leftrightarrow \quad C_1 \wedge C_2 \wedge \cdots \wedge C_m$$

H has a zero energy ground state $\Leftrightarrow$ $C_1 \wedge C_2 \wedge \cdots \wedge C_m$ 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

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

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.

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

$\text{SAT} \in \text{NP}$

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 $\boxed{\text{polynomial time algorithm } A}$ 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.

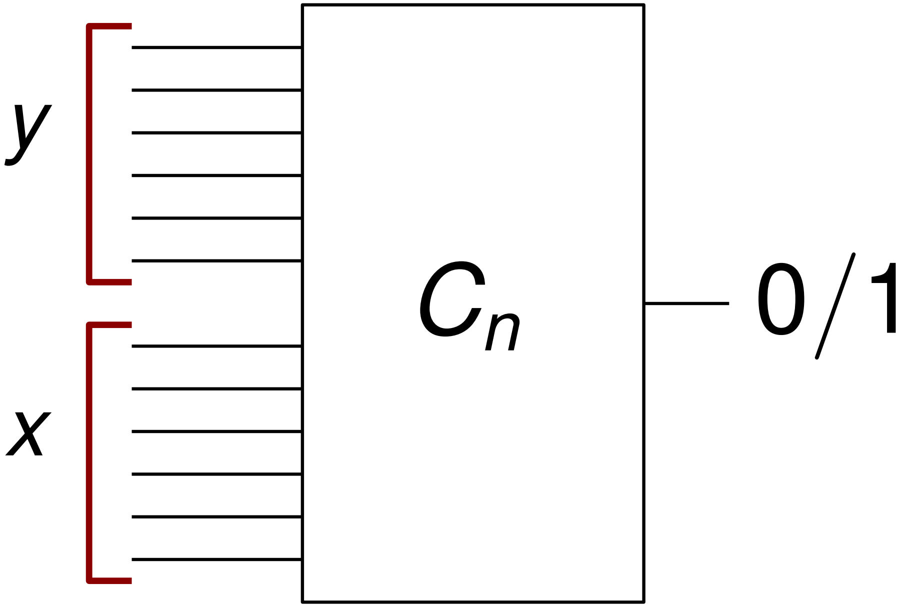
$|y| \leq \text{poly}(x)$

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

If $|x| = n$, then

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

$A(x, y)$ rejects $\leftrightarrow C_n(x, y) = 0$



The class NP

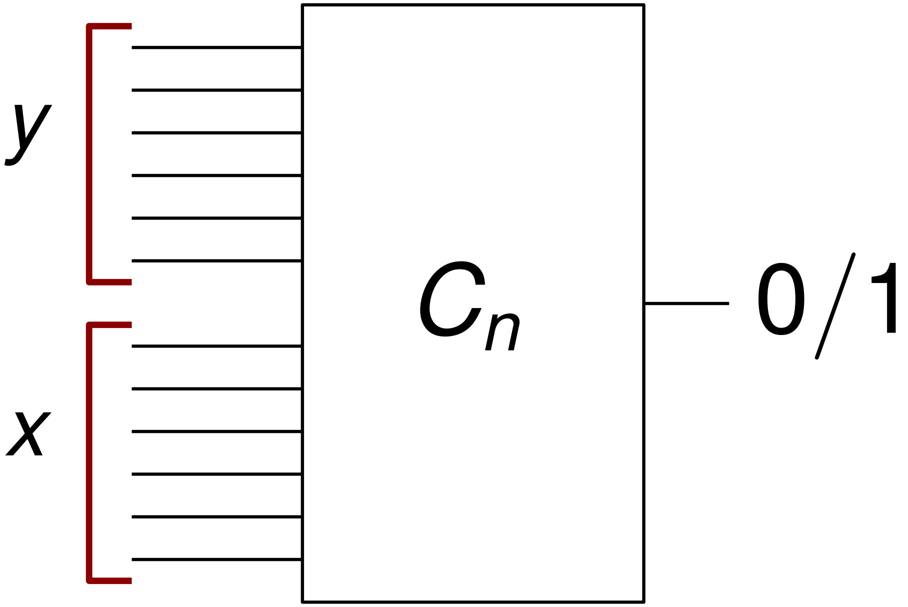
NP

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

Poly-sized circuit family $\{C_n\}$
If $|x| = n$, then
$A(x, y)$ accepts $\leftrightarrow C_n(x, y) = 1$
$A(x, y)$ rejects $\leftrightarrow C_n(x, y) = 0$

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”

$L \subseteq \{0, 1\}^*$

$x \in L \Rightarrow$ ”Yes”

$x \notin L \Rightarrow$ ”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)

MA

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

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

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

If $\underline{x \in \text{No}}$, 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 $\underline{x \in \text{Yes}}$, then there is a witness y such that $R(x, y)$ accepts with prob $\geq \frac{2}{3}$.

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

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

The class MA (Merlin-Arthur)

MA

R(x,y):

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

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

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

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

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

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

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

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

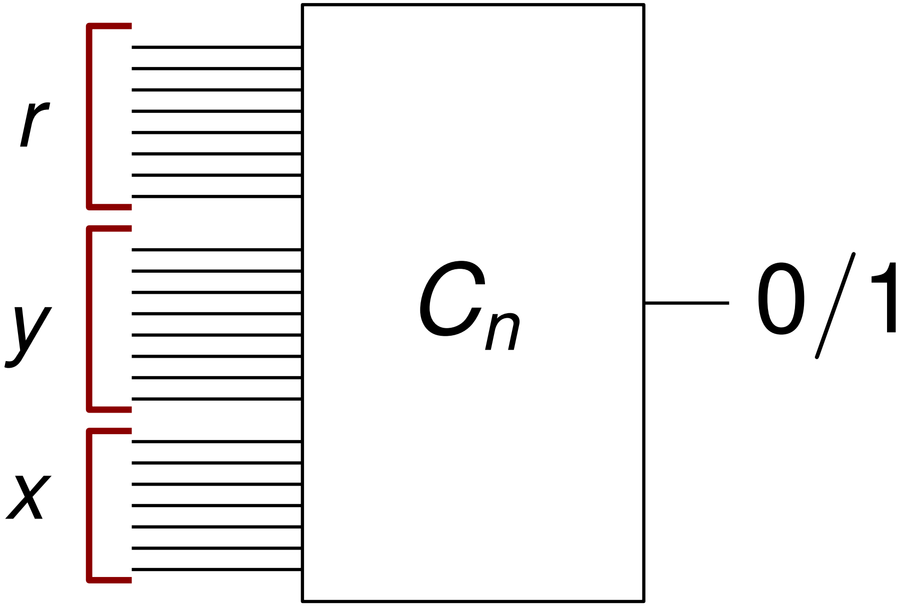
$|y| \leq \text{poly}(|x|)$

R(x,y):

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

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

$x \in \text{No} \leftrightarrow \forall y$
$\text{Prob}_r[C_n(x, y, r) = 1] \geq \frac{1}{3}$



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

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

If $\underline{x \in \text{No}}$, 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 $\text{Prob}[C_n(x, |\phi\rangle) = 1] \geq 2/3$.

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

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

The class QMA (Quantum Merlin Arthur)

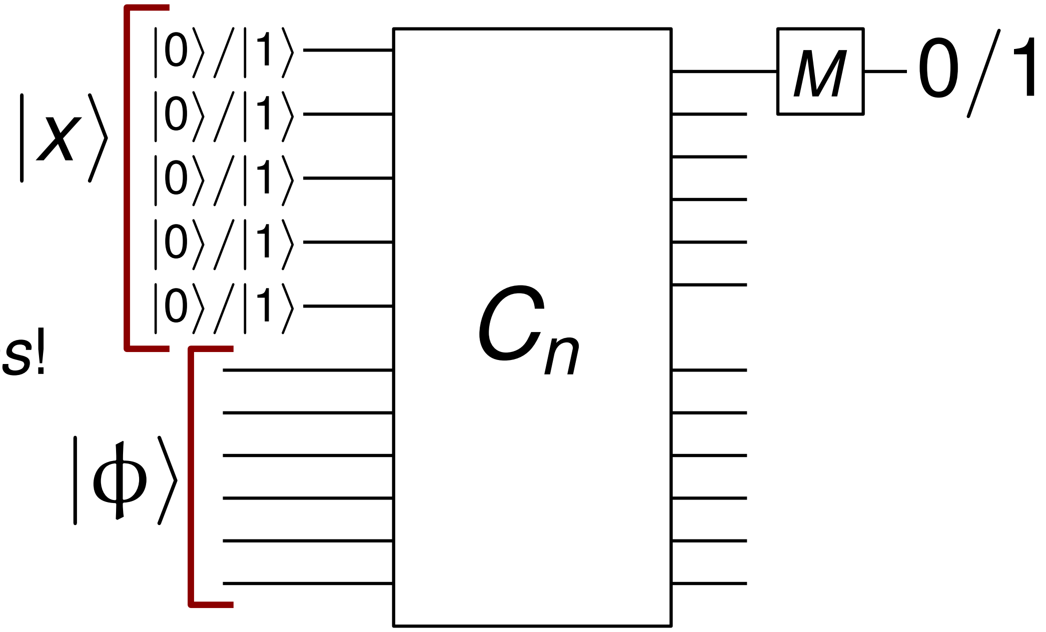
QMA

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

If $x \in$ No, then for every $|\phi\rangle$, Prob$[C_n(x, |\phi\rangle) = 1] \leq 1/3$.

If $x \in$ Invalid, then *no guarantees*!



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

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*:

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

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

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

$|x| = n$ 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 $x \in$ Yes, then there is a witness y such that
$R(x, y)$ accepts with prob $\geq$ **c**

If $x \in$ 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)

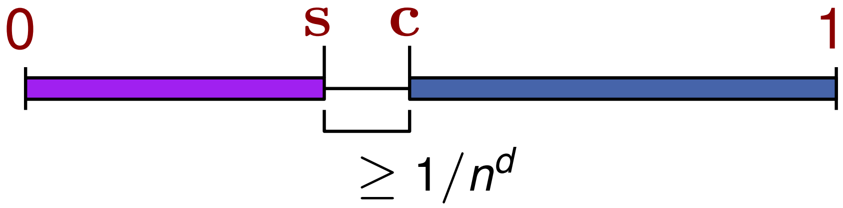
A *promise* problem is in NP if there is a polynomial time *randomized* algorithm R that takes two inputs, x and y:

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

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

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

$$\boxed{\begin{array}{c}\text{If } c - s \geq \frac{1}{n^d}, \quad \text{then} \\[2em] \text{MA}(c, s) = \text{MA}\left(1 - \frac{1}{2^n}, \frac{1}{2^n}\right)\end{array}}$$



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:

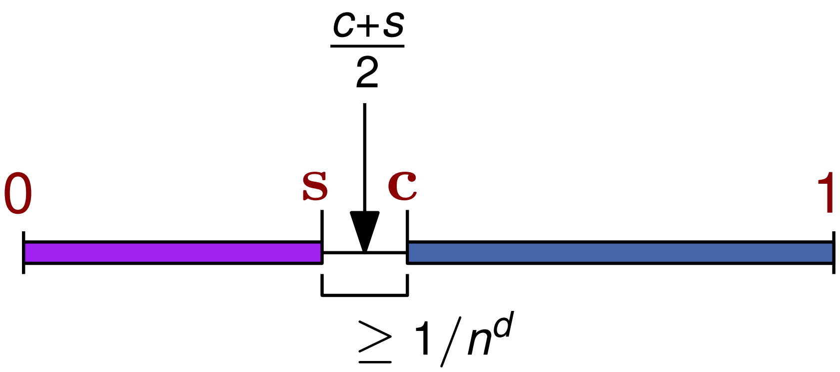
If $x \in$ Yes, then there is a witness y such that
$R(x, y)$ accepts with prob $\geq$ **c**

If $x \in$ 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:

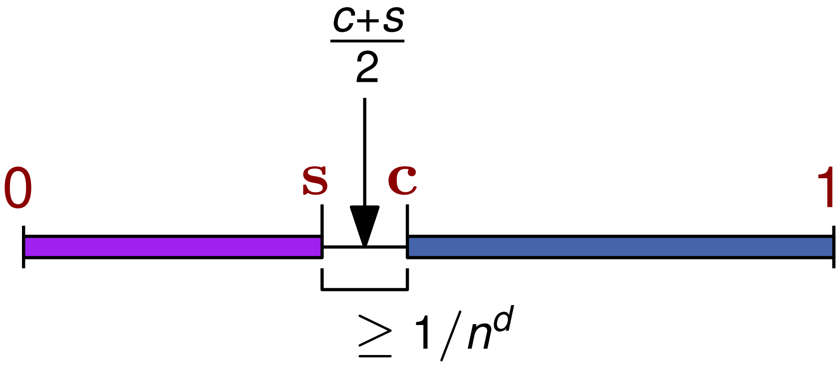
If $x \in \text{Yes}$, then there is a witness y such that $R(x, y)$ accepts with prob $\geq$ **c**

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 $c - s \geq \frac{1}{n^d}$, 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 = $\left(\frac{c+s}{2}\right) m$

By Chernoff's Inequality
For m = 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 $\underline{x \in \text{Yes}}$, then there is a
quantum witness $|\phi\rangle$ such that
Prob$[C_n(x, |\phi\rangle) = 1] \geq \mathbf{c}$

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

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

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

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

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

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

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

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

Completeness:

$$\underbrace{\boxed{|\Phi\rangle}\boxed{|\Phi\rangle}\boxed{|\Phi\rangle}\quad\cdots\quad\boxed{|\Phi\rangle}}_{m \text{ independent copies}}$$

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

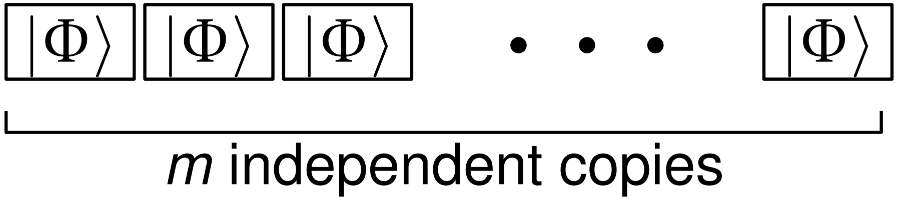
If $\underline{x \in \text{No}}$, then for every $|\phi\rangle$, $\text{Prob}[C_n(x, |\phi\rangle) = 1] \leq \mathbf{s}$

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

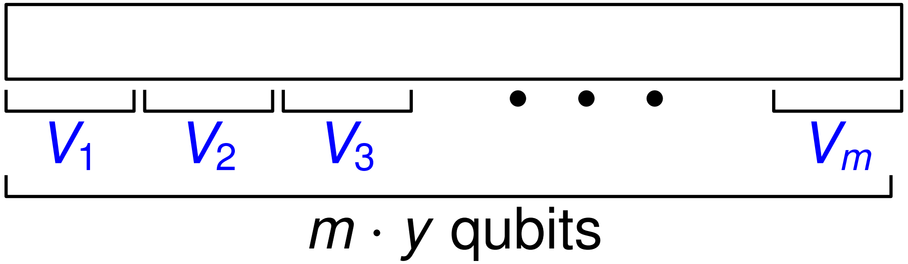
If $c - s \geq \frac{1}{n^d}$, then

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

Completeness:



Soundness:



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

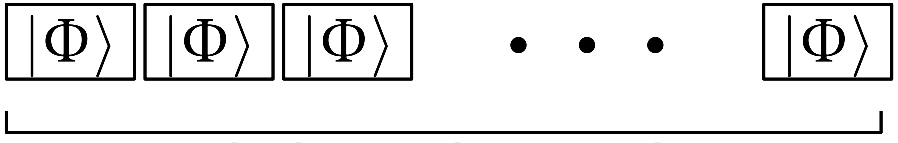
If $x \in$ No, then for every $|\phi\rangle$, Prob$[C_n(x, |\phi\rangle) = 1] \leq \mathbf{s}$

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

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

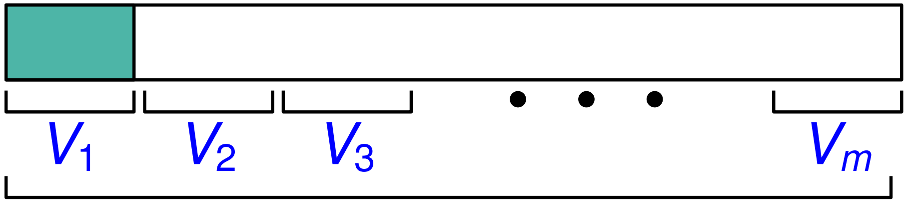
$$\text{QMA}(c, s) = \text{QMA}\left(1 - \frac{1}{2^n}, \frac{1}{2^n}\right)$$

Completeness:



m independent copies

Soundness:



$m \cdot y$ qubits

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

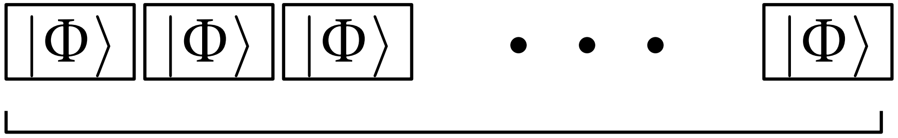
If $\underline{x \in \text{No}}$, then for every $|\phi\rangle$, $\text{Prob}[C_n(x, |\phi\rangle) = 1] \leq \mathbf{s}$

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

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

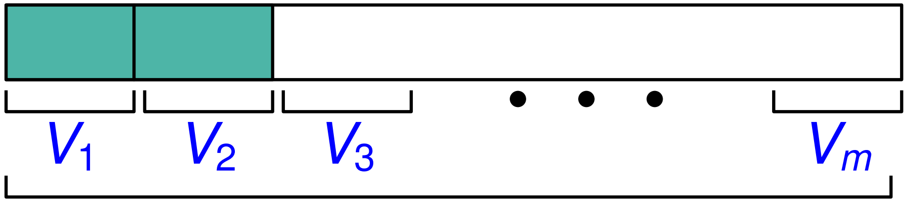
$\text{QMA}(c, s) = \text{QMA}\left(1 - \frac{1}{2^n}, \frac{1}{2^n}\right)$

Completeness:



m independent copies

Soundness:



$m \cdot y$ qubits

Prob ACC $\leq s$

Prob ACC $\leq s$

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

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

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

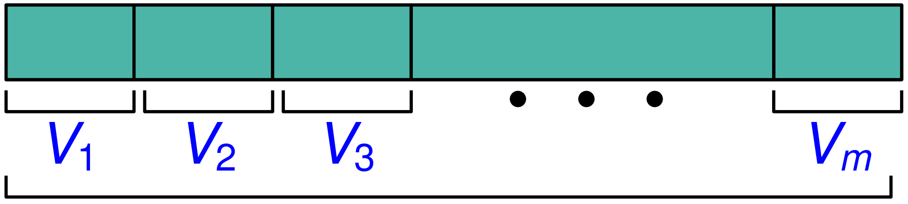
If $c - s \geq \frac{1}{n^d}$, then

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

Completeness:

$$\underbrace{\boxed{|\Phi\rangle}\boxed{|\Phi\rangle}\boxed{|\Phi\rangle}\quad \cdots \quad \boxed{|\Phi\rangle}}_{m \text{ independent copies}}$$

Soundness:



$m \cdot y$ qubits

Prob ACC $\leq s$

Prob ACC $\leq s$

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

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

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

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

$|\phi\rangle$ 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”

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

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

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

$|\phi\rangle$ 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

Probabilistically try and back up after a measurement.

Measure for a successful back up.

Principle of deferred measurements.

Complexity Classes and Complete Problems

NP $\subseteq$ MA $\subseteq$ QMA

Complexity Classes and Complete Problems

$$\mathrm{NP} \subseteq \mathrm{MA} \subseteq \mathrm{QMA} \subseteq \mathrm{PP} \subseteq \mathrm{PSPACE}$$

Complexity Classes and Complete Problems

Boolean satisfiability
is complete for NP
[Cook-Levin]

$$\mathrm{NP} \subseteq \mathrm{MA} \subseteq \mathrm{QMA} \subseteq \mathrm{PP} \subseteq \mathrm{PSPACE}$$

Complexity Classes and Complete Problems

Boolean satisfiability
is complete for NP
[Cook-Levin]

Local Hamiltonian
is complete for QMA
[Kitaev]

$$\mathrm{NP} \subseteq \mathrm{MA} \subseteq \mathrm{QMA} \subseteq \mathrm{PP} \subseteq \mathrm{PSPACE}$$

Complexity Classes and Complete Problems

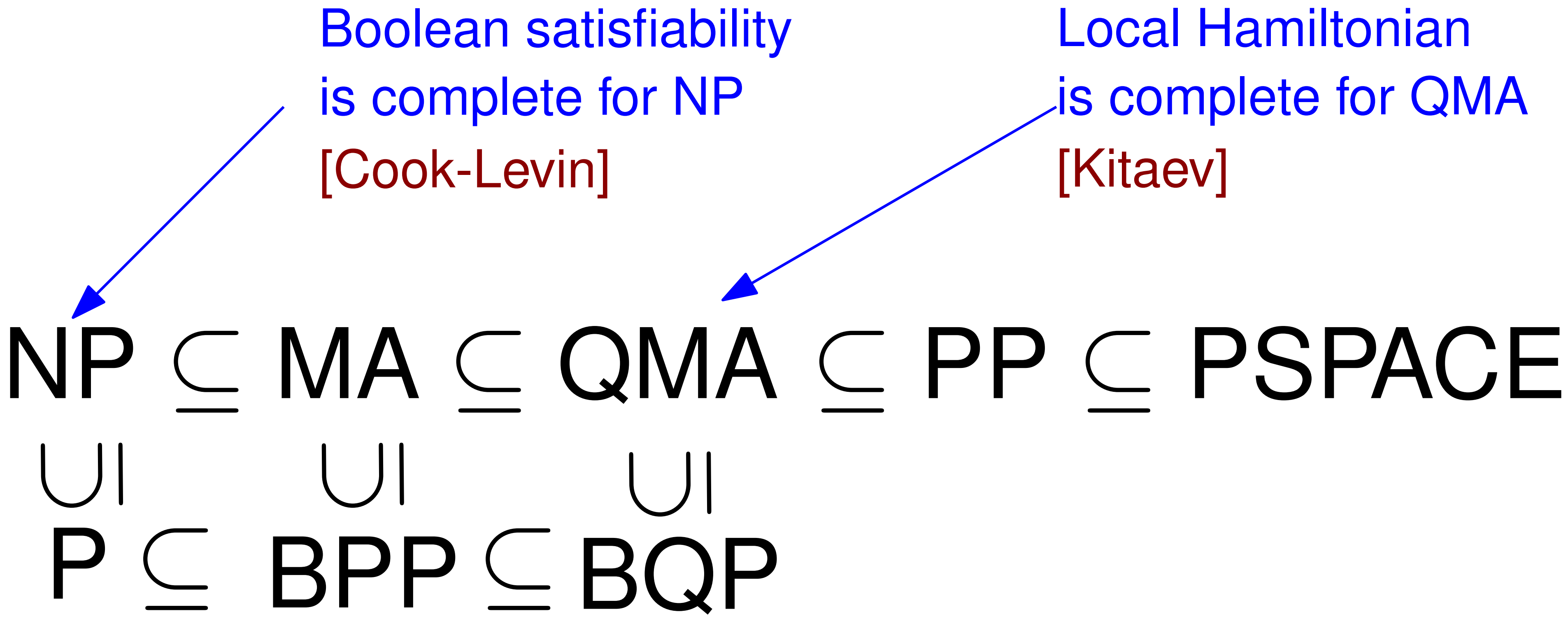
Boolean satisfiability
is complete for NP
[Cook-Levin]

Local Hamiltonian
is complete for QMA
[Kitaev]

$$\mathrm{NP} \subseteq \mathrm{MA} \subseteq \mathrm{QMA} \subseteq \mathrm{PP} \subseteq \mathrm{PSPACE}$$

$$\mathrm{P} \subseteq \mathrm{BPP} \subseteq \mathrm{BQP}$$

Complexity Classes and Complete Problems



Boolean satisfiability
is complete for NP
[Cook-Levin]

Local Hamiltonian
is complete for QMA
[Kitaev]

$$\begin{tikzcd}
\mathrm{NP} \arrow[r, phantom, "\subseteq"] & \mathrm{MA} \arrow[r, phantom, "\subseteq"] & \mathrm{QMA} \arrow[r, phantom, "\subseteq"] & \mathrm{PP} \arrow[r, phantom, "\subseteq"] & \mathrm{PSPACE} \\
\mathrm{P} \arrow[u, phantom, "\rotatebox{90}{$\subseteq$}"] \arrow[r, phantom, "\subseteq"] & \mathrm{BPP} \arrow[u, phantom, "\rotatebox{90}{$\subseteq$}"] \arrow[r, phantom, "\subseteq"] & \mathrm{BQP} \arrow[u, phantom, "\rotatebox{90}{$\subseteq$}"] & &
\end{tikzcd}$$

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 $\|H_i\| \leq 1$.

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 poly(n) bits.

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

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.

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 α

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

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 $\Delta \geq 1/\text{poly}(n)$

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 $\in$ NP

Local Hamiltonian is in QMA

Boolean Satisfiability $\in$ 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

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*?

$\langle\Phi|H|\Phi\rangle \leq E$?

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: $|\Phi\rangle$

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 $|\Phi\rangle$, $\langle \Phi | H | \Phi \rangle \geq E + \Delta$

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 $|\Phi\rangle$, $\langle \Phi | H | \Phi \rangle \geq E + \Delta$

$\Longrightarrow$

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:

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

Measure last qubit

Local Hamiltonian is in QMA

$H = H_1 + H_2 + \cdots + H_r$ 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:

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

Measure last qubit

$$|\Phi\rangle|0\rangle = \sum_j \alpha_{aj} \underbrace{|v_{aj}\rangle}_{k \text{ qubits } H_a \text{ acts on}} \underbrace{|\beta_{aj}\rangle}_{\text{the rest of the qubits}} |0\rangle$$

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:

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

Measure last qubit

$$|\Phi\rangle|0\rangle = \sum_j \alpha_{aj}|v_{aj}\rangle|\beta_{aj}\rangle|0\rangle \quad \Rightarrow$$

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

Local Hamiltonian is in QMA

$H = H_1 + H_2 + \cdots + H_r$ 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:

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

Measure last qubit

$$|\Phi\rangle|0\rangle = \sum_j \alpha_{aj}|v_{aj}\rangle|\beta_{aj}\rangle|0\rangle \quad \Rightarrow$$

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

Prob of measuring 1:

Local Hamiltonian is in QMA

$H = H_1 + H_2 + \cdots + H_r$ 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:

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

Measure last qubit

$$|\Phi\rangle|0\rangle = \sum_j \alpha_{aj}|v_{aj}\rangle|\beta_{aj}\rangle|0\rangle \quad \Rightarrow$$

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

Prob of measuring 1: $\sum_j |\alpha_{aj}|^2\lambda_{aj}$

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:

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

Measure last qubit

$$|\Phi\rangle|0\rangle = \sum_j \alpha_{aj}|v_{aj}\rangle|\beta_{aj}\rangle|0\rangle \quad \Rightarrow$$

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

Prob of measuring 1: $\sum_j |\alpha_{aj}|^2\lambda_{aj} = \langle\Phi|H_a|\Phi\rangle$

Local Hamiltonian is in QMA

$H = H_1 + H_2 + \cdots + H_r$ 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 $0 \leq \lambda_{aj} \leq 1$)

If H_a is picked, prob of measuring 1: $= \langle\Phi|H_a|\Phi\rangle$

Local Hamiltonian is in QMA

$H = H_1 + H_2 + \cdots + H_r$ 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: $= \langle \Phi | H_a | \Phi \rangle$

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

$H = H_1 + H_2 + \cdots + H_r$ 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: $= \langle \Phi | H_a | \Phi \rangle$

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 $\leq E/r$ OR $\geq (E + \Delta)/r$

Boolean Satisfiability is NP-hard [Cook-Levin]

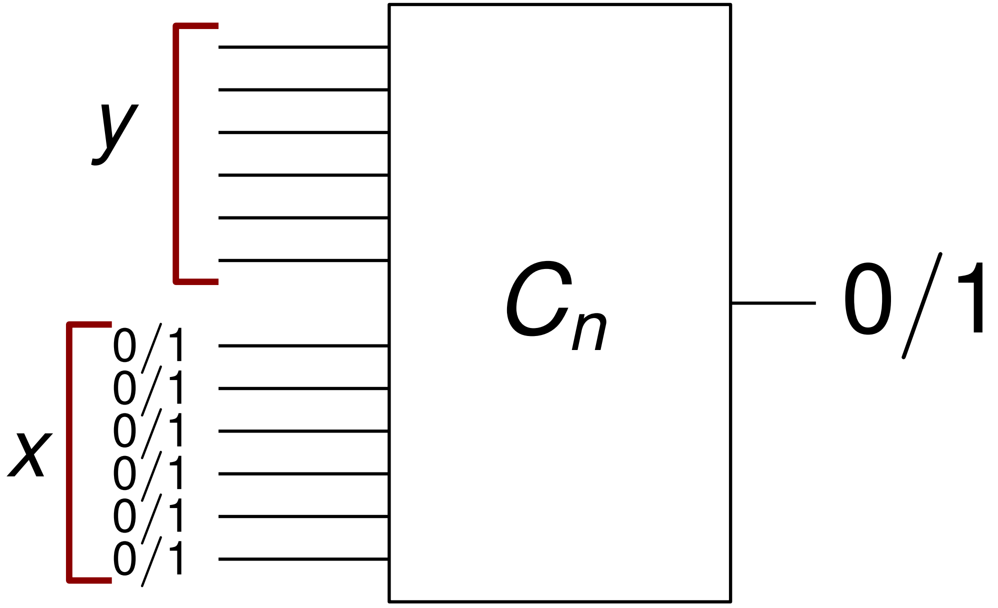
Start with a generic language L in NP

Is $x \in L$?

Boolean Satisfiability is NP-hard [Cook-Levin]

Start with a generic language L in NP

Is $x \in L$?

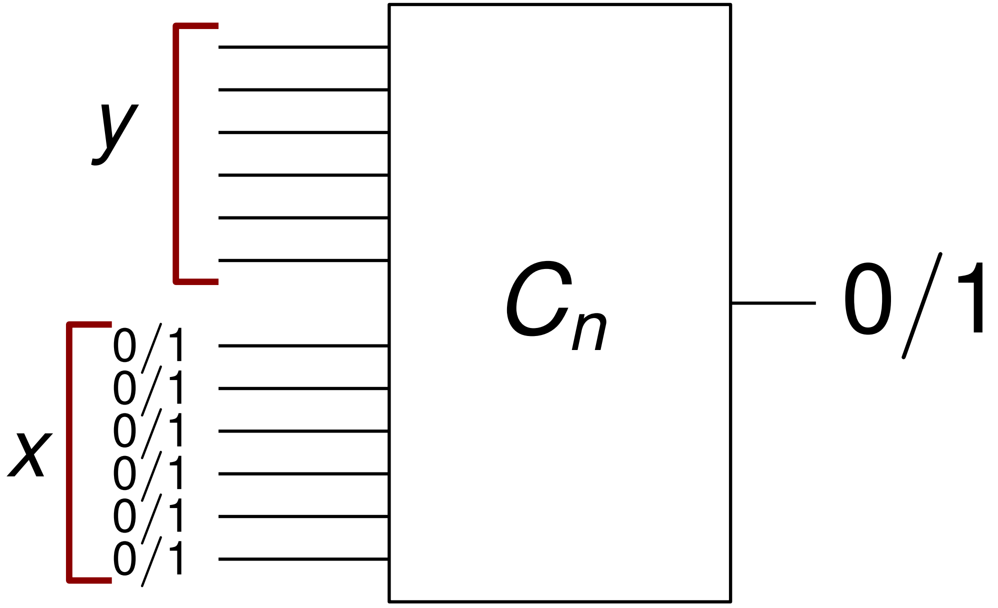


Is there a string y that causes this circuit to output 1?

Boolean Satisfiability is NP-hard [Cook-Levin]

Start with a generic language L in NP

Is $x \in L$?



$\Rightarrow$

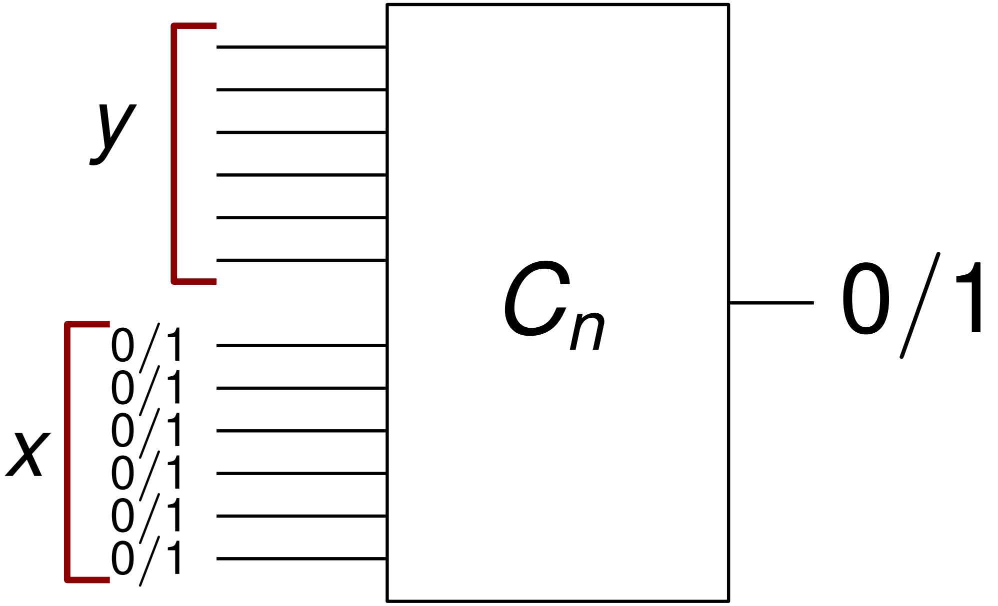
Boolean
Formula:
$\Phi_x(y)$

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 $x \in L$?



$\Rightarrow$

Boolean Formula: $\Phi_x(y)$

Is there a string y that causes this circuit to output 1? $\Leftrightarrow$ Is $\Phi_x(y)$ satisfiable?

Local Hamiltonian is QMA-hard

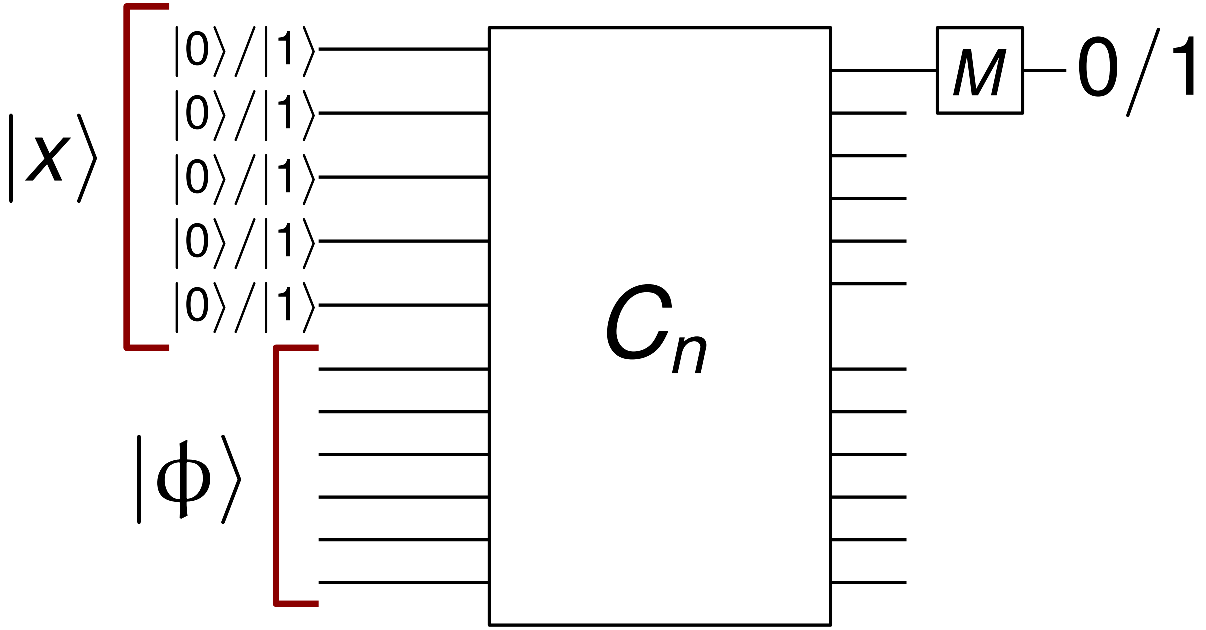
Start with a generic promise problem in QMA

Is $x \in$ Yes? or is $x \in$ NO?

Local Hamiltonian is QMA-hard

Start with a generic promise problem in QMA

Is $x \in$ Yes? or is $x \in$ NO?

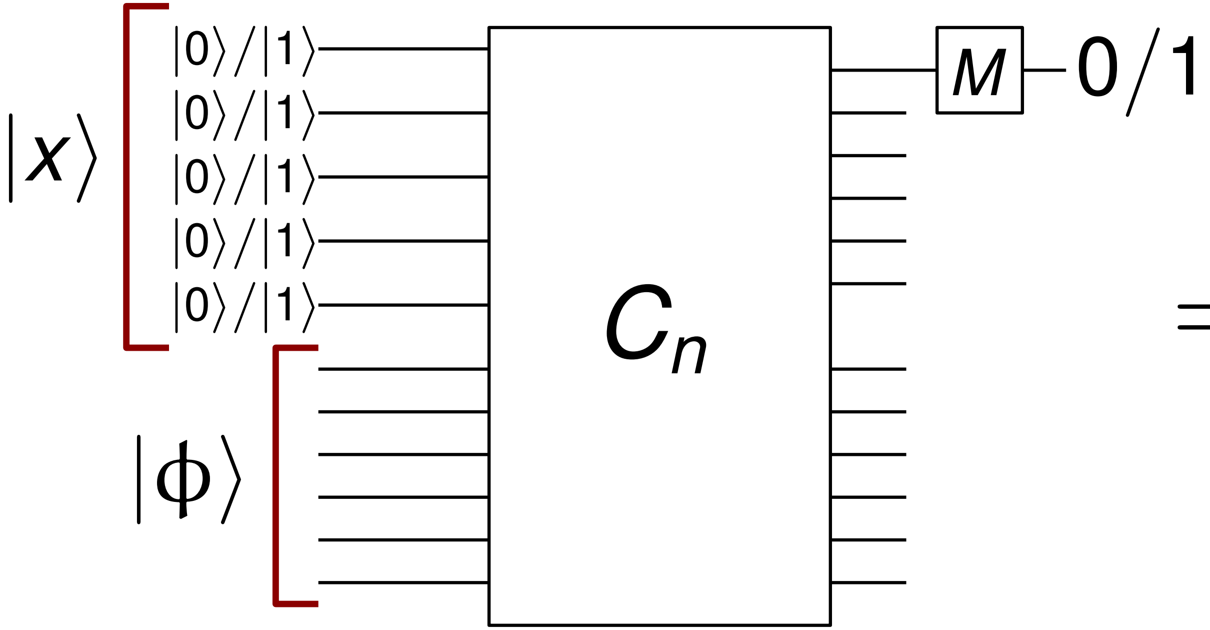


Is there a quantum state $\phi\rangle$
that causes this quantum circuit
to output 1 with high probability?

Local Hamiltonian is QMA-hard

Start with a generic promise problem in QMA

Is $x \in$ Yes? or is $x \in$ NO?



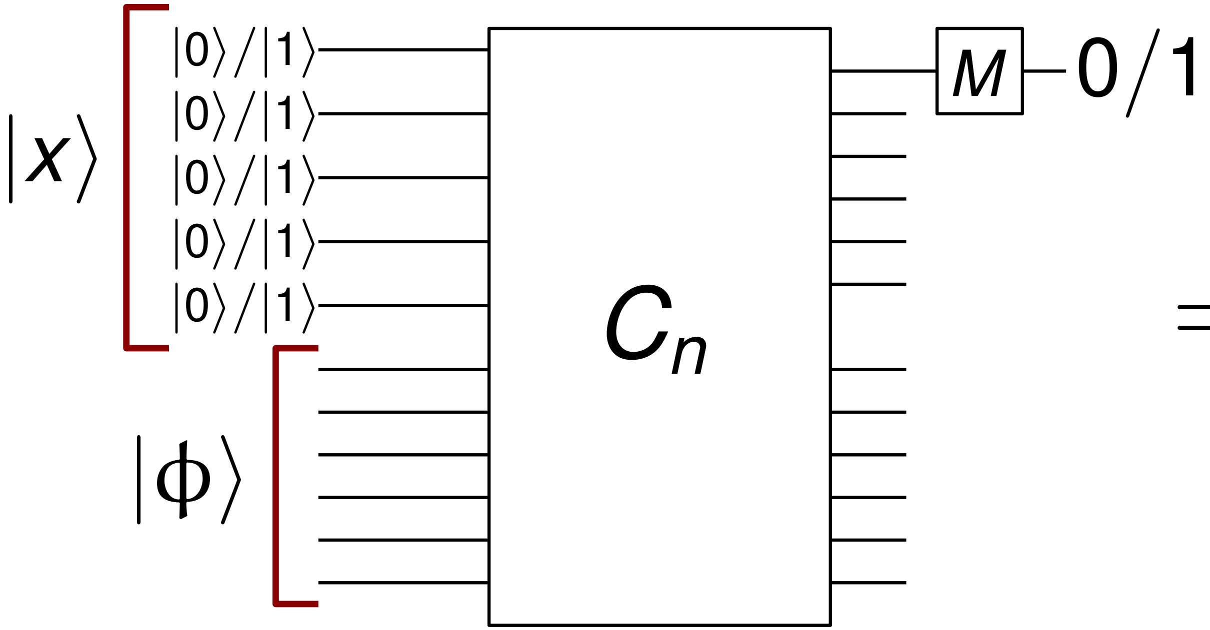
$\Rightarrow$ 5-Local Hamiltonian: (H_x, E, Δ)

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 $x \in$ Yes?   or is $x \in$ NO?



$\Rightarrow$ 5-Local Hamiltonian: (H_x, E, Δ)

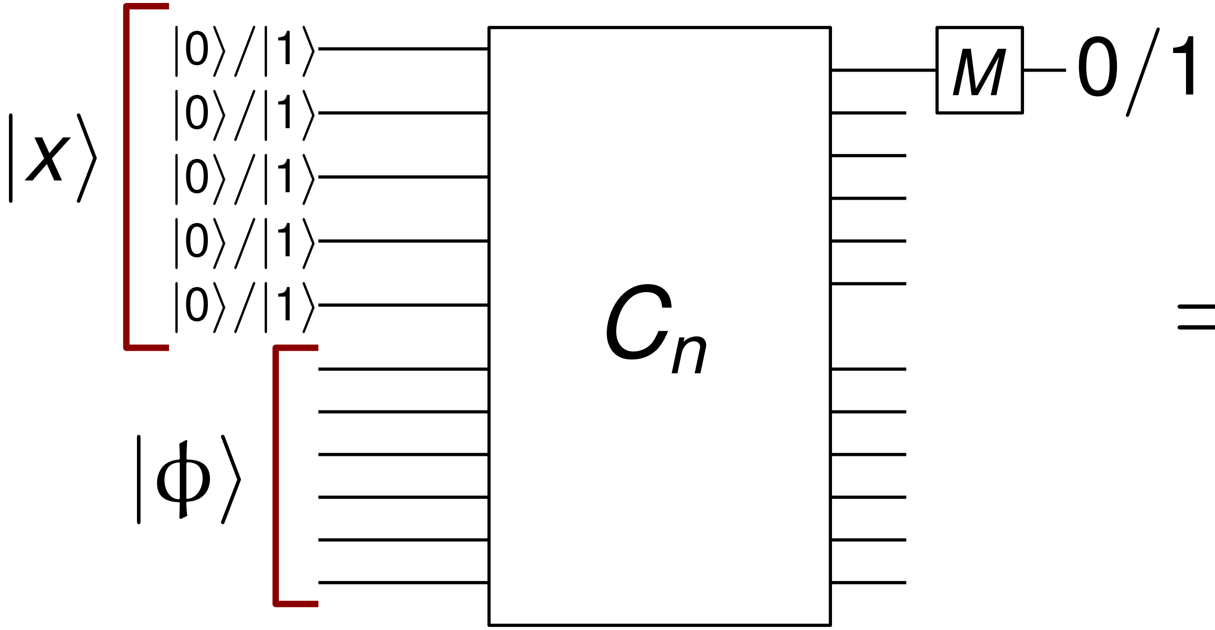
Is there a quantum state $\phi\rangle$
that causes this quantum circuit
to output 1 with high probability?

$\Rightarrow$ Ground energy of $H_x \leq E$

Local Hamiltonian is QMA-hard [Kitaev 1995]

Start with a generic promise problem in QMA

Is $x \in$ Yes? or is $x \in$ NO?



$\Rightarrow$ 5-Local Hamiltonian: (H_x, E, Δ)

Is there a quantum state $\phi\rangle$ that causes this quantum circuit to output 1 with high probability? $\Rightarrow$ Ground energy of $H_x \leq E$

For every $|\phi\rangle$, circuit outputs 0 w.h.p. $\Rightarrow$ Ground energy of $H_x \geq E + \Delta$

Boolean Satisfiability is NP-hard

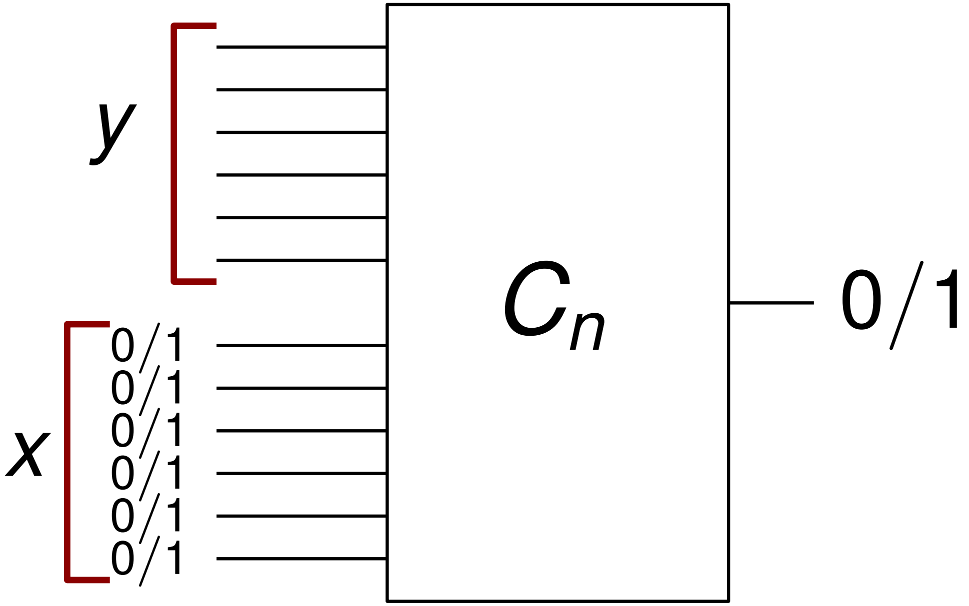
Start with a generic language L in NP

Is $x \in L$?

Boolean Satisfiability is NP-hard

Start with a generic language L in NP

Is $x \in L$?

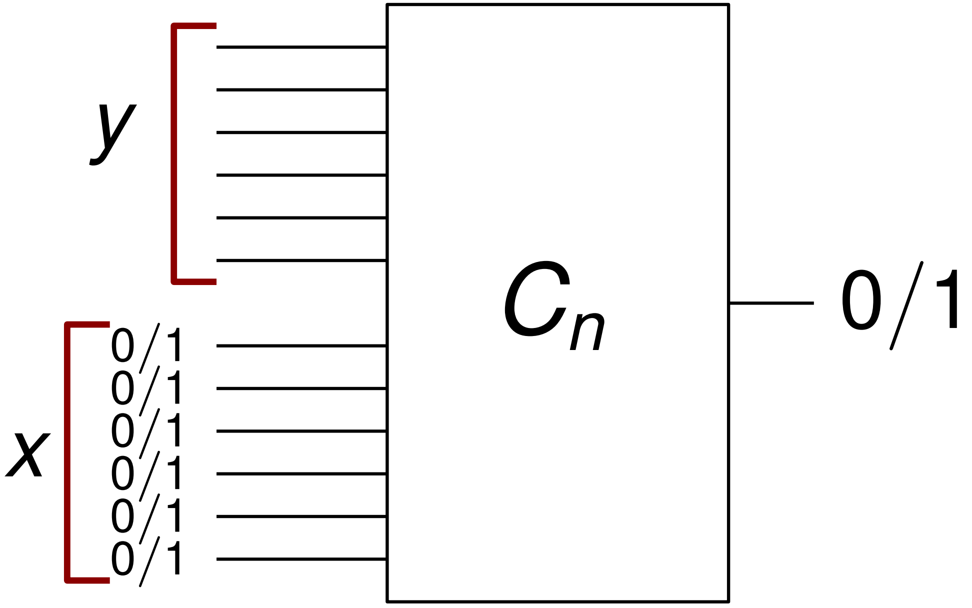


Is there a string y that causes this circuit to output 1?

Boolean Satisfiability is NP-hard

Start with a generic language L in NP

Is $x \in L$?



Is there a string y that causes this circuit to output 1?

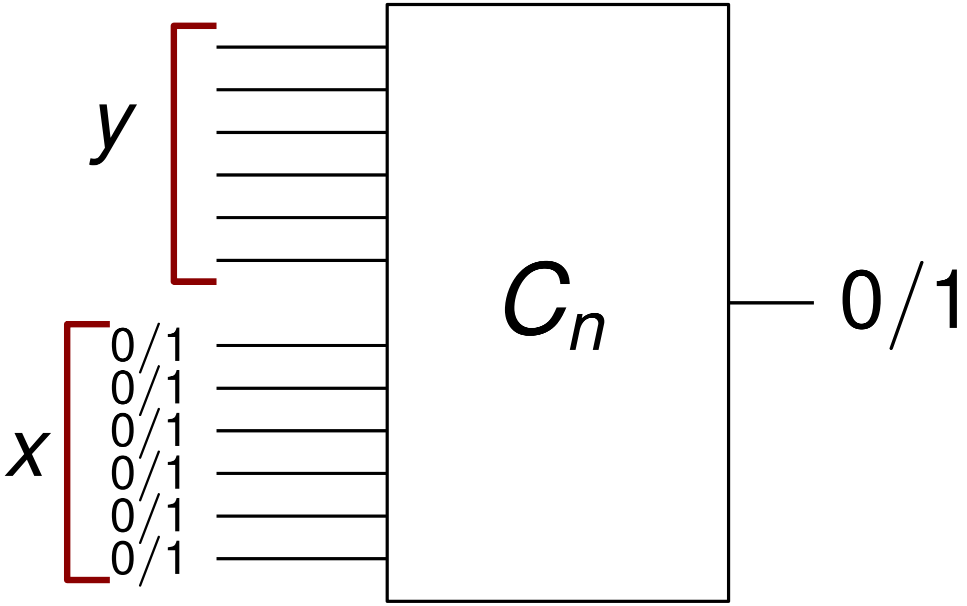
Reduction: input x

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

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?

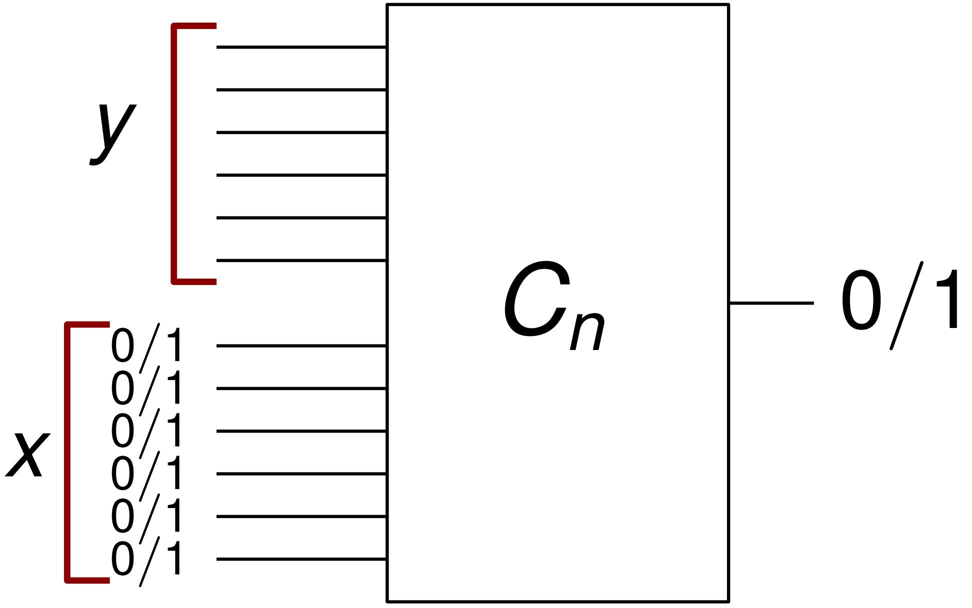
Reduction: input x

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

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?

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.

Circuit to Boolean formula

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

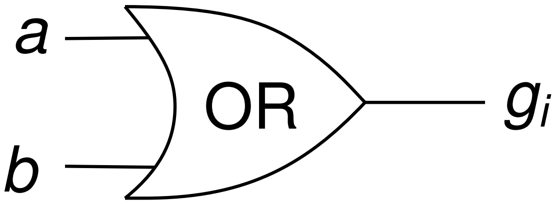
Add variables $g_1, \ldots, g_m$, one for each gate.

Circuit to Boolean formula

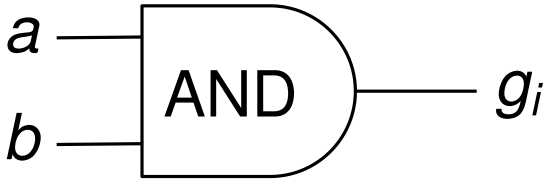
Circuit C_n has gates $G_1, \ldots, G_m$, where $m = poly(n)$.

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

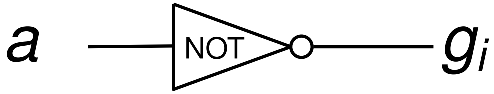
Add a clause for each gate:



$\Rightarrow \quad ((a \vee b) \leftrightarrow g_i)$



$\Rightarrow \quad ((a \wedge b) \leftrightarrow g_i)$



$\Rightarrow \quad (\neg a \leftrightarrow g_i)$

Circuit to Boolean formula

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

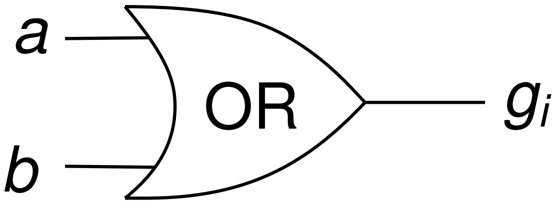
Add variables $g_1, \dots, g_m$, one for each gate.

Add a clause for each gate:

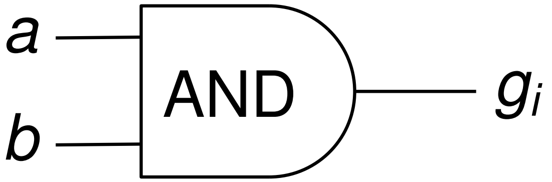
Hard-code x:

$x_i = 0 \quad \rightarrow$ add clause $(\neg x_i)$

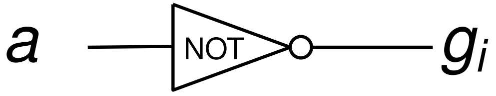
$x_i = 1 \quad \rightarrow$ add clause (x_i)



$\Rightarrow \quad ((a \vee b) \leftrightarrow g_i)$



$\Rightarrow \quad ((a \wedge b) \leftrightarrow g_i)$



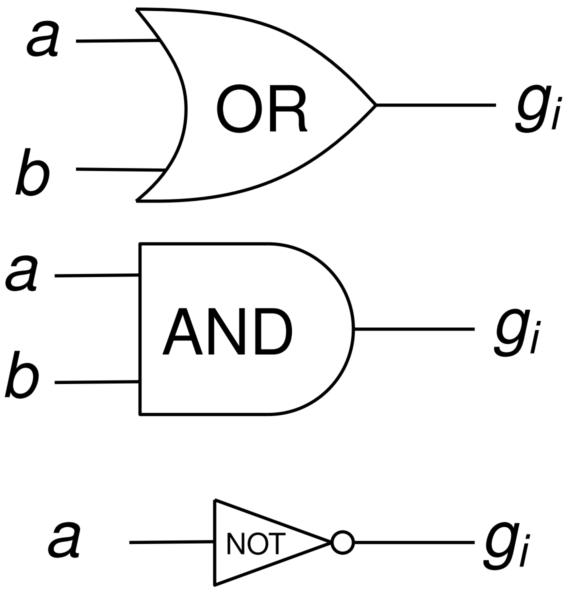
$\Rightarrow \quad (\neg a \leftrightarrow g_i)$

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:



$\Rightarrow \quad ((a \vee b) \leftrightarrow g_i)$

$\Rightarrow \quad ((a \wedge b) \leftrightarrow g_i)$

$\Rightarrow \quad (\neg a \leftrightarrow g_i)$

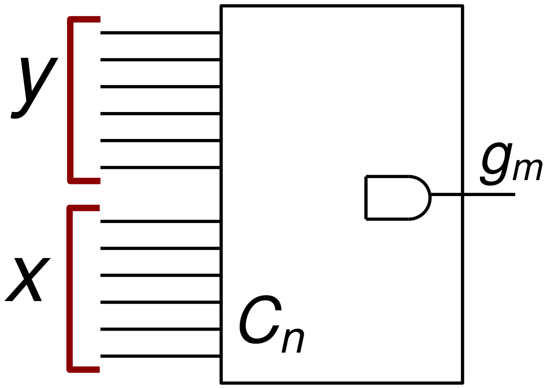
Hard-code x:

$x_i = 0 \quad \rightarrow$ add clause $(\neg x_i)$

$x_i = 1 \quad \rightarrow$ add clause (x_i)

Output of G_m = output of circuit:

Add clause (g_m)

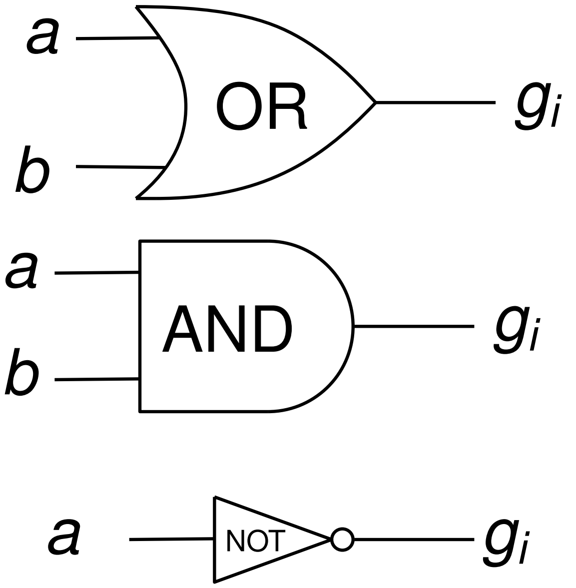


Circuit to Boolean formula

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:



$\Rightarrow \quad ((a \vee b) \leftrightarrow g_i)$

$\Rightarrow \quad ((a \wedge b) \leftrightarrow g_i)$

$\Rightarrow \quad (\neg a \leftrightarrow g_i)$

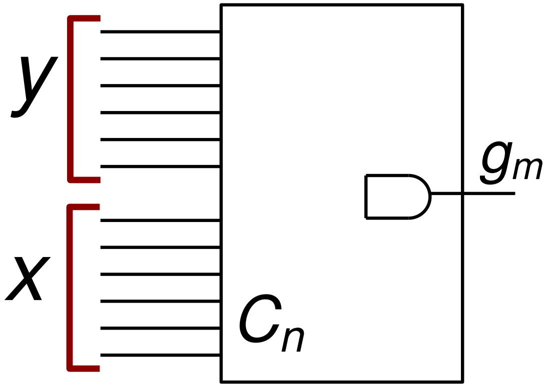
Hard-code x:

$x_i = 0 \quad \rightarrow$ add clause $(\neg x_i)$

$x_i = 1 \quad \rightarrow$ add clause (x_i)

Output of G_m = output of circuit:

Add clause (g_m)



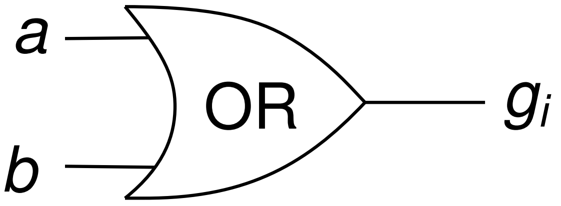
Boolean formula is the conjunction of all the clauses.

Circuit to Boolean formula

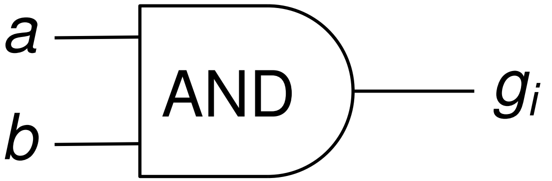
Circuit C_n has gates $G_1, \ldots, G_m$, where $m = poly(n)$.

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

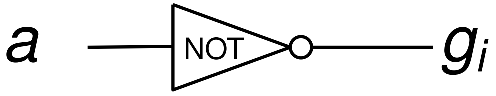
Add a clause for each gate:



$\Rightarrow$ $((a \vee b) \leftrightarrow g_i)$



$\Rightarrow$ $((a \wedge b) \leftrightarrow g_i)$



$\Rightarrow$ $(\neg a \leftrightarrow g_i)$

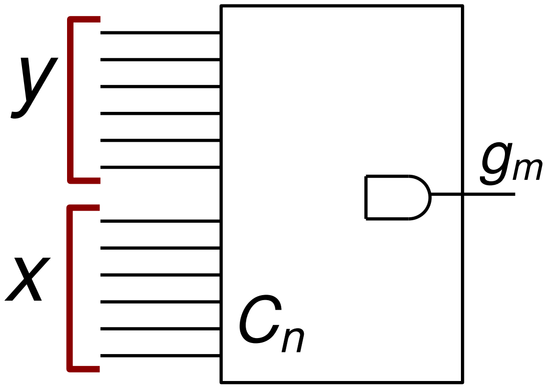
Hard-code x:

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

$x_i = 1 \rightarrow$ add clause (x_i)

Output of G_m = output of circuit:

Add clause (g_m)



Boolean formula is the conjunction of all the clauses.

Can reduce to CNF or 3SAT form.

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.

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

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:

Turing Machine that runs in time poly(n), where $|x| = n$

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

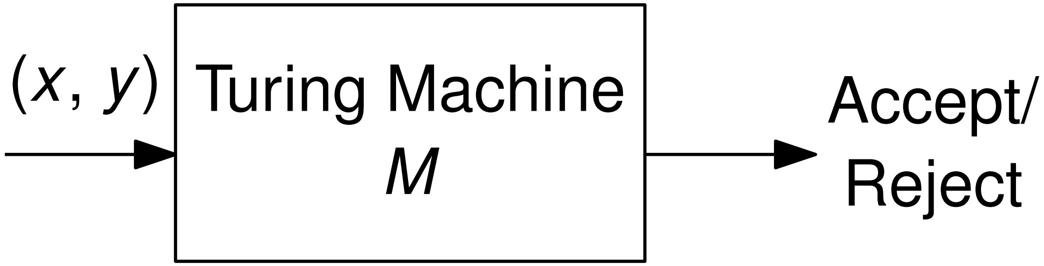
Start with a generic language L in NP

Is $x \in L$?

Boolean Satisfiability is NP-hard

Start with a generic language L in NP

Is $x \in L$?

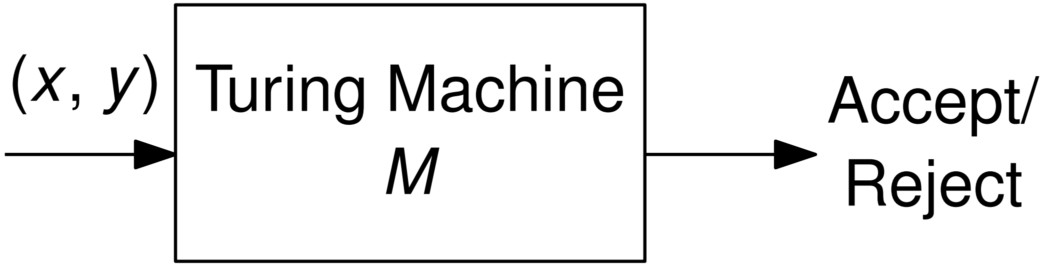


Is there a string y that causes this Turing Machine to accept?

Boolean Satisfiability is NP-hard

Start with a generic language L in NP

Is $x \in L$?



$\Rightarrow$

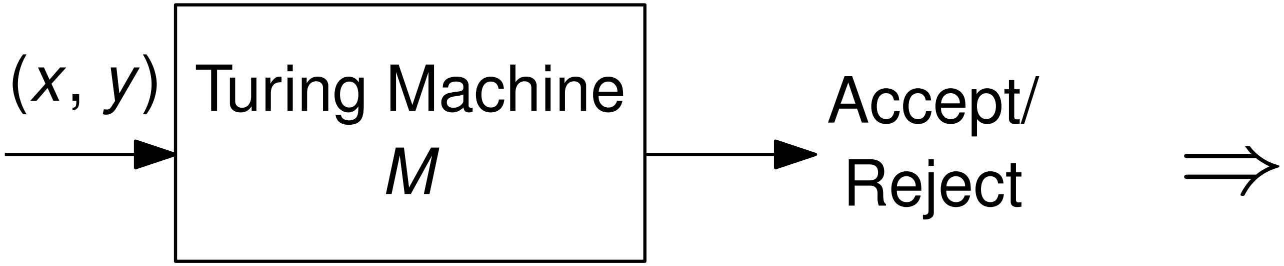
Boolean
Forumla:
$\Phi_x(y)$

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)

Turing Machine

M

Accept/

Reject

$\Rightarrow$

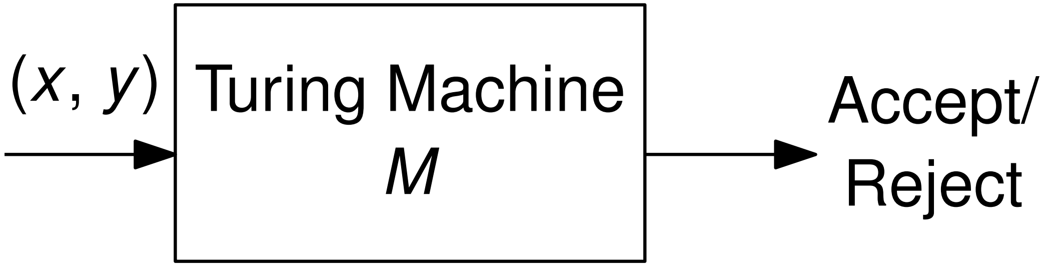
Boolean

Forumla:

$\Phi_x(y)$

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

Turing Machine Tableau

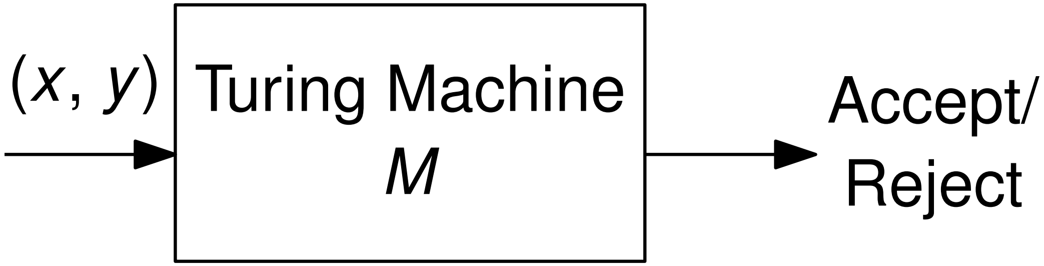


(x, y)

Turing Machine
M

Accept/
Reject

Turing Machine Tableau

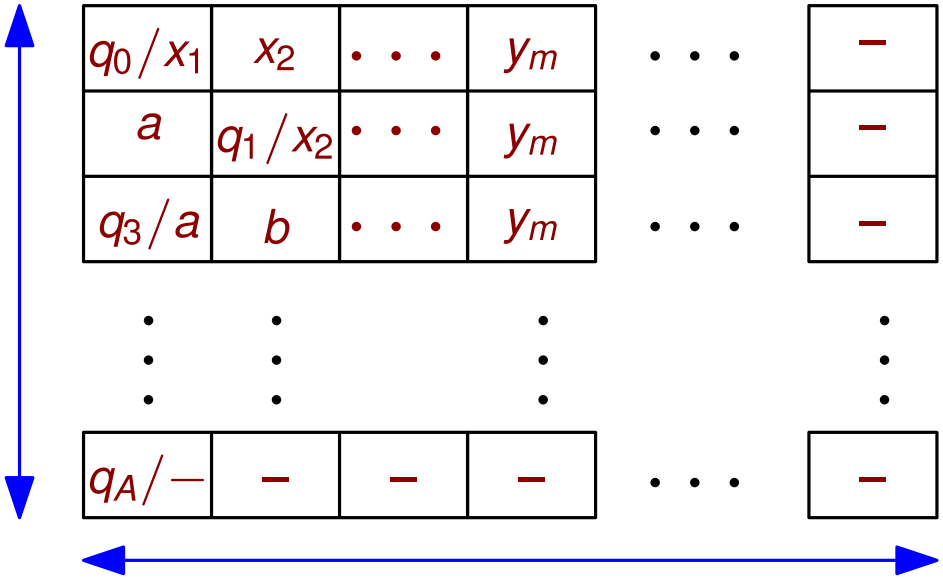


(x, y)

Turing Machine M

Accept/ Reject

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



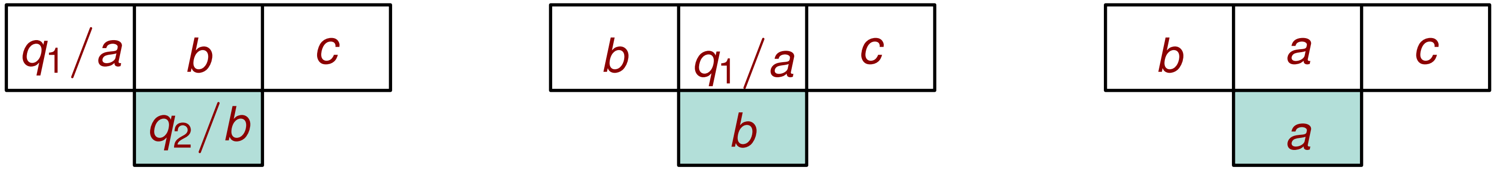
Height: running time $= n^c$.

q_0/x_1	x_2	$\cdots$	y_m	$\cdots$	$-$
a	q_1/x_2	$\cdots$	y_m	$\cdots$	$-$
q_3/a	b	$\cdots$	y_m	$\cdots$	$-$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$q_A/-$	$-$	$-$	$-$	$\cdots$	$-$

Width: space used $\leq n^c$

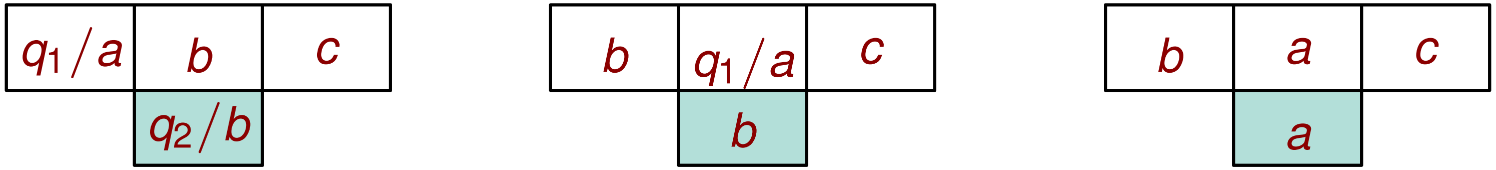
Turing Machine to Boolean formula

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



Turing Machine to Boolean formula

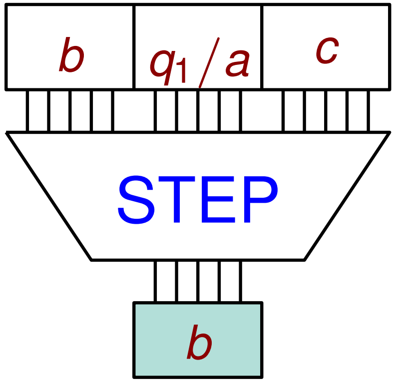
Contents of a cell determined by the three cells above it.



Can build Boolean circuit STEP

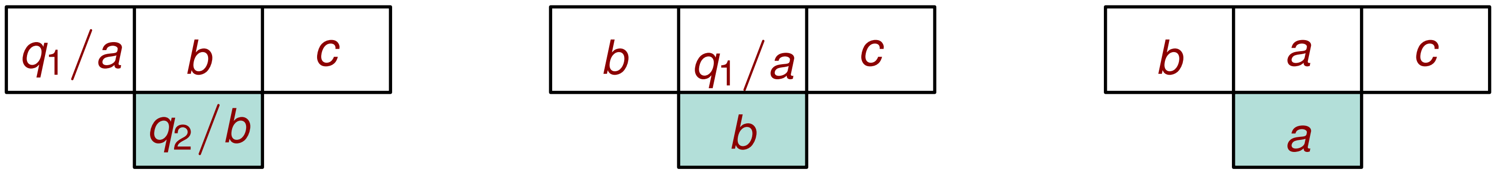
- input (binary encording of) 3 cells
- output (binary encording of) 1 cell

Circuit is constant size



Turing Machine to Boolean formula

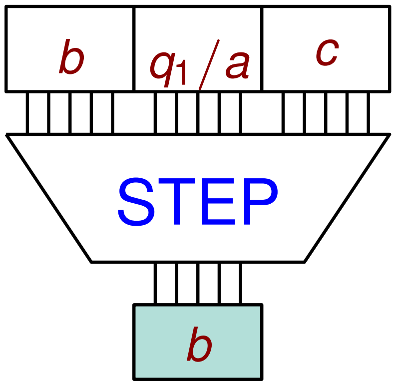
Contents of a cell determined by the three cells above it.



Can build Boolean circuit STEP

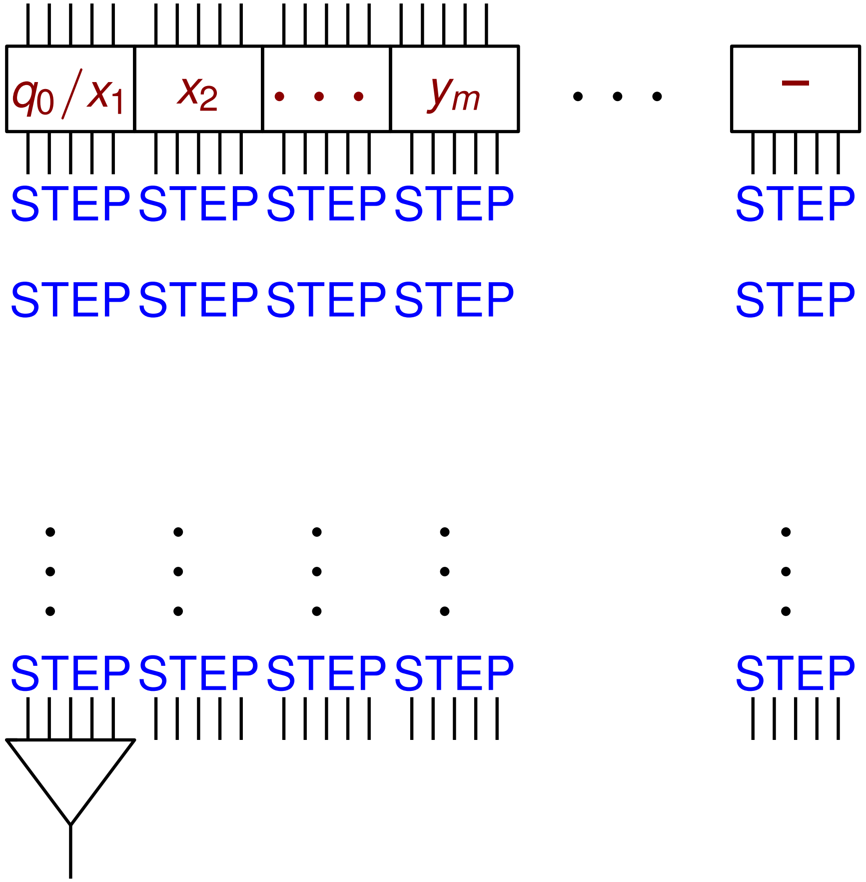
- input (binary encording of) 3 cells
- output (binary encording of) 1 cell

Circuit is constant size

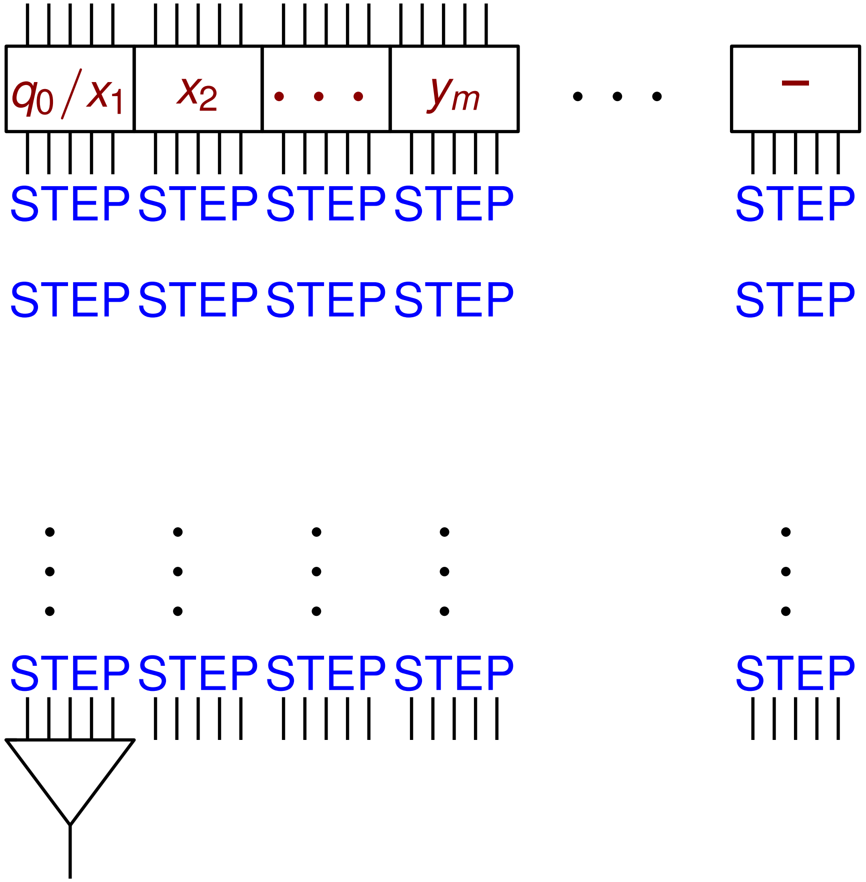


Each circuit can be converted to a Boolean formula
(set of Boolean constraints)

Turing Machine to Boolean formula



Turing Machine to Boolean formula



q_0 / x_1 x_2 . . . y_m . . . $-$

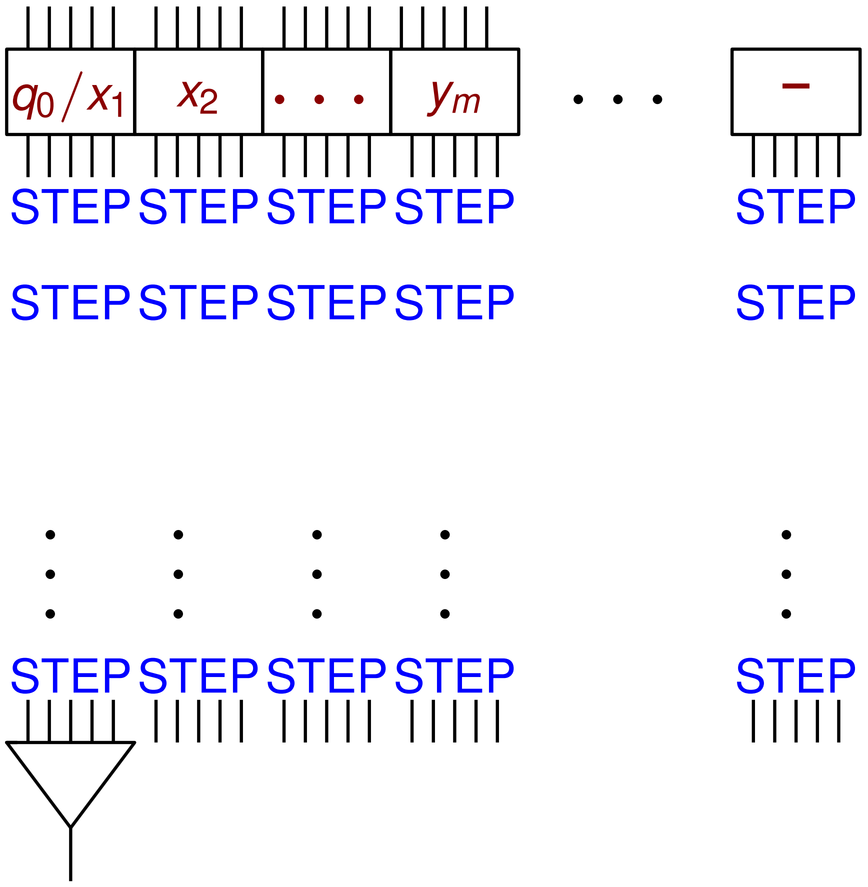
STEP STEP STEP STEP STEP

STEP STEP STEP STEP STEP

STEP STEP STEP STEP STEP

Output 1 iff cell contains q_{ACC}

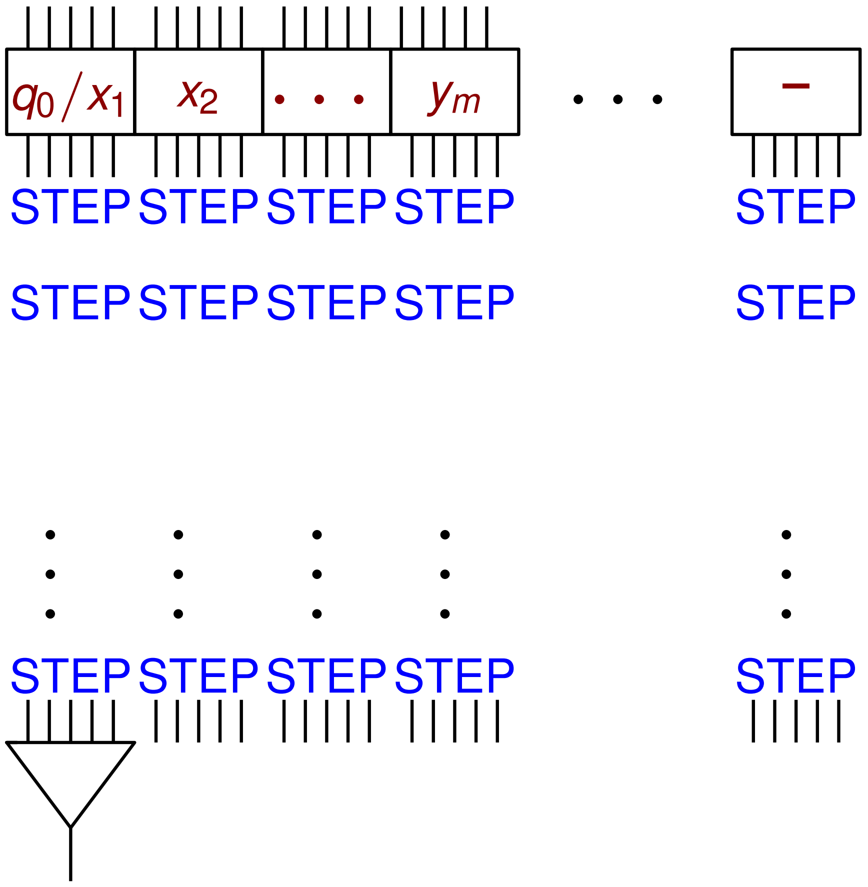
Turing Machine to Boolean formula



Output 1 iff cell contains q_{ACC}

Features we will keep for QMA

Turing Machine to Boolean formula

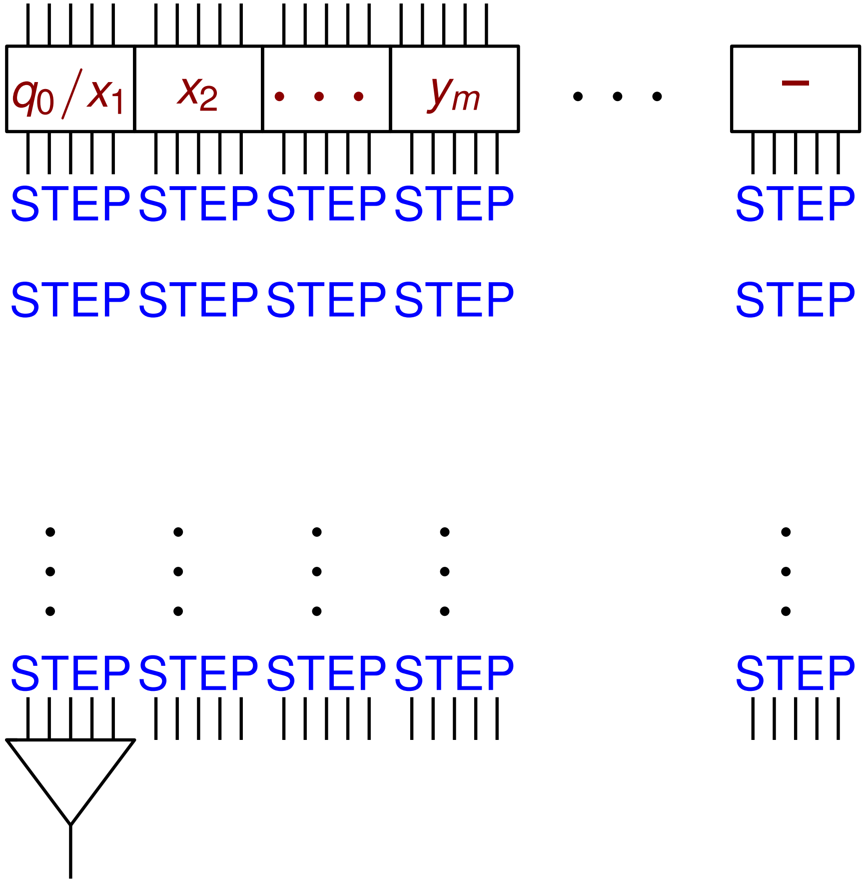


Output 1 iff cell contains q_{ACC}

Features we will keep for QMA

Hard code input *x* into circuit/constraints

Turing Machine to Boolean formula



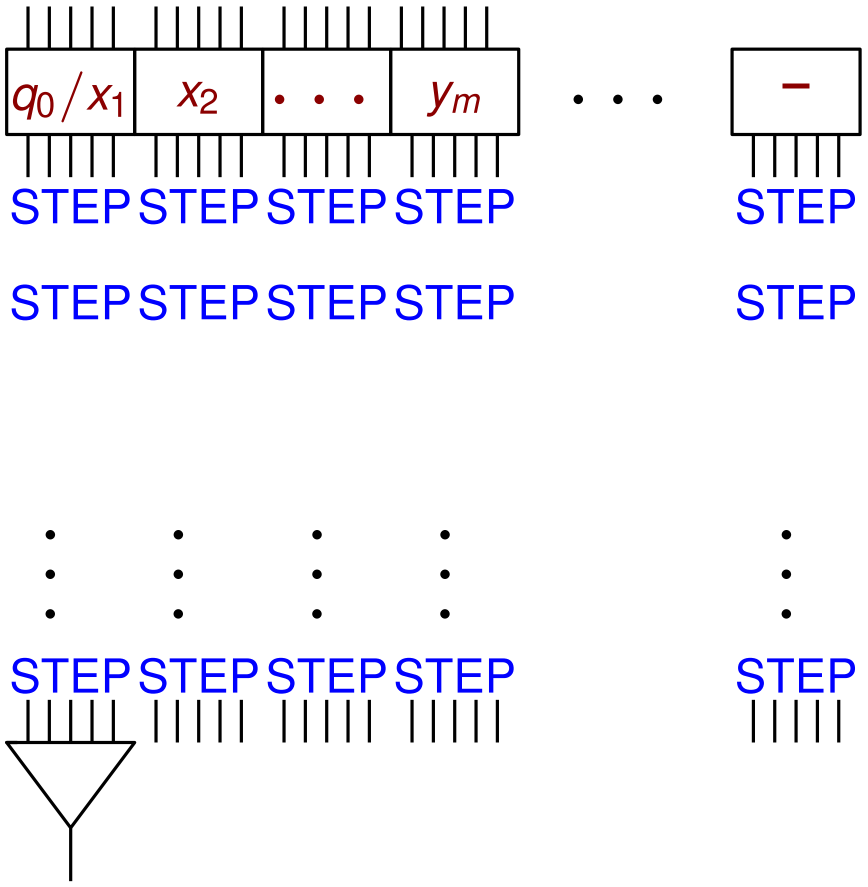
Output 1 iff cell contains q_{ACC}

Features we will keep for QMA

Hard code input x into circuit/constraints

Input y (witness) is variable

Turing Machine to Boolean formula



Output 1 iff cell contains q_{ACC}

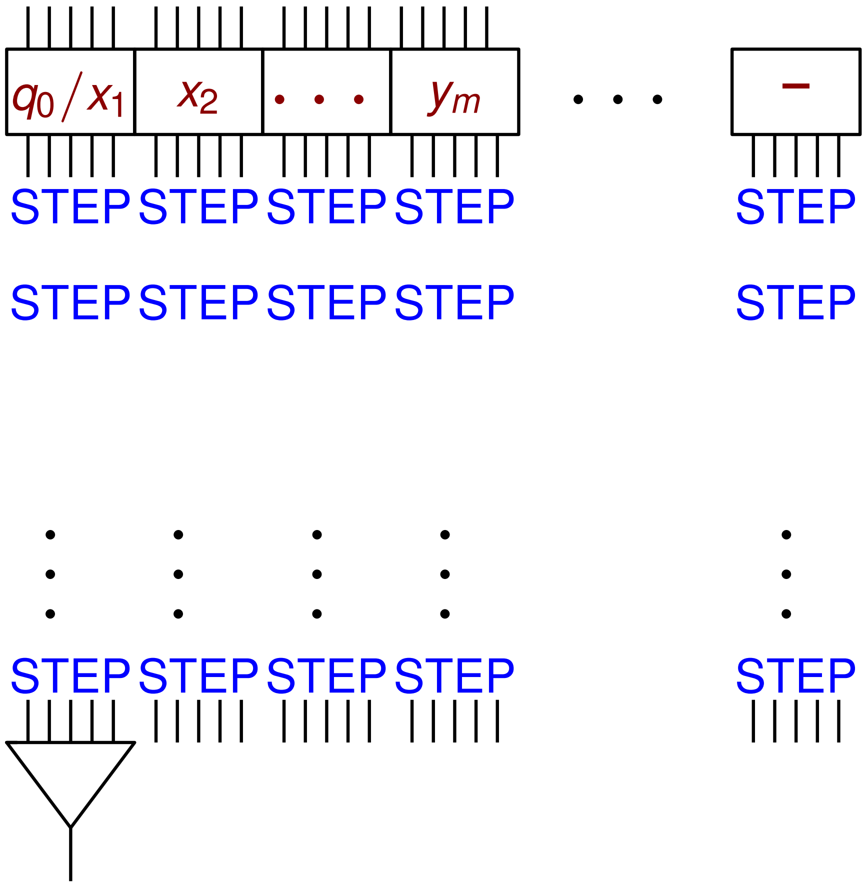
Features we will keep for QMA

Hard code input x into circuit/constraints

Input y (witness) is variable

Satisfied constraints $\Rightarrow$ State represents entire history of the computation. (Configuration after each step)

Turing Machine to Boolean formula



Output 1 iff cell contains q_{ACC}

Features we will keep for QMA

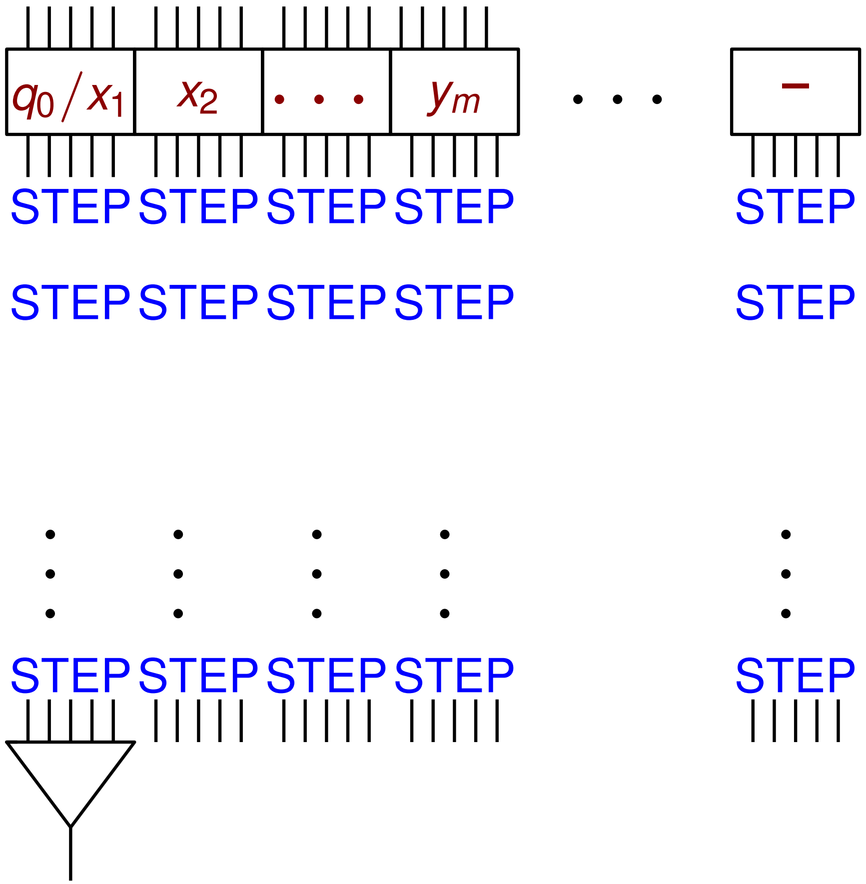
Hard code input *x* into circuit/constraints

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

Turing Machine to Boolean formula



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 $\Rightarrow$ State represents entire history of the computation. (Configuration after each step)

Additional term to test if computation accepts

On to Part II...