Adaptive Reprogramming in the QROM

QIP 2012
Virtual

Alex Grilo, Kathrin Hövelmanns, Andreas Hülsing and
Christian Majenz

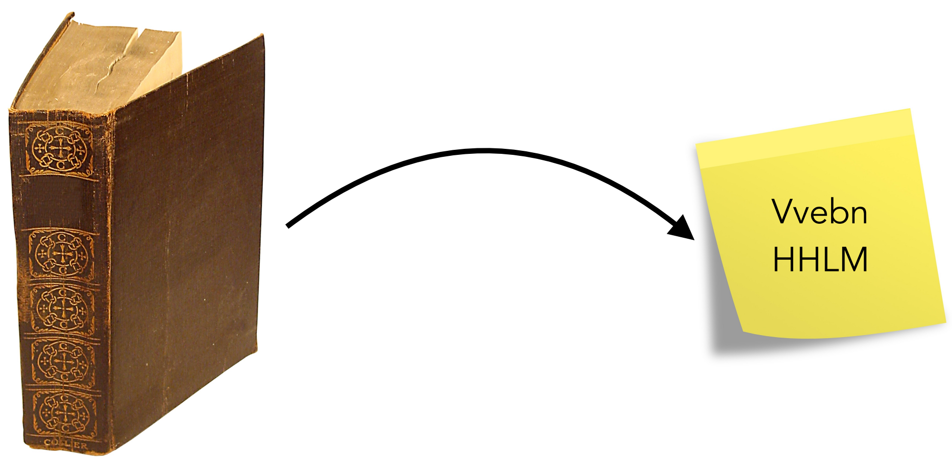
Outline

- Motivation — the quantum random oracle model
- The adaptive reprogramming game
- Results
- Reprogramming superposition oracles
- A matching algorithm

Motivation — The Quantum Random Oracle Model (QROM)

Hash functions

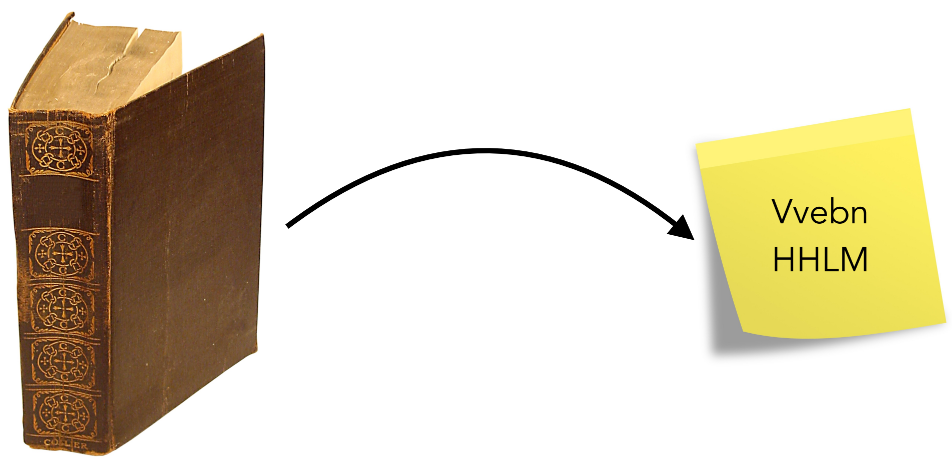
Hash functions are everywhere in crypto



Hash functions

Hash functions are everywhere in crypto

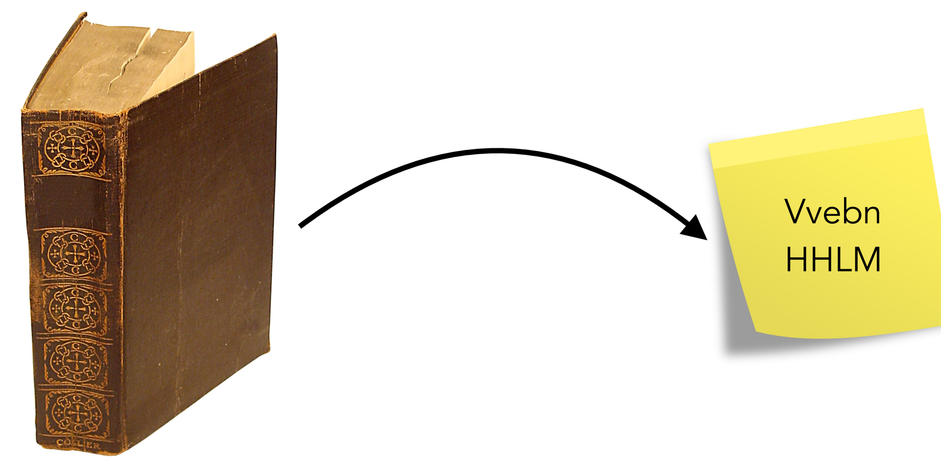
- Digital signatures
- Message authentication
- Chosen-ciphertext security
- Commitments
- …



Hash functions

Hash functions are everywhere in crypto

- Digital signatures
- Message authentication
- Chosen-ciphertext security
- Commitments
- …

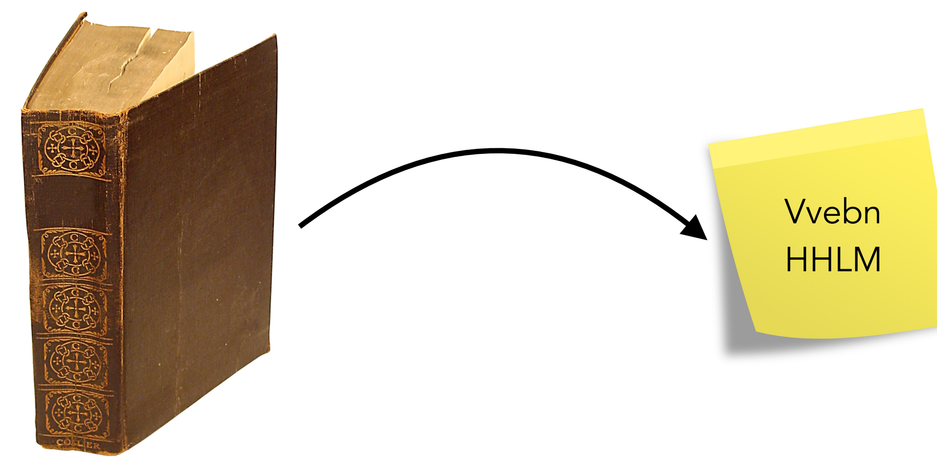


Concept: simple

Hash functions

Hash functions are everywhere in crypto

- Digital signatures
- Message authentication
- Chosen-ciphertext security
- Commitments
- …



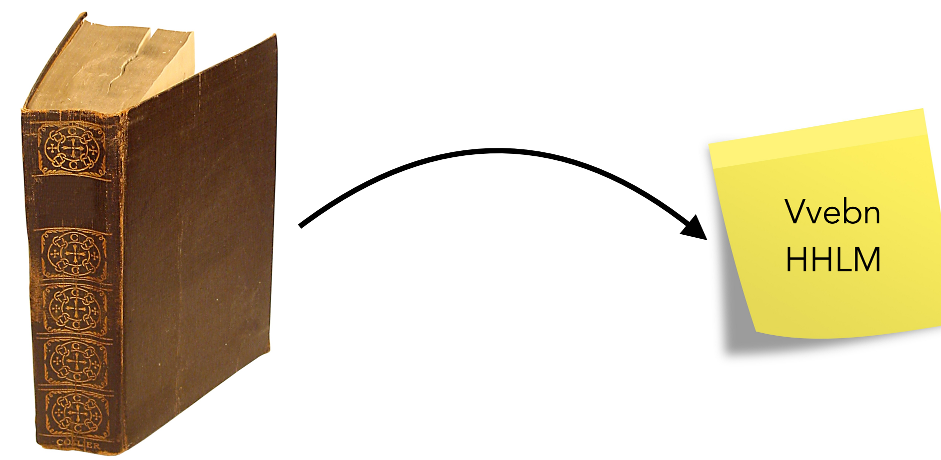
Concept: simple

Proving security: Hard

Hash functions

Hash functions are everywhere in crypto

- Digital signatures
- Message authentication
- Chosen-ciphertext security
- Commitments
- …



Concept: simple

Proving security:
Hard

Solution:
(Quantum) Random Oracle Model

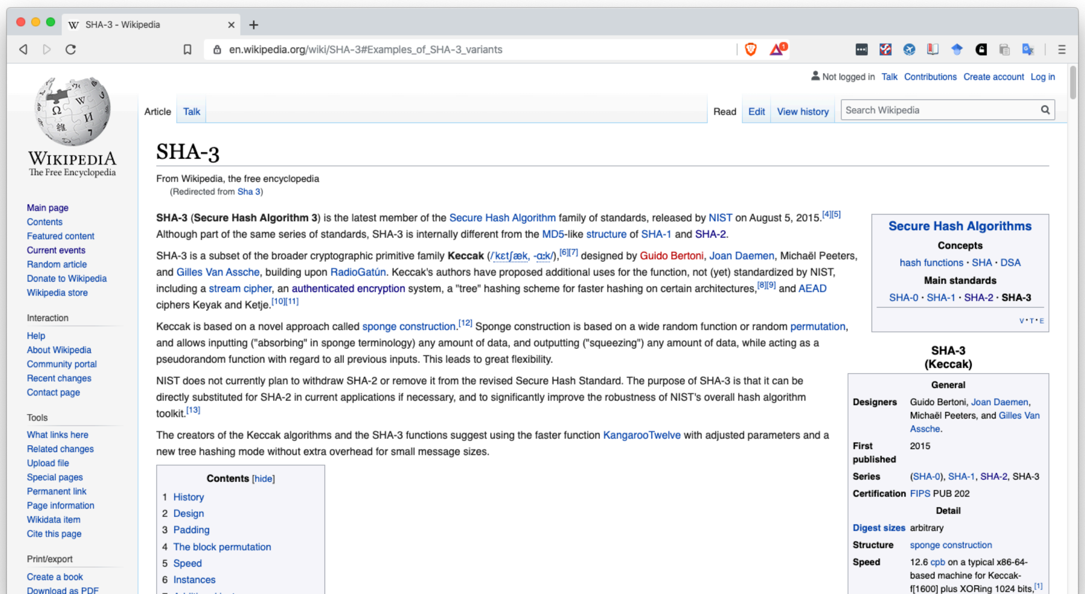
Random Oracle Model

Idealized model of cryptographic hash functions

Random Oracle Model

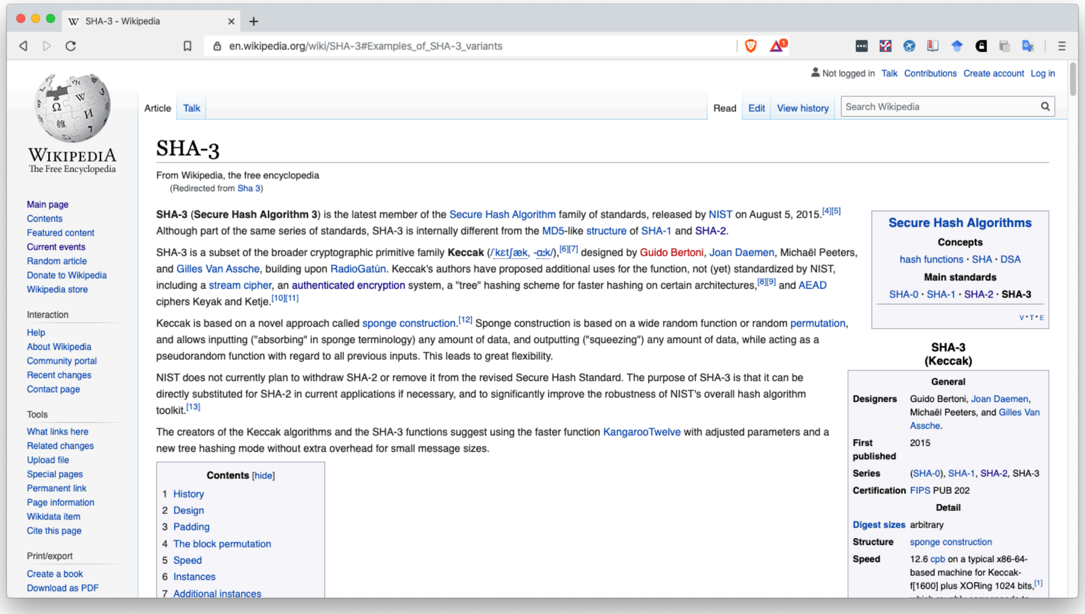
Idealized model of cryptographic hash functions

Reality



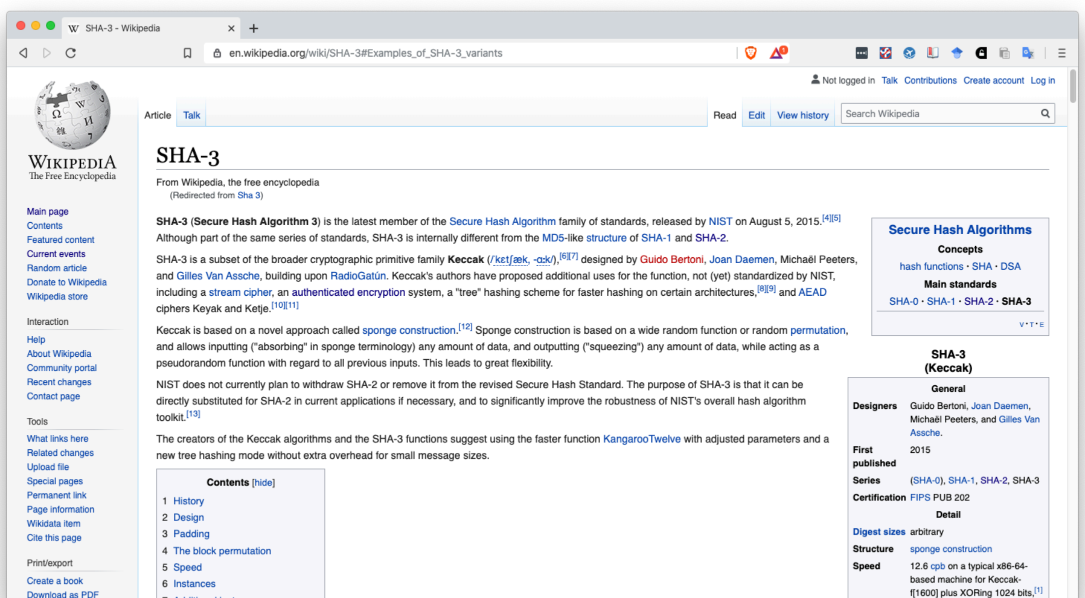
Random Oracle Model

Idealized model of cryptographic hash functions

Reality	Model
	$H : \{0,1\}^* \rightarrow \{0,1\}^n$ Uniformly random All agents have black-box access to H

Random Oracle Model

Idealized model of cryptographic hash functions

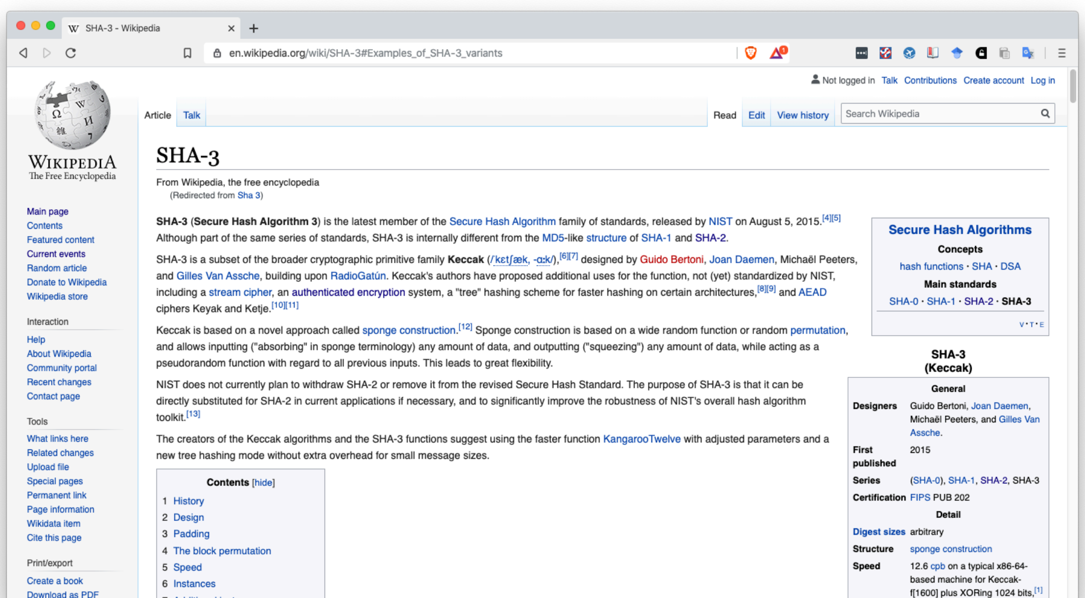
Reality	Model
	$H : \{0,1\}^* \to \{0,1\}^n$ Uniformly random All agents have black-box access to H

+ Simpler proofs

Random Oracle Model

Idealized model of cryptographic hash functions

Reality



Model

$H : \{0,1\}^* \rightarrow \{0,1\}^n$

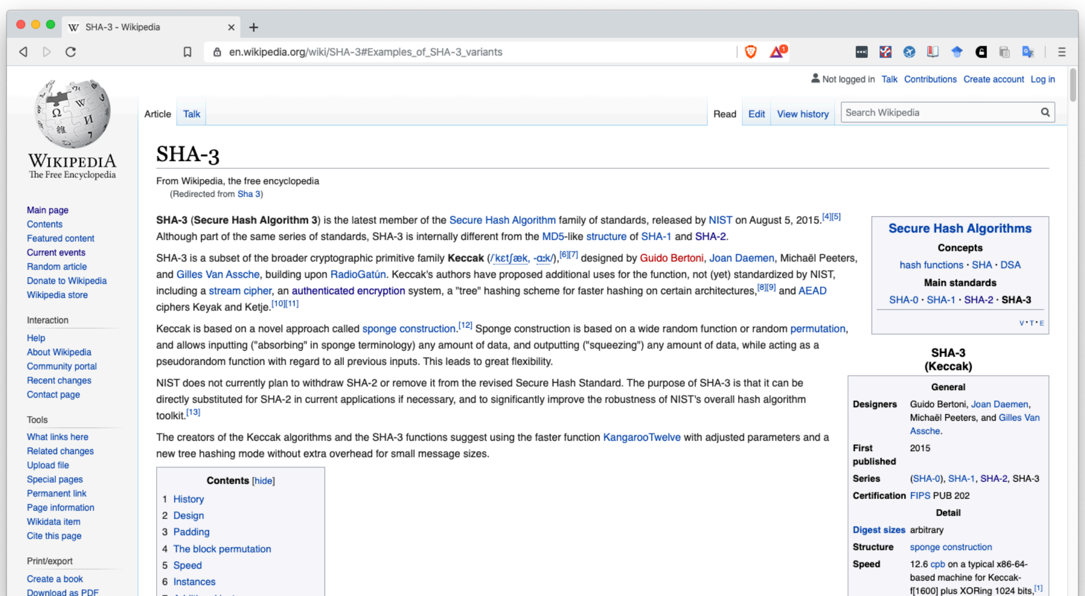
Uniformly random

All agents have black-box access to H

+ Simpler proofs
+ More efficient constructions with provable security

Quantum Random Oracle Model

Attackers with quantum computer can evaluate hash function on it!

Reality	Model
	$H : \{0,1\}^* \to \{0,1\}^n$ Uniformly random All agents have quantum black-box access to H

Quantum Random Oracle Model (Boneh et al. '10)

- Security reductions are quantum algorithms
- Quantum query complexity

The adaptive reprogramming game

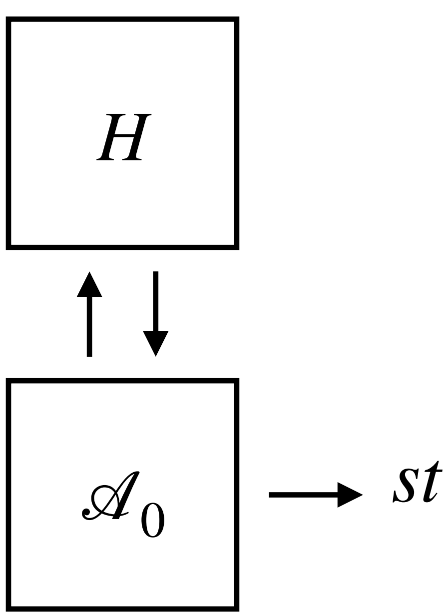
The game (simplest version)

Uniformly random function $H : \{0,1\}^n \to \{0,1\}^n$

two-stage oracle algorithm $\mathscr{A} = (\mathscr{A}_0, \mathscr{A}_1)$

The game (simplest version)

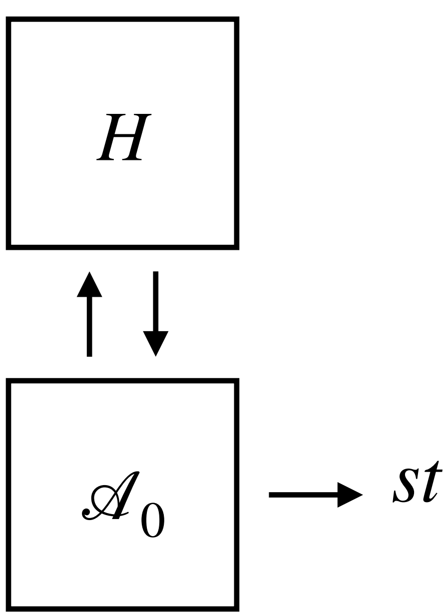
Uniformly random function $H : \{0,1\}^n \rightarrow \{0,1\}^n$
two-stage oracle algorithm $\mathscr{A} = (\mathscr{A}_0, \mathscr{A}_1)$



The game (simplest version)

Uniformly random function $H : \{0,1\}^n \to \{0,1\}^n$

two-stage oracle algorithm $\mathscr{A} = (\mathscr{A}_0, \mathscr{A}_1)$



$$b \xleftarrow{\$} \{0,1\}$$

$$x^* \xleftarrow{\$} \{0,1\}^n$$

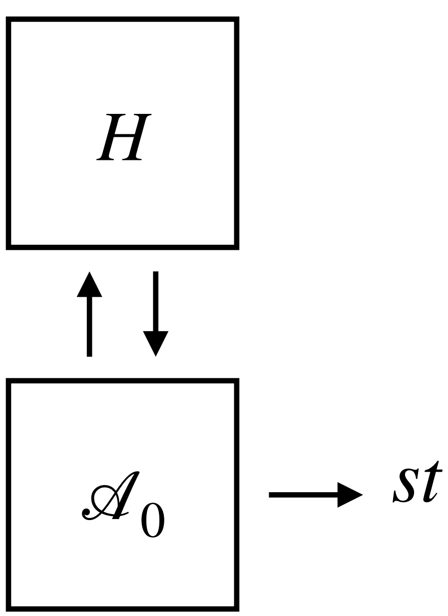
$$y^* \xleftarrow{\$} \{0,1\}^n$$

The game (simplest version)

Uniformly random function $H : \{0,1\}^n \to \{0,1\}^n$

two-stage oracle algorithm $\mathscr{A} = (\mathscr{A}_0, \mathscr{A}_1)$

$$H_{x^* \mapsto y^*}(x) = \begin{cases} y^* & x = x^* \\ H(x) & \text{else} \end{cases}$$



$$b \xleftarrow{\$} \{0,1\}$$

$$x^* \xleftarrow{\$} \{0,1\}^n$$

$$y^* \xleftarrow{\$} \{0,1\}^n$$

The game (simplest version)

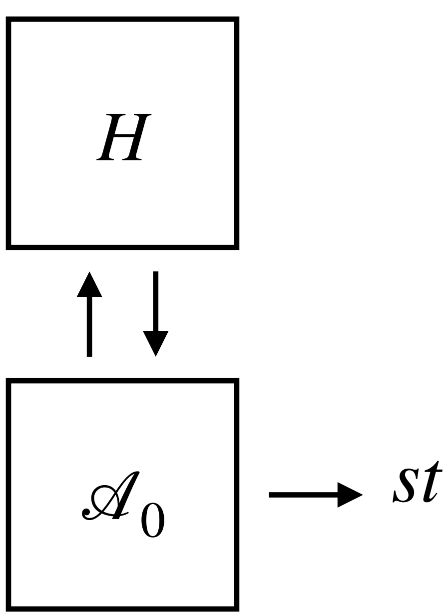
Uniformly random function $H : \{0,1\}^n \to \{0,1\}^n$

two-stage oracle algorithm $\mathscr{A} = (\mathscr{A}_0, \mathscr{A}_1)$

$$H_{x^* \mapsto y^*}(x) = \begin{cases} y^* & x = x^* \\ H(x) & \text{else} \end{cases}$$

$$H^0 = H$$

$$H^1 = H_{x^* \mapsto y^*}$$



$$b \xleftarrow{\$} \{0,1\}$$

$$x^* \xleftarrow{\$} \{0,1\}^n$$

$$y^* \xleftarrow{\$} \{0,1\}^n$$

The game (simplest version)

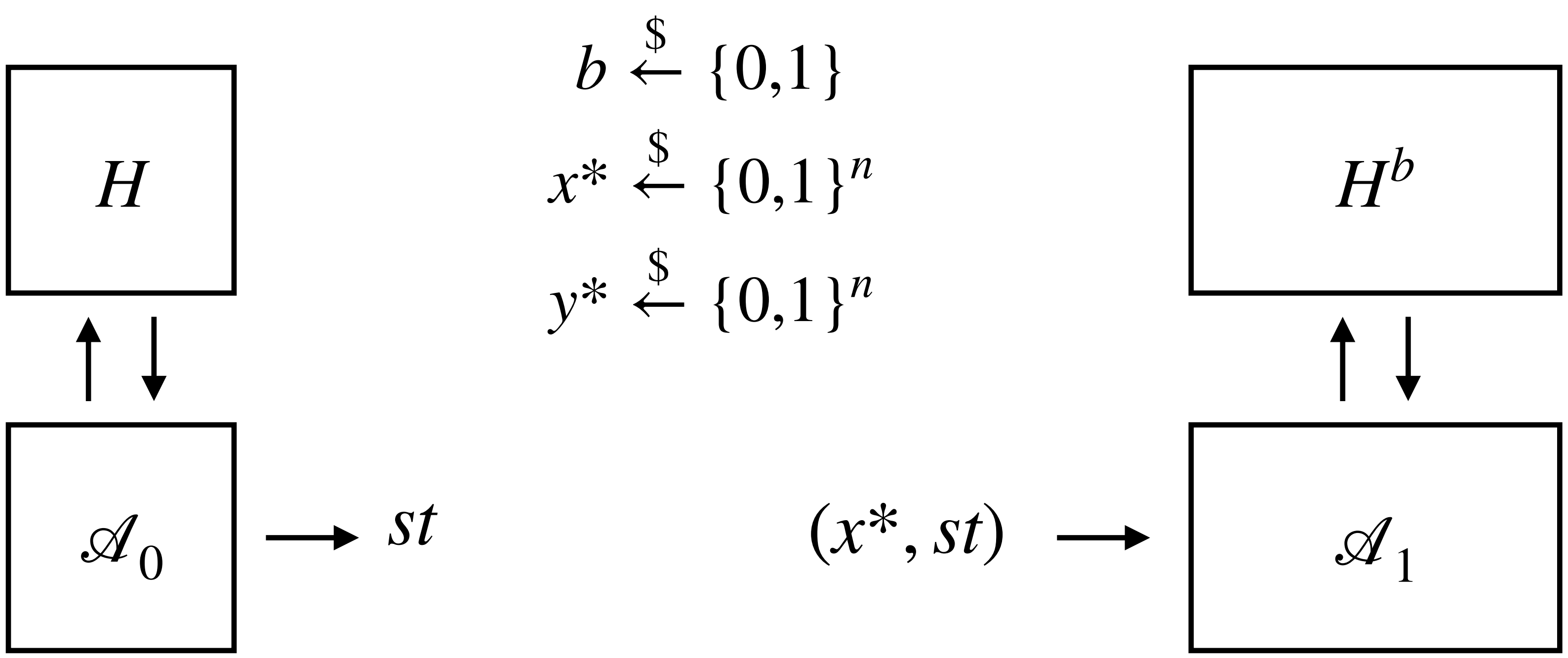
Uniformly random function $H : \{0,1\}^n \to \{0,1\}^n$

two-stage oracle algorithm $\mathscr{A} = (\mathscr{A}_0, \mathscr{A}_1)$

$$H_{x^*\mapsto y^*}(x) = \begin{cases} y^* & x = x^* \\ H(x) & \text{else} \end{cases}$$

$H^0 = H$

$H^1 = H_{x^*\mapsto y^*}$



The game (simplest version)

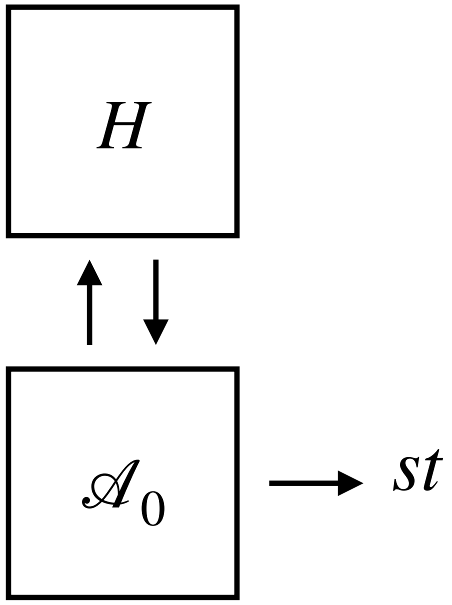
Uniformly random function $H : \{0,1\}^n \to \{0,1\}^n$

two-stage oracle algorithm $\mathscr{A} = (\mathscr{A}_0, \mathscr{A}_1)$

$$H_{x^* \mapsto y^*}(x) = \begin{cases} y^* & x = x^* \\ H(x) & \text{else} \end{cases}$$

$H^0 = H$

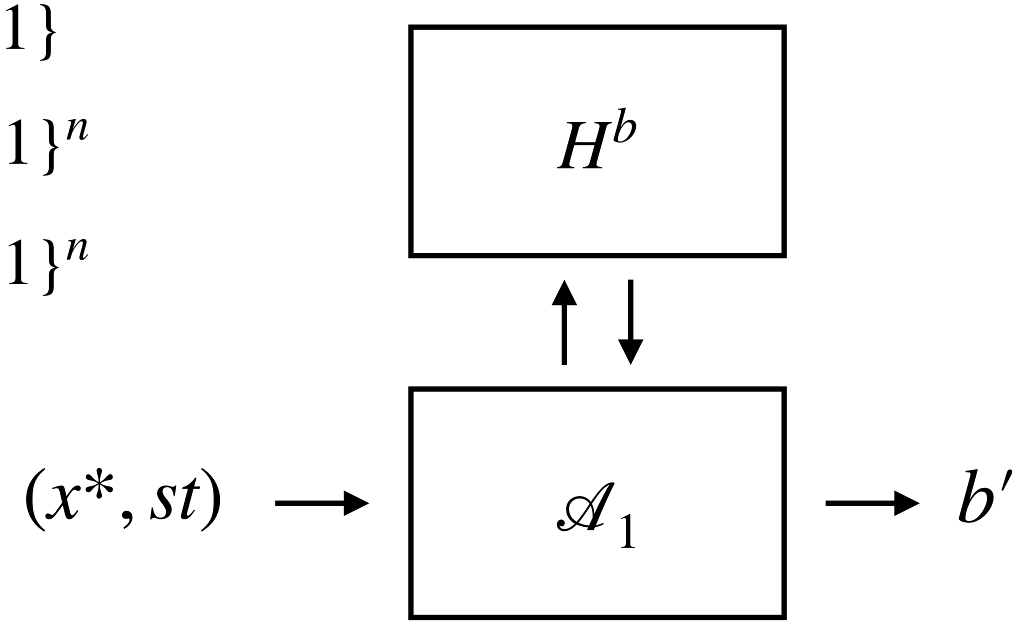
$H^1 = H_{x^* \mapsto y^*}$



$b \xleftarrow{\$} \{0,1\}$

$x^* \xleftarrow{\$} \{0,1\}^n$

$y^* \xleftarrow{\$} \{0,1\}^n$



The game (simplest version)

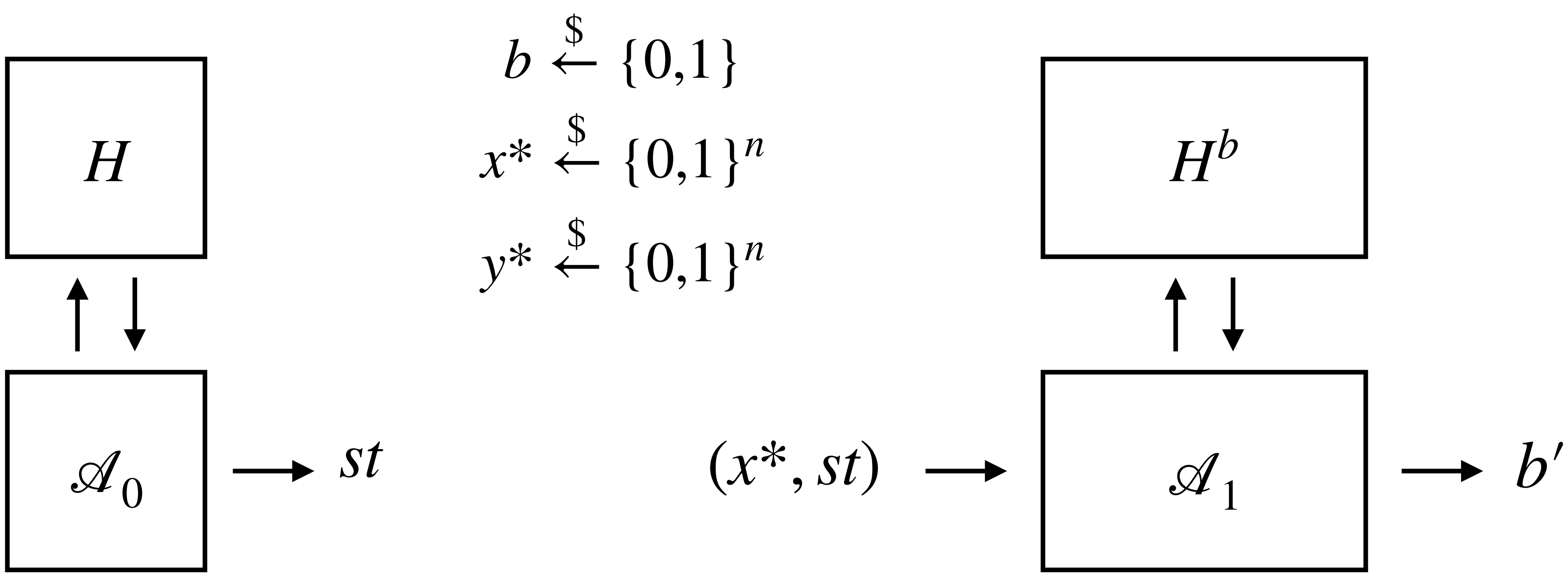
Uniformly random function $H : \{0,1\}^n \to \{0,1\}^n$

two-stage oracle algorithm $\mathscr{A} = (\mathscr{A}_0, \mathscr{A}_1)$

$$H_{x^* \mapsto y^*}(x) = \begin{cases} y^* & x = x^* \\ H(x) & \text{else} \end{cases}$$

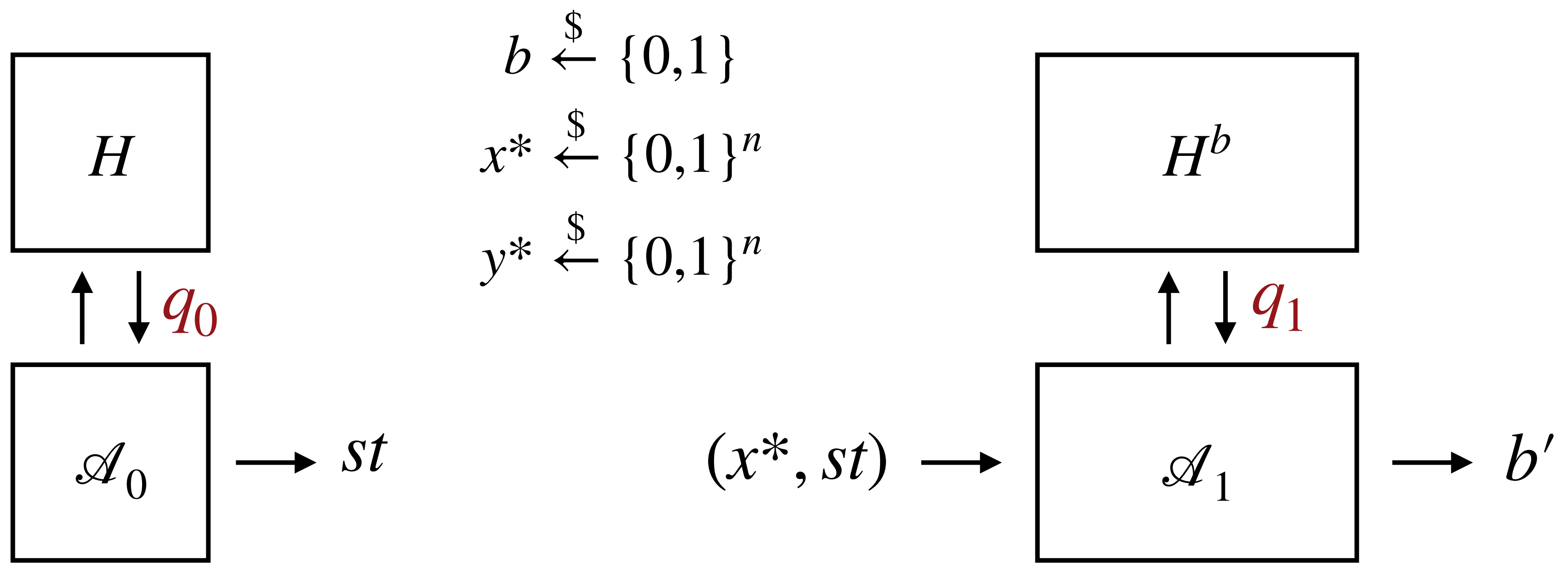
$H^0 = H$

$H^1 = H_{x^* \mapsto y^*}$



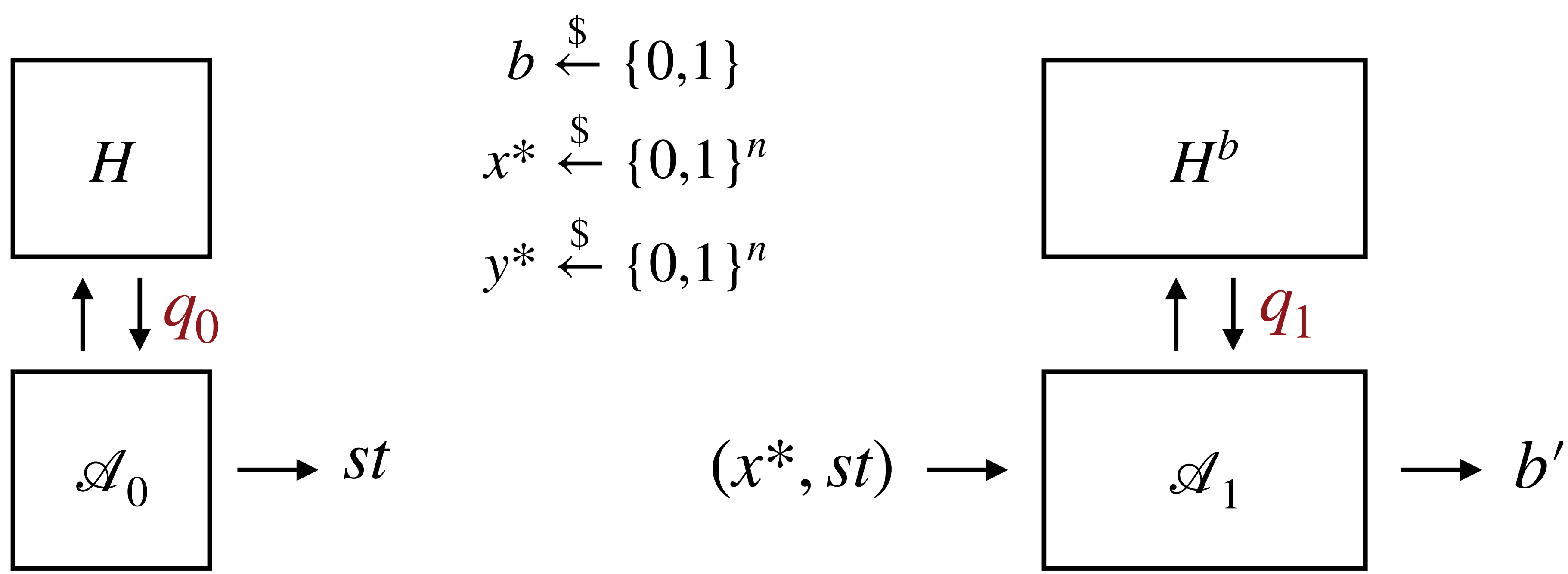
$\mathscr{A}$ wins if $b' = b$

Query lower bound



$\mathscr{A}$ wins if $b' = b$

Query lower bound

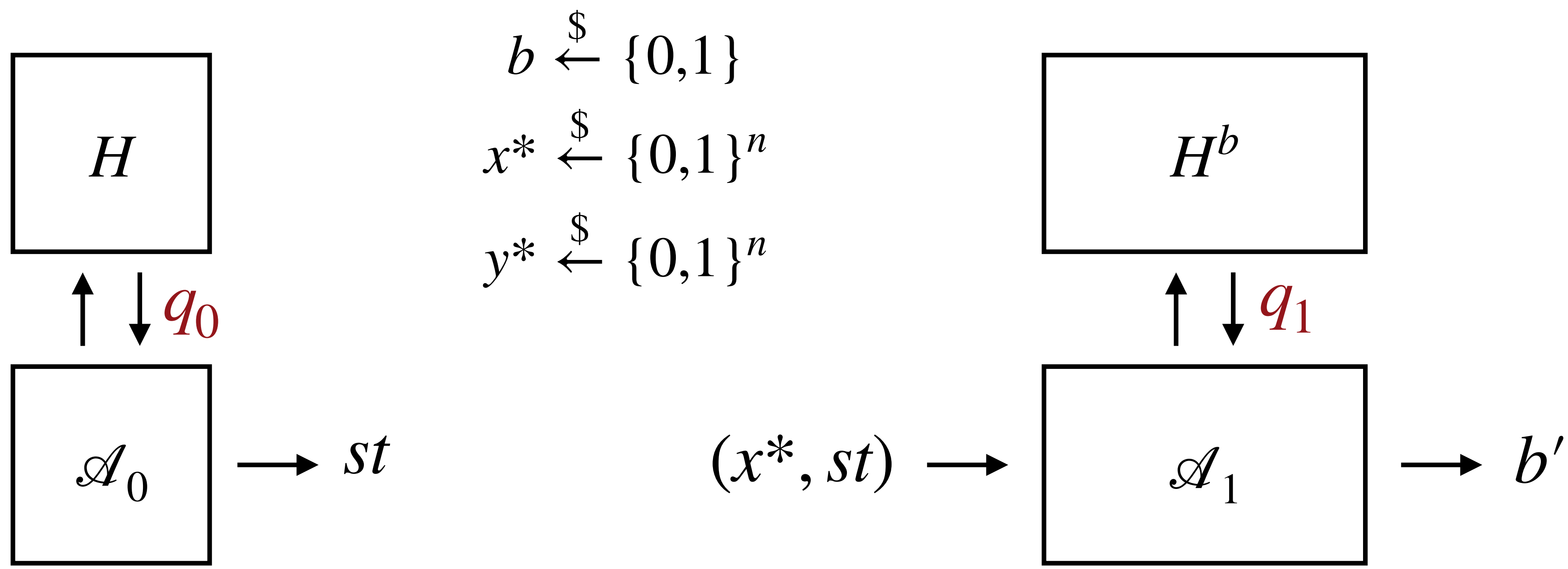


$\mathscr{A}$ wins if $b' = b$

Theorem: For classical $\mathscr{A}$,

$$\Pr[\mathscr{A} \text{ wins}] \leq \frac{1}{2}\left(1 + q_0 2^{-n}\right)$$

Query lower bound



$\mathscr{A}$ wins if $b' = b$

Theorem: For classical $\mathscr{A}$,

$$\Pr[\mathscr{A} \text{ wins}] \leq \frac{1}{2}\left(1 + q_0 2^{-n}\right)$$

This is tight, matching algorithm using $O(q_0)$ time, constant space, $q_1 = q_0$

Applications

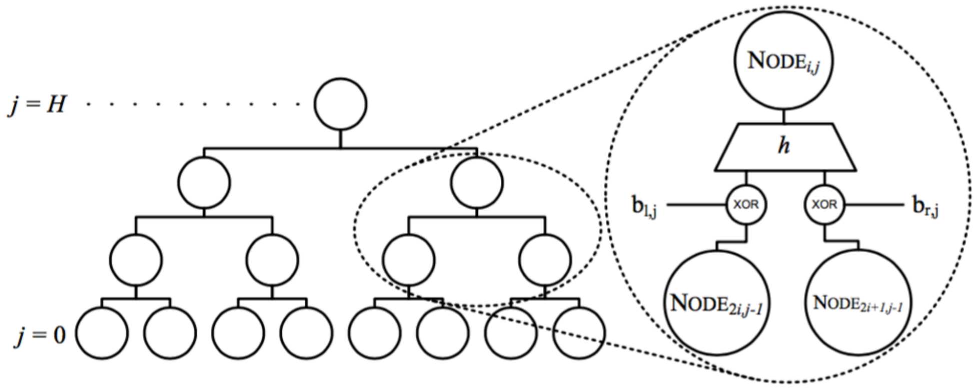
Applications

Security proofs in the ROM for digital signature schemes:

Applications

Security proofs in the ROM for digital signature schemes:

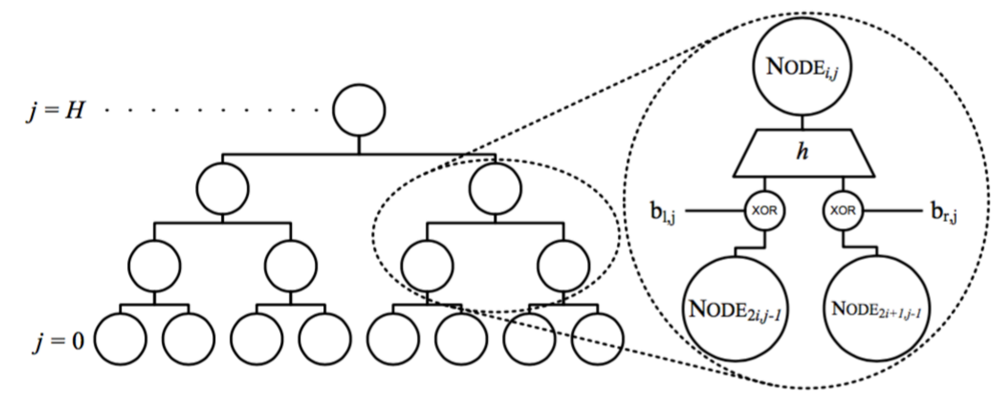
- Hash based signatures (XMSS, standardized as RFC 8391)



Applications

Security proofs in the ROM for digital signature schemes:

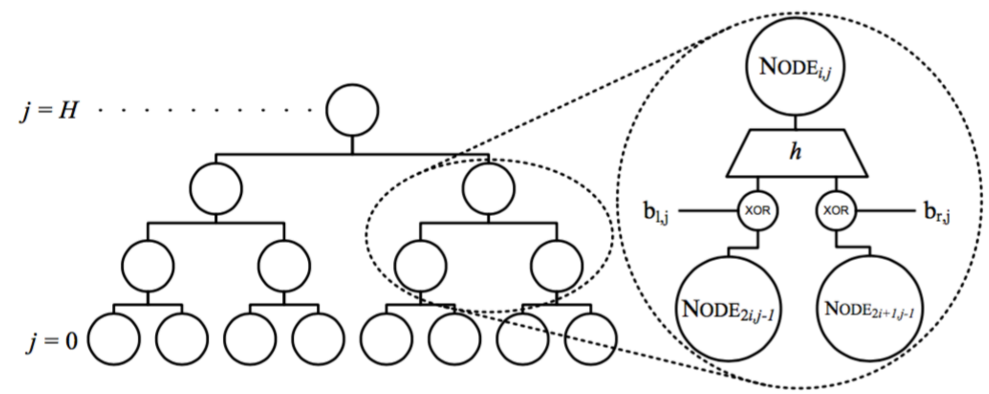
- Hash based signatures (XMSS, standardized as RFC 8391)
- Fiat-Shamir signatures

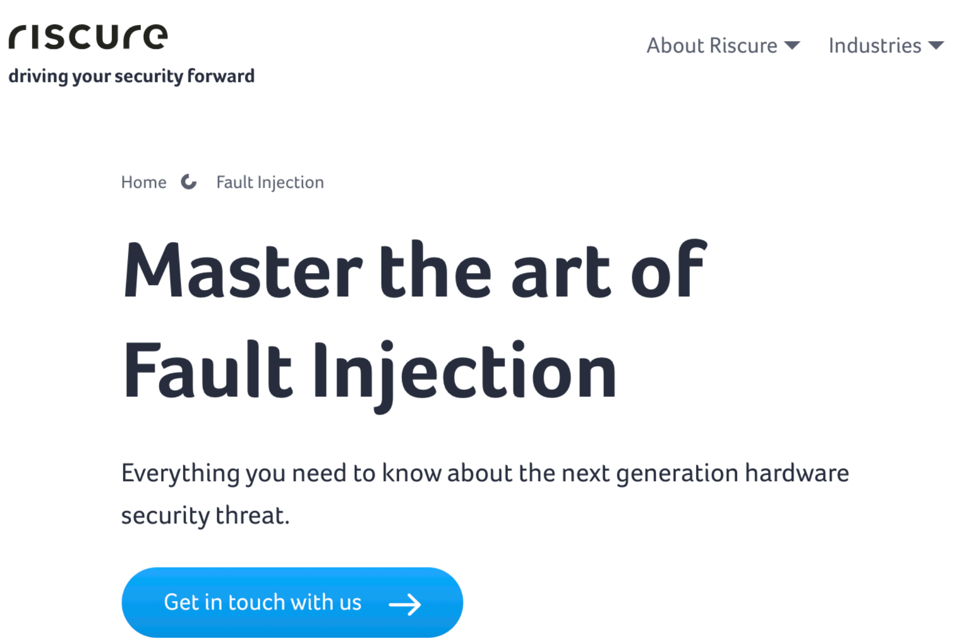


Applications

Security proofs in the ROM for digital signature schemes:

- Hash based signatures (XMSS, standardized as RFC 8391)
- Fiat-Shamir signatures
- The hedged Fiat-Shamir transformation
- etc.

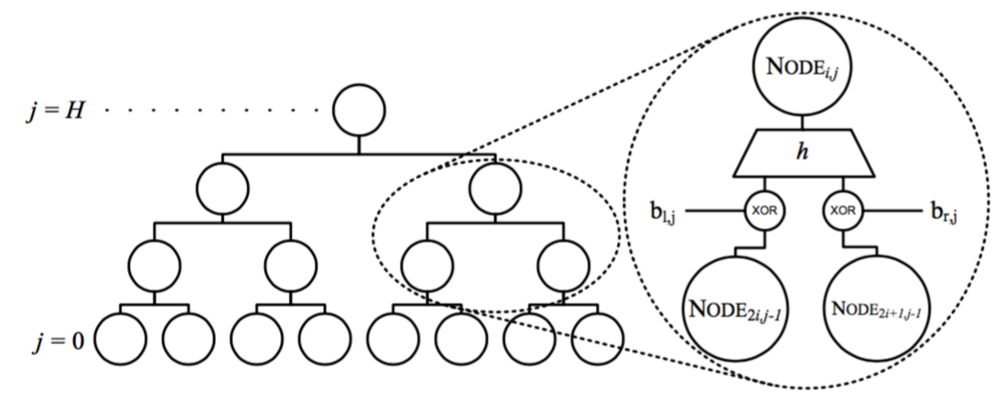


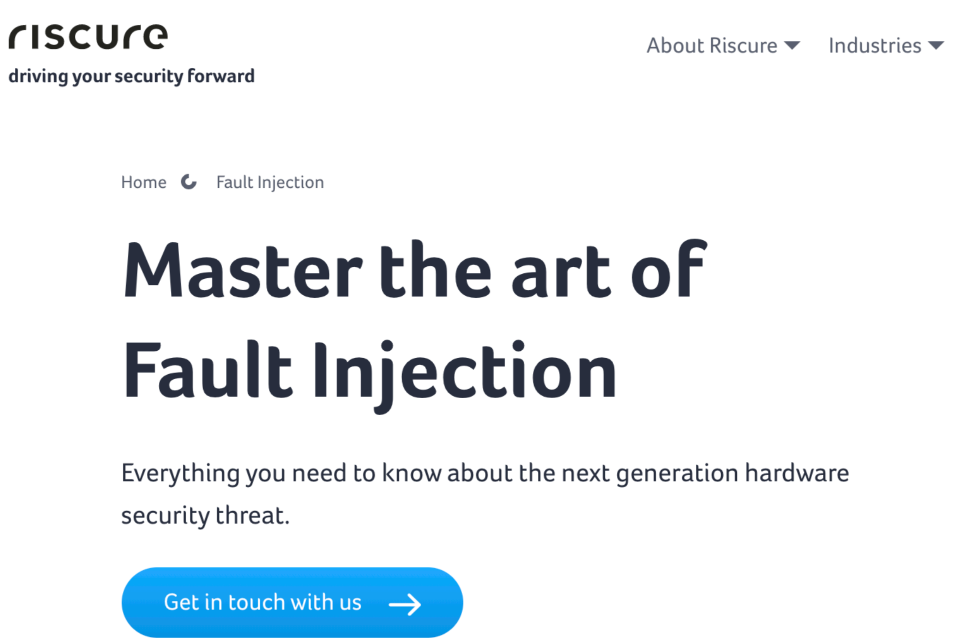


Applications

Security proofs in the ROM for digital signature schemes:

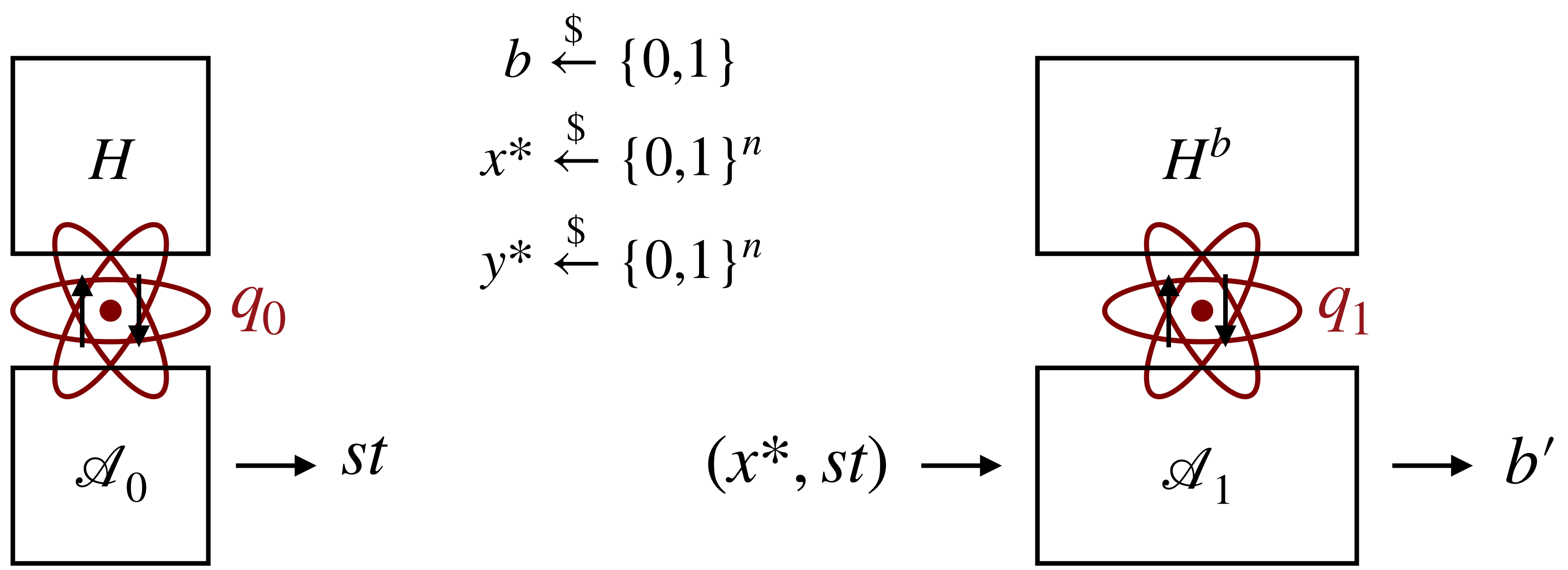
- Hash based signatures (XMSS, standardized as RFC 8391)
- Fiat-Shamir signatures
- The hedged Fiat-Shamir transformation
- etc.





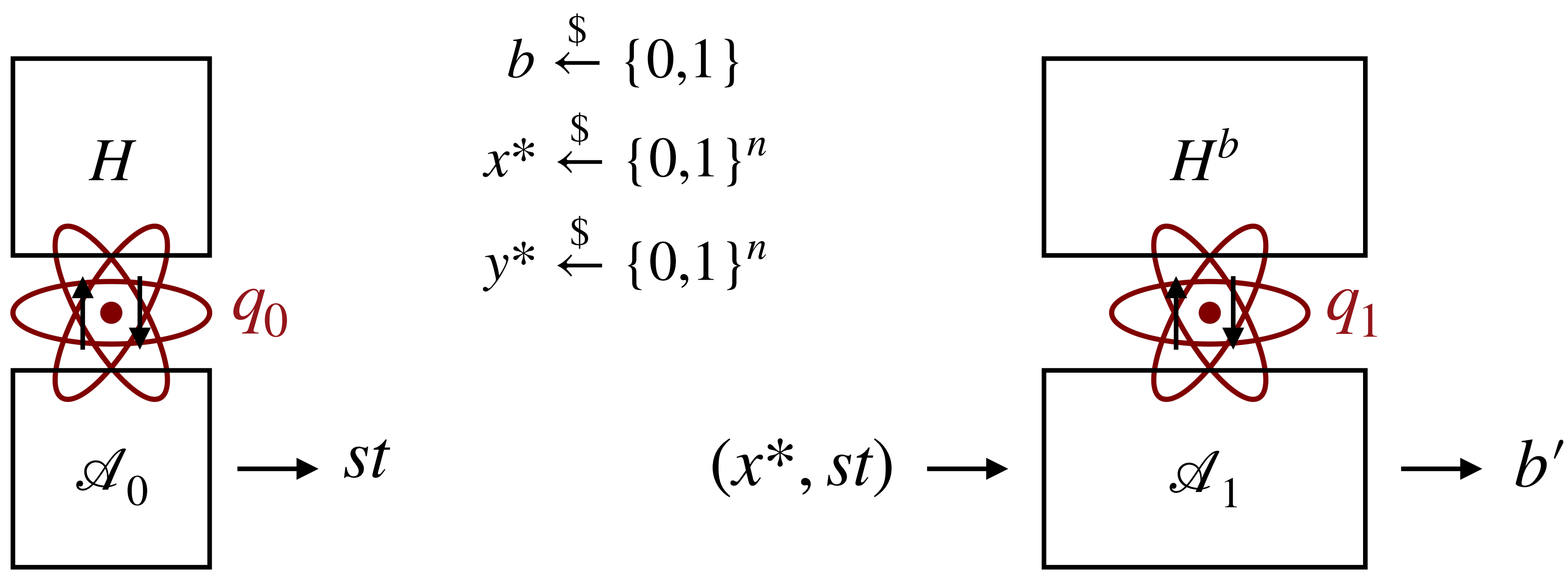
What about post-quantum security?

Quantum query lower bound



$\mathcal{A}$ wins if $b' = b$

Quantum query lower bound

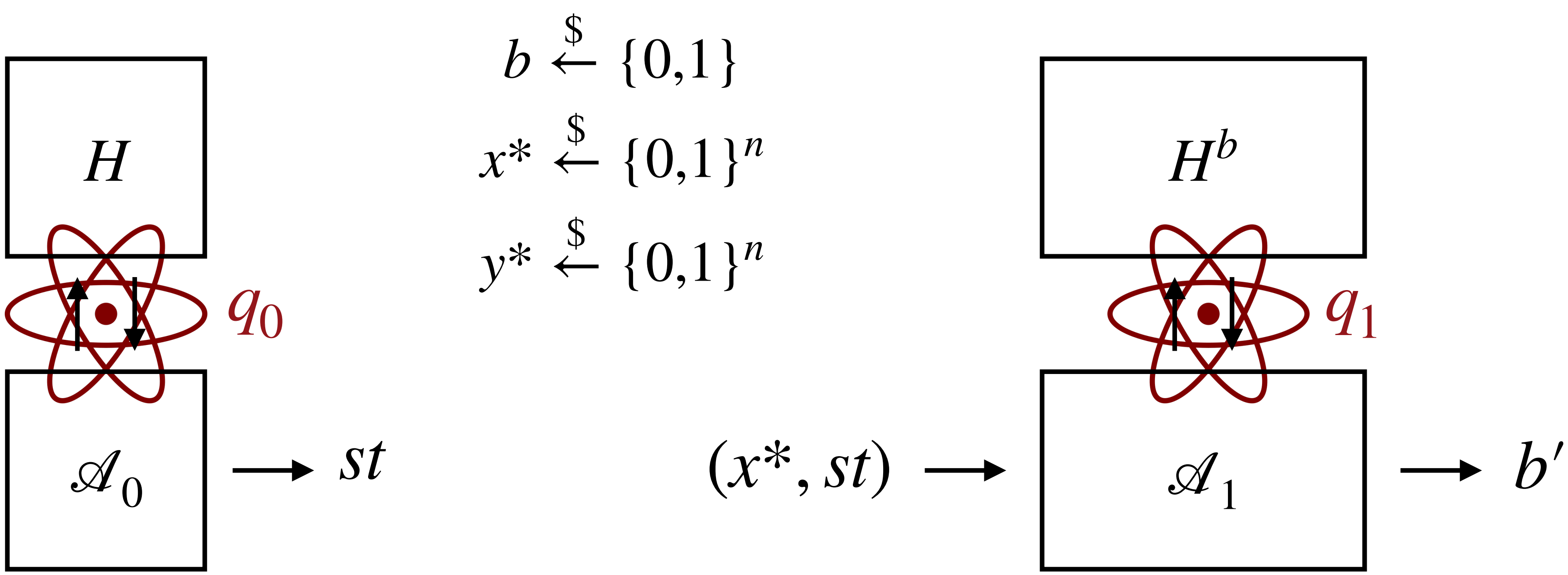


$\mathscr{A}$ wins if $b' = b$

Theorem (Unruh '14):

$$\Pr[\mathscr{A} \text{ wins}] \leq \frac{1}{2} + O\left(q_0 2^{-\frac{n}{2}}\right)$$

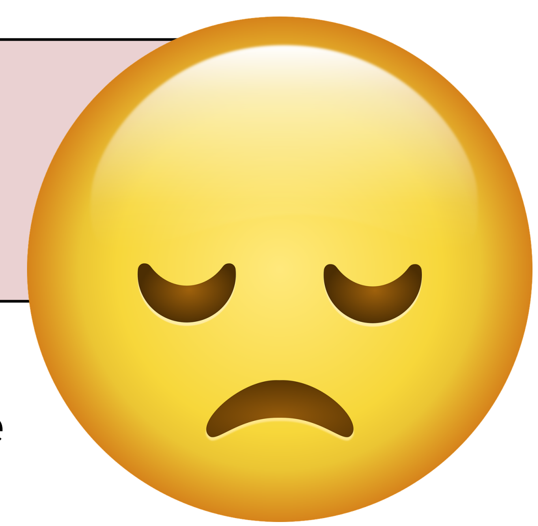
Quantum query lower bound



$\mathscr{A}$ wins if $b' = b$

Theorem (Unruh '14):

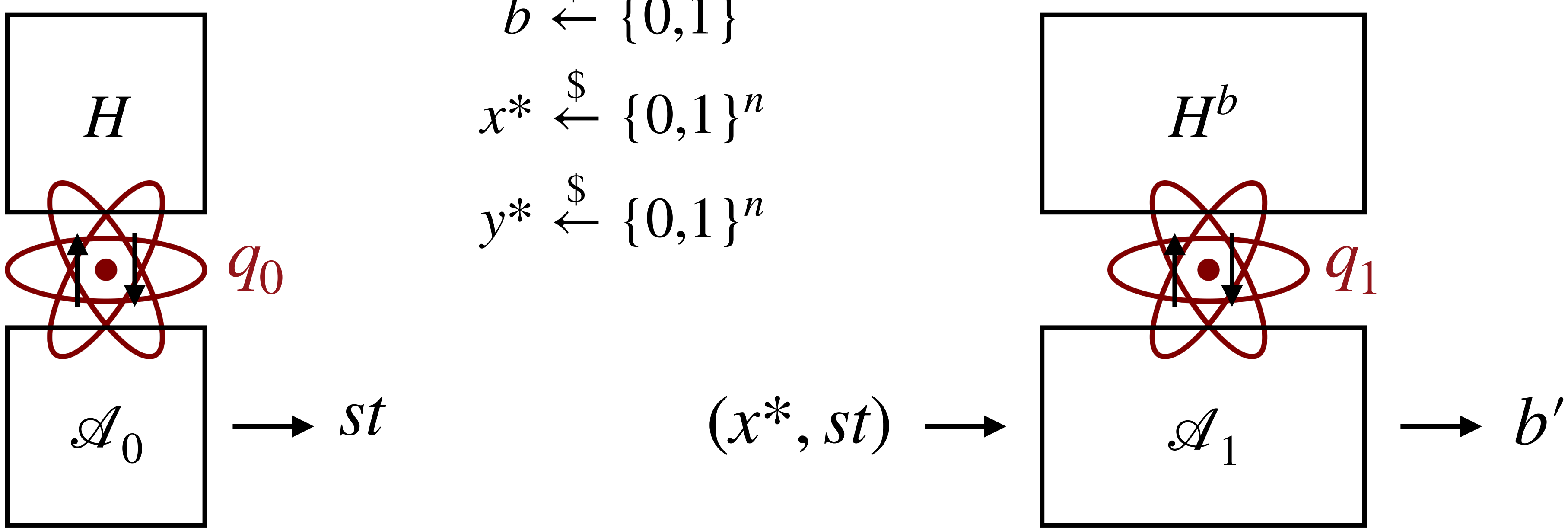
$$\Pr[\mathscr{A} \text{ wins}] \leq \frac{1}{2} + O\left(q_0 2^{-\frac{n}{2}}\right)$$



$q_0 = O\left(2^{\frac{n}{2}}\right)$ allows for constant advantage

Quantum query lower bound

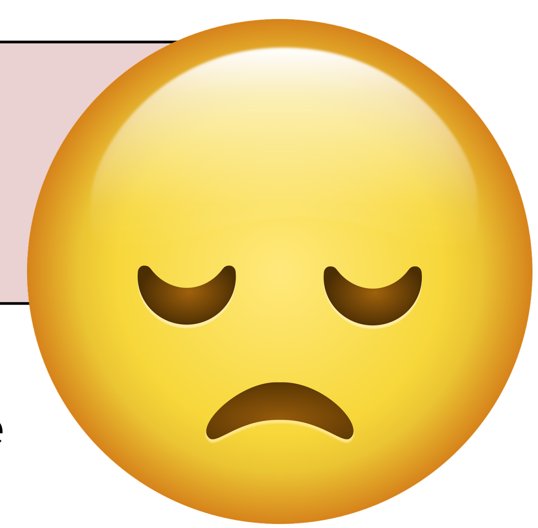
$$b \xleftarrow{\$} \{0,1\}$$
$$x^* \xleftarrow{\$} \{0,1\}^n$$
$$y^* \xleftarrow{\$} \{0,1\}^n$$



$\mathscr{A}$ wins if $b' = b$

Theorem (Unruh '14):

$$\Pr[\mathscr{A} \text{ wins}] \leq \frac{1}{2} + O\left(q_0 2^{-\frac{n}{2}}\right)$$

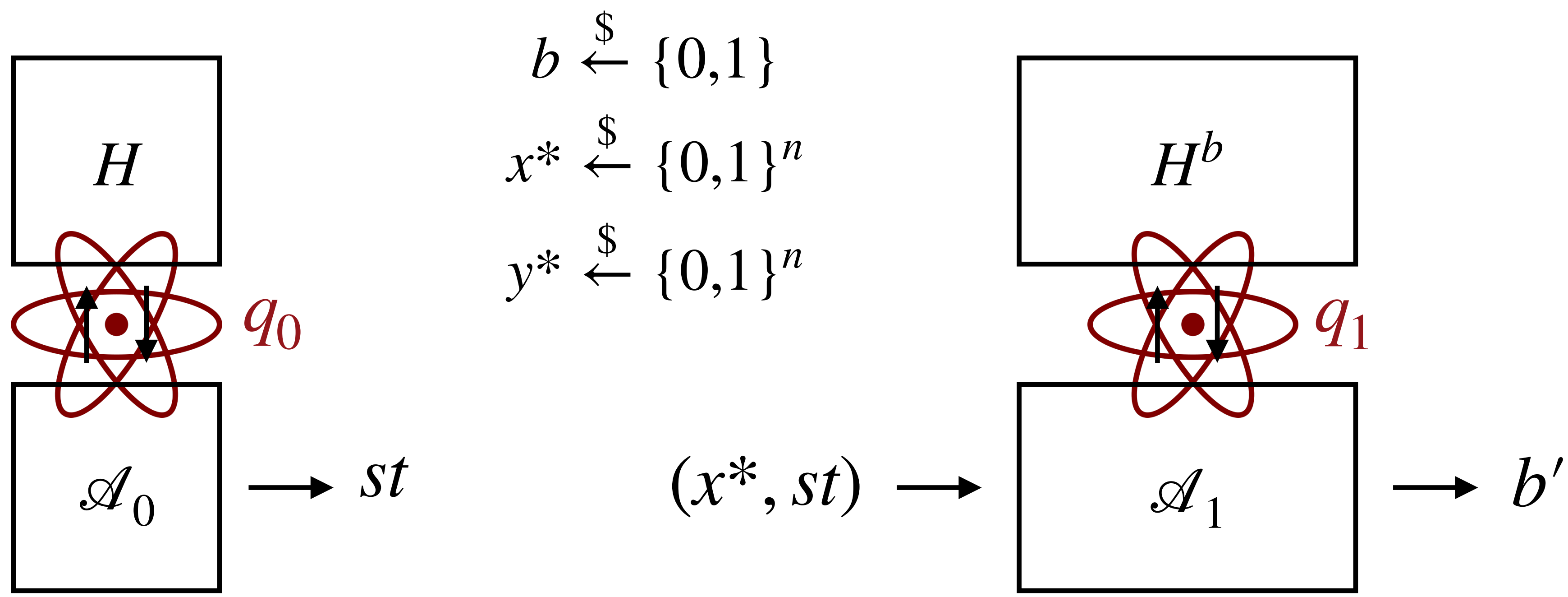


$q_0 = O\left(2^{\frac{n}{2}}\right)$ allows for constant advantage

Tightness unlikely: $\mathscr{A}_0$ doesn't know what it is searching for $\Rightarrow$ no Grover!

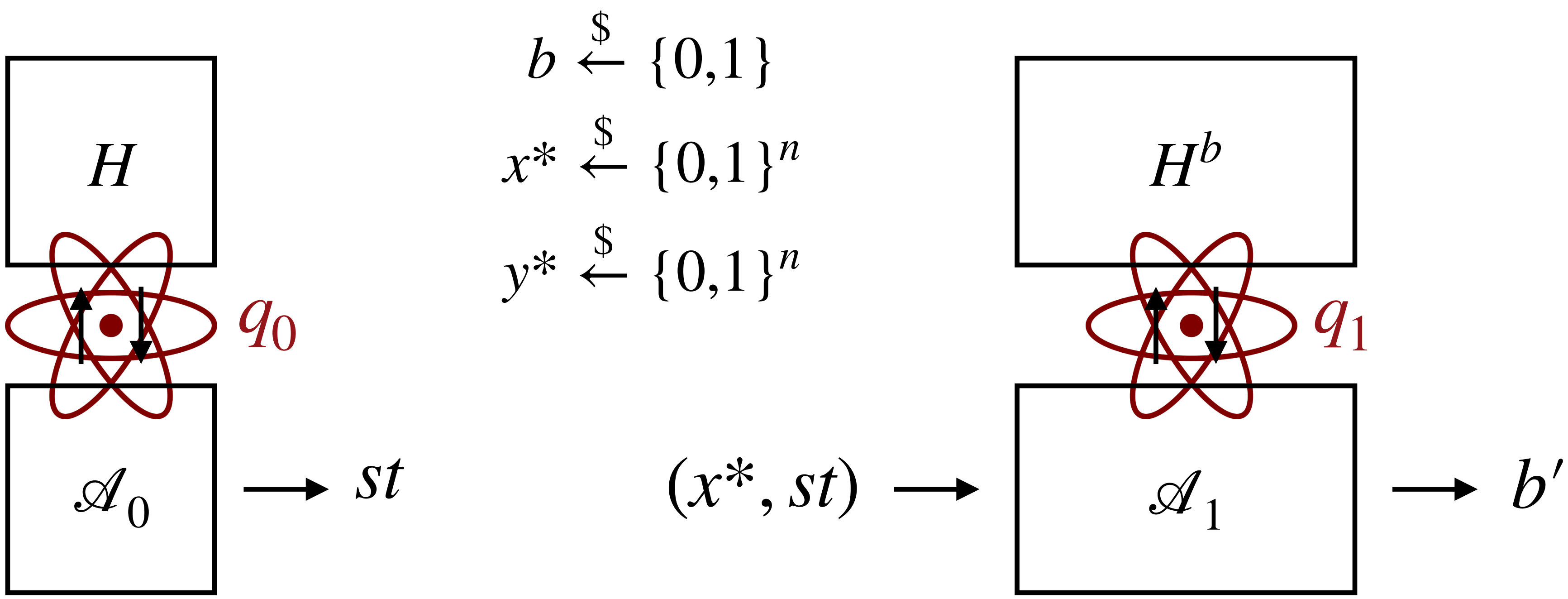
Results

Tight quantum query lower bound



$\mathscr{A}$ wins if $b' = b$

Tight quantum query lower bound

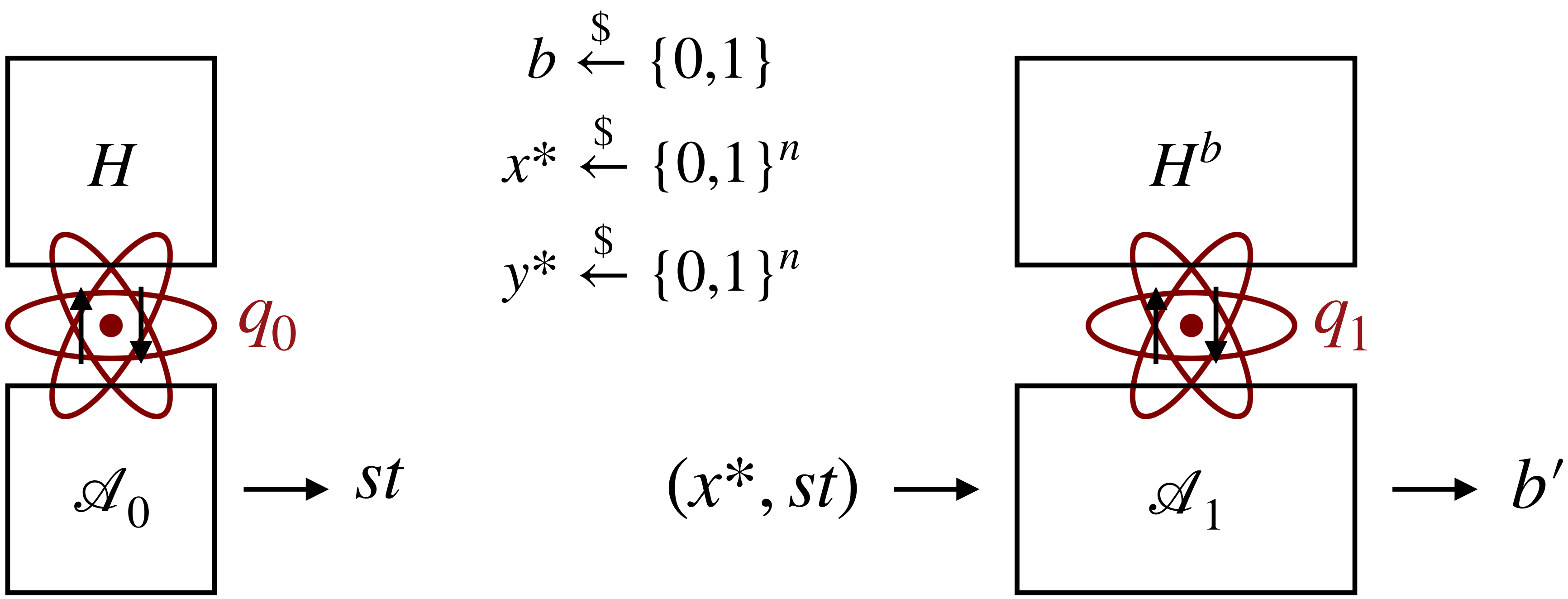


$\mathscr{A}$ wins if $b' = b$

Theorem (Grilo, Hövelmanns, Hülsing, CM):

$$\Pr[\mathscr{A} \text{ wins}] \leq \frac{1}{2} + \frac{3}{2}\sqrt{q_0 2^{-n}}$$

Tight quantum query lower bound



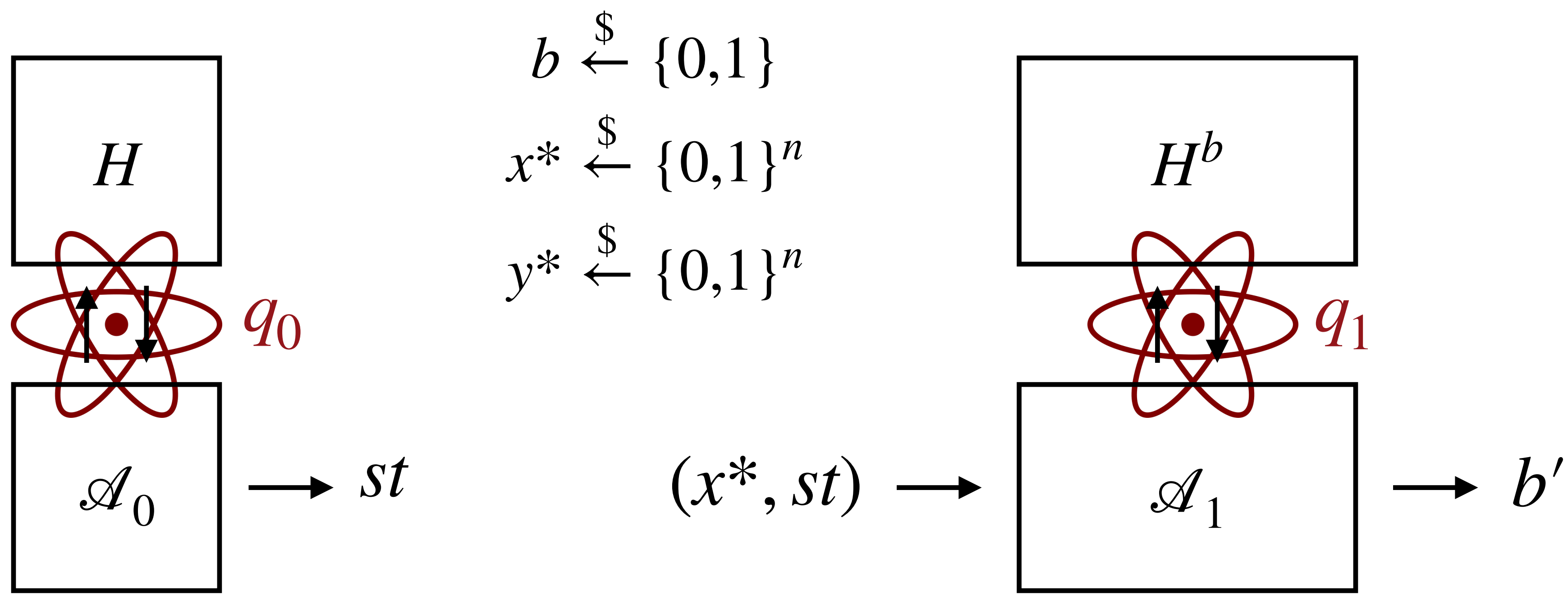
$\mathscr{A}$ wins if $b' = b$

Theorem (Grilo, Hövelmanns, Hülsing, CM):

$$\Pr[\mathscr{A} \text{ wins}] \leq \frac{1}{2} + \frac{3}{2}\sqrt{q_0 2^{-n}}$$

$q_0 = \Omega\left(2^n\right)$ necessary for constant advantage

Tight quantum query lower bound



$\mathscr{A}$ wins if $b' = b$

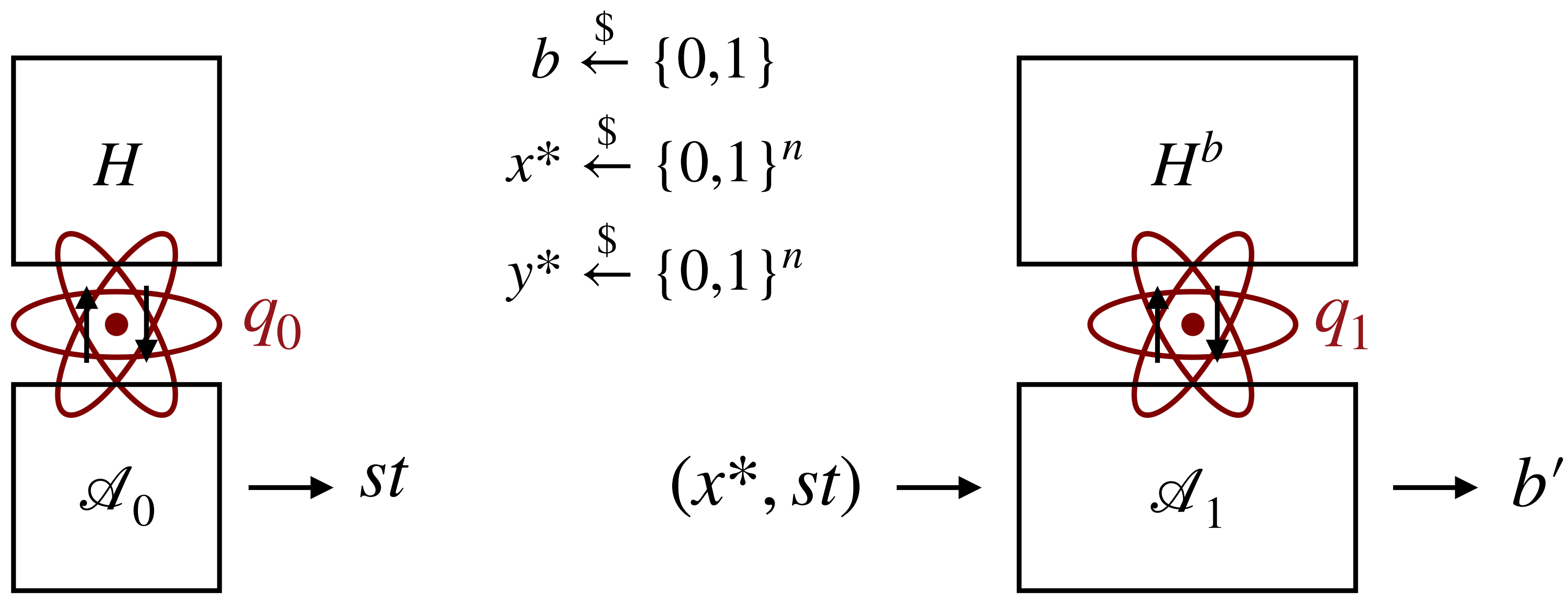
Theorem (Grilo, Hövelmanns, Hülsing, CM):

$$\Pr[\mathscr{A}\text{ wins}] \leq \frac{1}{2} + \frac{3}{2}\sqrt{q_0 2^{-n}}$$

+ some generalizations

$q_0 = \Omega\left(2^n\right)$ necessary for constant advantage

Tight quantum query lower bound



$\mathscr{A}$ wins if $b' = b$

Theorem (Grilo, Hövelmanns, Hülsing, CM):

$$\Pr[\mathscr{A} \text{ wins}] \leq \frac{1}{2} + \frac{3}{2}\sqrt{q_0 2^{-n}}$$

+ some generalizations

$q_0 = \Omega\left(2^n\right)$ necessary for constant advantage

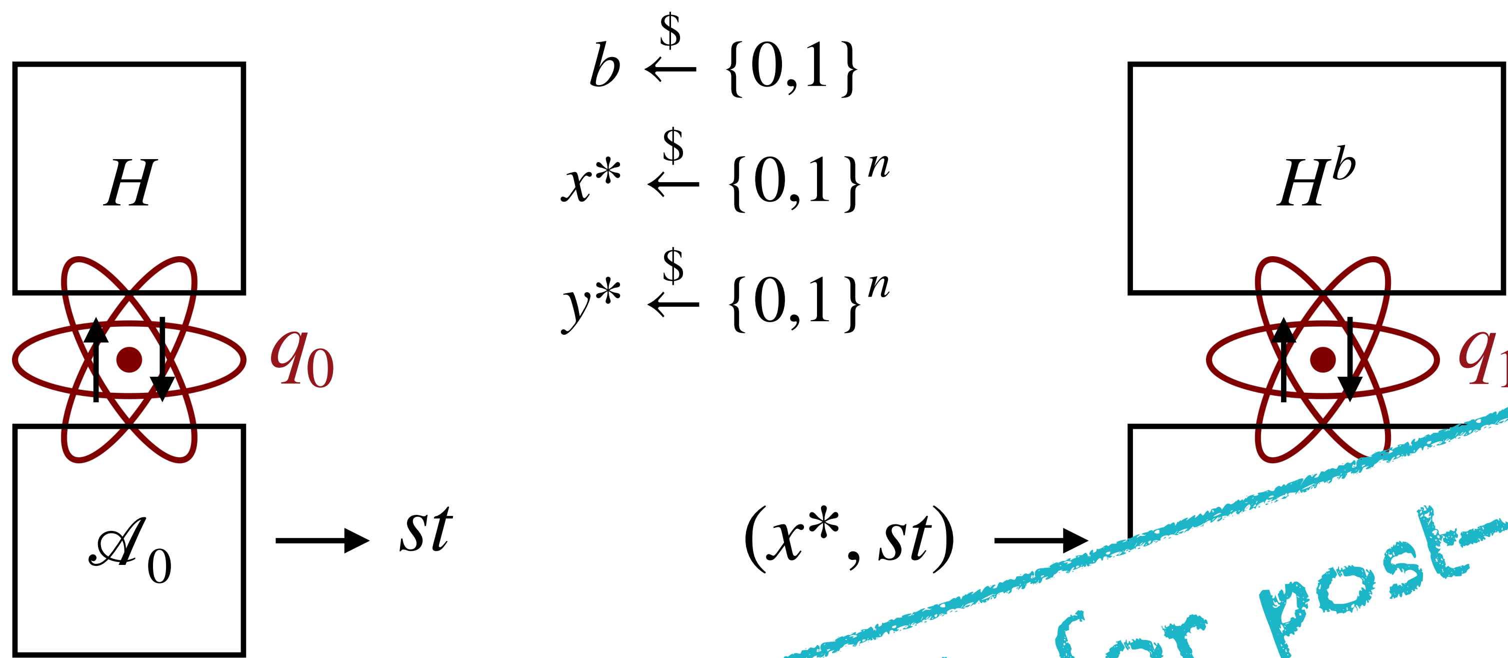
Tightness:

Theorem (Grilo, Hövelmanns, Hülsing, CM):

There exists a quantum algorithm that achieves

$$\Pr[\mathscr{A} \text{ wins}] = \frac{1}{2} + \Omega\left(\sqrt{q_0 2^{-n}}\right)$$

Tight quantum query lower bound



$\mathscr{A}$ wins if $b' = b$

Tighter security proofs for post-quantum signatures, including NIST candidates.

Th… …n):

… $\leq \frac{1}{2} + \frac{3}{2}\sqrt{q_0 2^{-n}}$

+ some generalizations

$q_0 = \Omega\left(2^n\right)$ necessary for constant advantage

Tightness:

Theorem (Grilo, Hövelmanns, Hülsing, CM):
There exists a quantum algorithm that achieves

$$\Pr[\mathscr{A} \text{ wins}] = \frac{1}{2} + \Omega\left(\sqrt{q_0 2^{-n}}\right)$$

Reprogramming superposition oracles

The superposition oracle

For simplicity: $H : \{0,1\}^n \to \{0,1\}^n$

Random oracle

Superposition oracle (Zhandry '18)

The superposition oracle

For simplicity: $H : \{0,1\}^n \rightarrow \{0,1\}^n$

Random oracle

For each $x \in \{0,1\}^n$:
$H(x) \leftarrow \{0,1\}^n$

Superposition oracle (Zhandry '18)

The superposition oracle

For simplicity: $H : \{0,1\}^n \rightarrow \{0,1\}^n$

Random oracle

For each $x \in \{0,1\}^n$:
$H(x) \leftarrow \{0,1\}^n$

Superposition oracle (Zhandry '18)

For each $x \in \{0,1\}^n$:
Initialize n-qubit register F_x
in state $|\phi_0\rangle = |+\rangle^{\otimes n}$

The superposition oracle

For simplicity: $H : \{0,1\}^n \to \{0,1\}^n$

Random oracle

For each $x \in \{0,1\}^n$:
$H(x) \leftarrow \{0,1\}^n$

Query unitary:
$U_H |x\rangle_X |y\rangle_Y = |x\rangle_X |y \oplus H(x)\rangle_Y$

Superposition oracle (Zhandry '18)

For each $x \in \{0,1\}^n$:
Initialize n-qubit register F_x
in state $|\phi_0\rangle = |+\rangle^{\otimes n}$

The superposition oracle

For simplicity: $H : \{0,1\}^n \to \{0,1\}^n$

Random oracle

For each $x \in \{0,1\}^n$:
$H(x) \leftarrow \{0,1\}^n$

Query unitary:
$U_H |x\rangle_X |y\rangle_Y = |x\rangle_X |y \oplus H(x)\rangle_Y$

Superposition oracle (Zhandry '18)

For each $x \in \{0,1\}^n$:
Initialize n-qubit register F_x
in state $|\phi_0\rangle = |+\rangle^{\otimes n}$

Query unitary:
$U_H |x\rangle_X = \mathrm{CNOT}^{\otimes n}_{F_x:Y}$

The superposition oracle

For simplicity: $H : \{0,1\}^n \to \{0,1\}^n$

Random oracle

For each $x \in \{0,1\}^n$:
$H(x) \leftarrow \{0,1\}^n$

Query unitary:
$U_H |x\rangle_X |y\rangle_Y = |x\rangle_X |y \oplus H(x)\rangle_Y$

Reprogramming at x^*: $y^* \leftarrow \{0,1\}^n$,
$$H'(x) = \begin{cases} y^* & x = x^* \\ H(x) & \text{else} \end{cases}$$

Superposition oracle (Zhandry '18)

For each $x \in \{0,1\}^n$:
Initialize n-qubit register F_x
in state $|\phi_0\rangle = |+\rangle^{\otimes n}$

Query unitary:
$U_H |x\rangle_X = \mathrm{CNOT}^{\otimes n}_{F_x:Y}$

The superposition oracle

For simplicity: $H : \{0,1\}^n \rightarrow \{0,1\}^n$

Random oracle

For each $x \in \{0,1\}^n$:

$H(x) \leftarrow \{0,1\}^n$

Query unitary:

$U_H|x\rangle_X|y\rangle_Y = |x\rangle_X|y \oplus H(x)\rangle_Y$

Reprogramming at x^*: $y^* \leftarrow \{0,1\}^n$,

$$H'(x) = \begin{cases} y^* & x = x^* \\ H(x) & \text{else} \end{cases}$$

Superposition oracle (Zhandry '18)

For each $x \in \{0,1\}^n$:

Initialize n-qubit register F_x

in state $|\phi_0\rangle = |+\rangle^{\otimes n}$

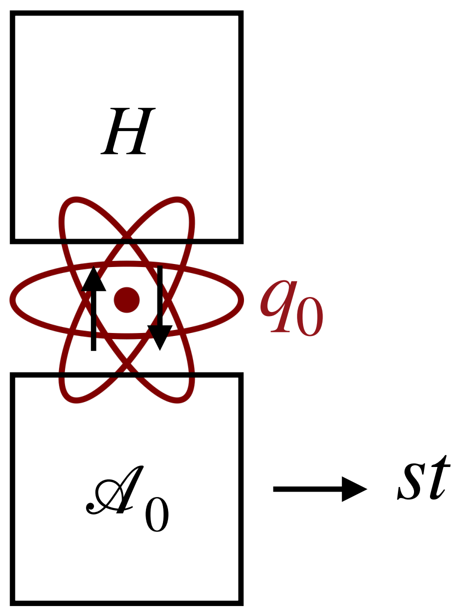
Query unitary:

$U_H|x\rangle_X = \mathrm{CNOT}^{\otimes n}_{F_x:Y}$

Reprogramming at x^*:

- Discard contents of F_{x^*}
- Prepare F_{x^*} in state $|\phi_0\rangle$

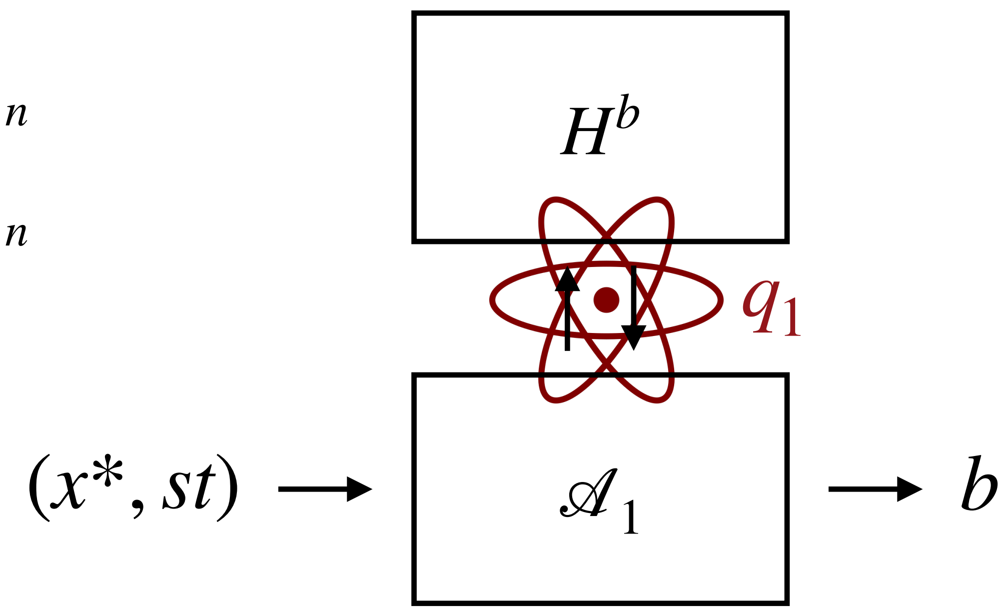
Proof ideas



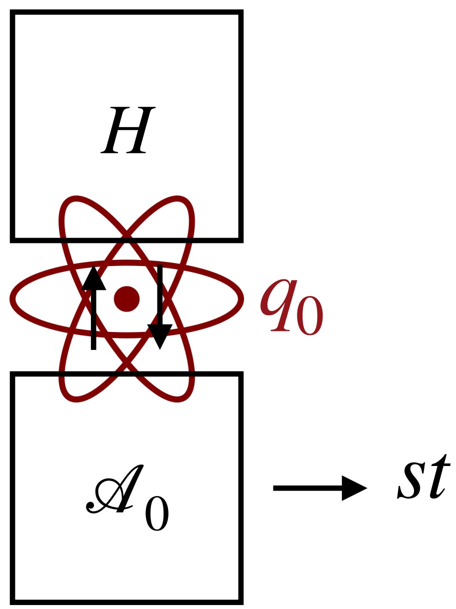
$b \xleftarrow{\$} \{0,1\}$

$x^* \xleftarrow{\$} \{0,1\}^n$

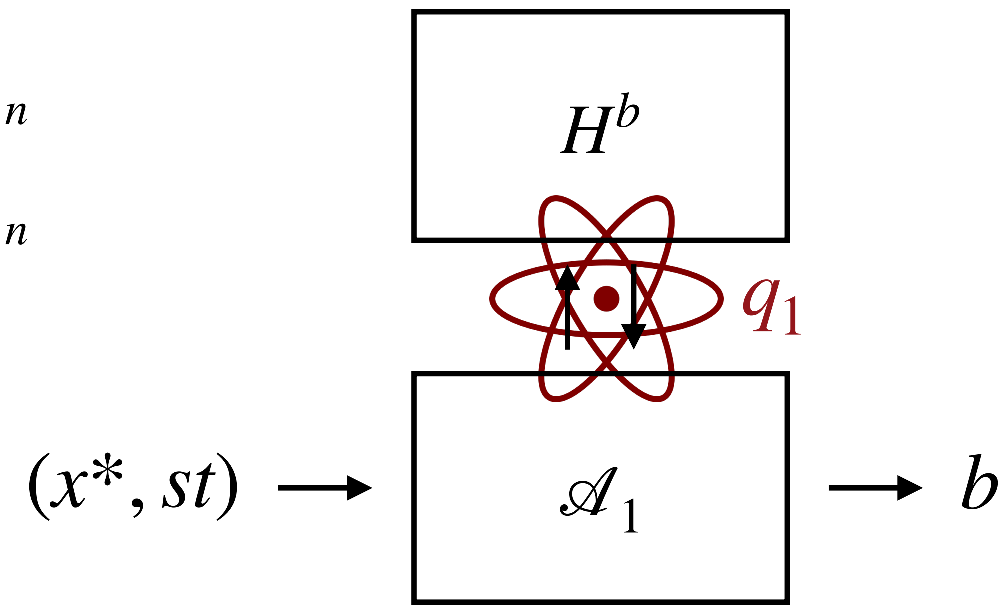
$y^* \xleftarrow{\$} \{0,1\}^n$



Proof ideas

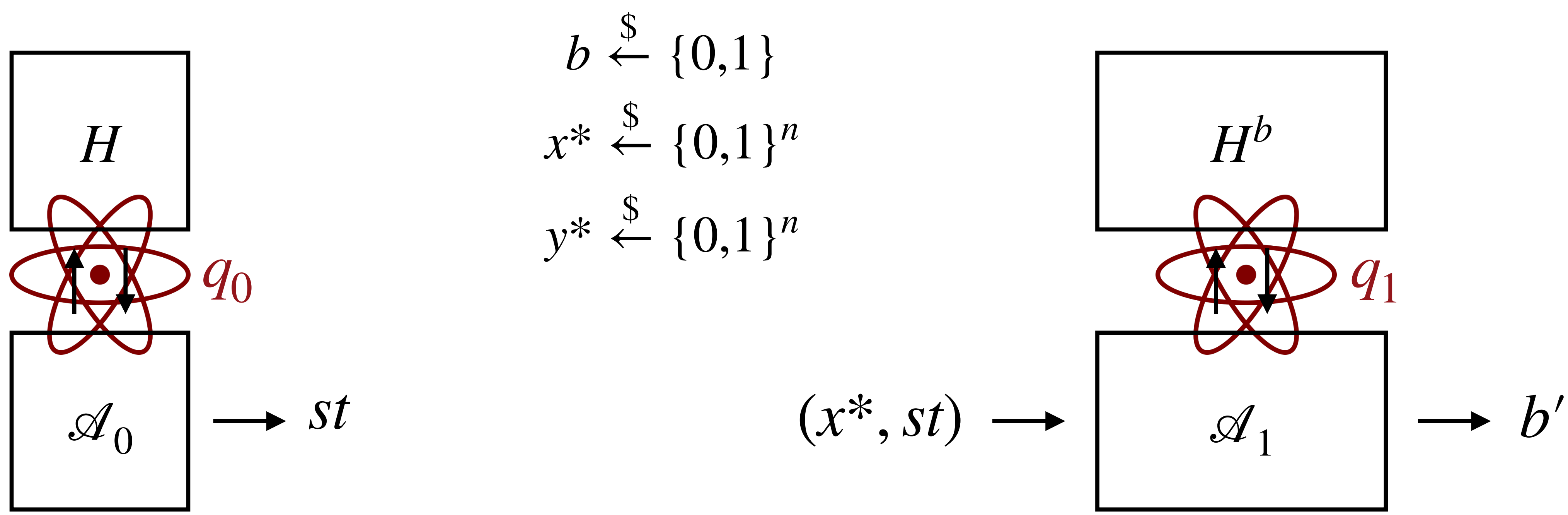


$$b \xleftarrow{\$} \{0,1\}$$
$$x^* \xleftarrow{\$} \{0,1\}^n$$
$$y^* \xleftarrow{\$} \{0,1\}^n$$



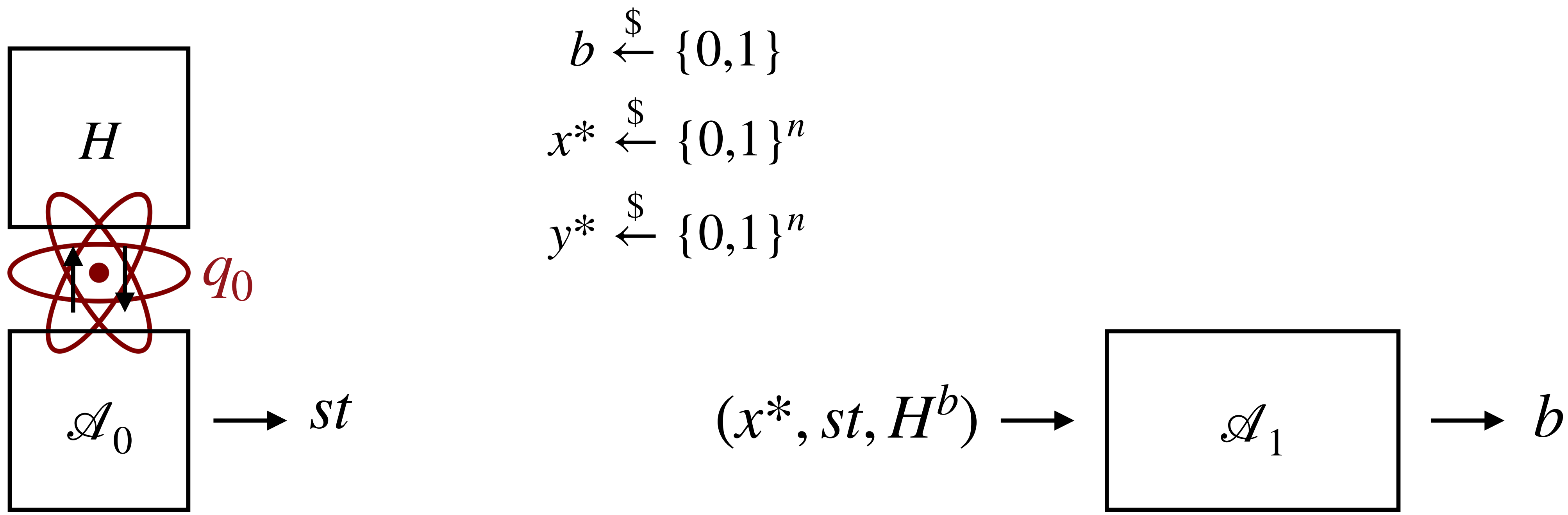
- Intuition: q_0 is limiting quantity

Proof ideas



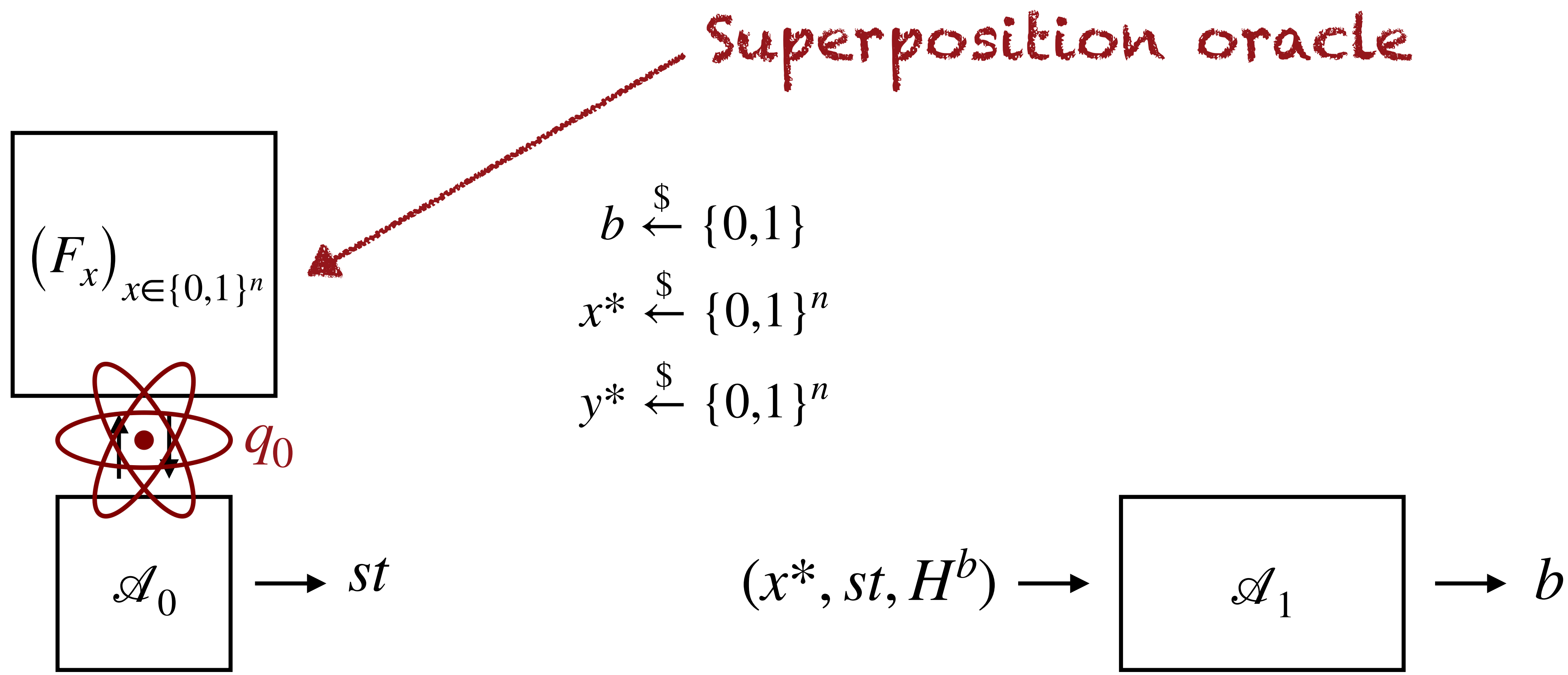
- Intuition: q_0 is limiting quantity
- simplification: allow $q_1 = 2^n$

Proof ideas



- Intuition: q_0 is limiting quantity
- simplification: allow $q_1 = 2^n$

Proof ideas

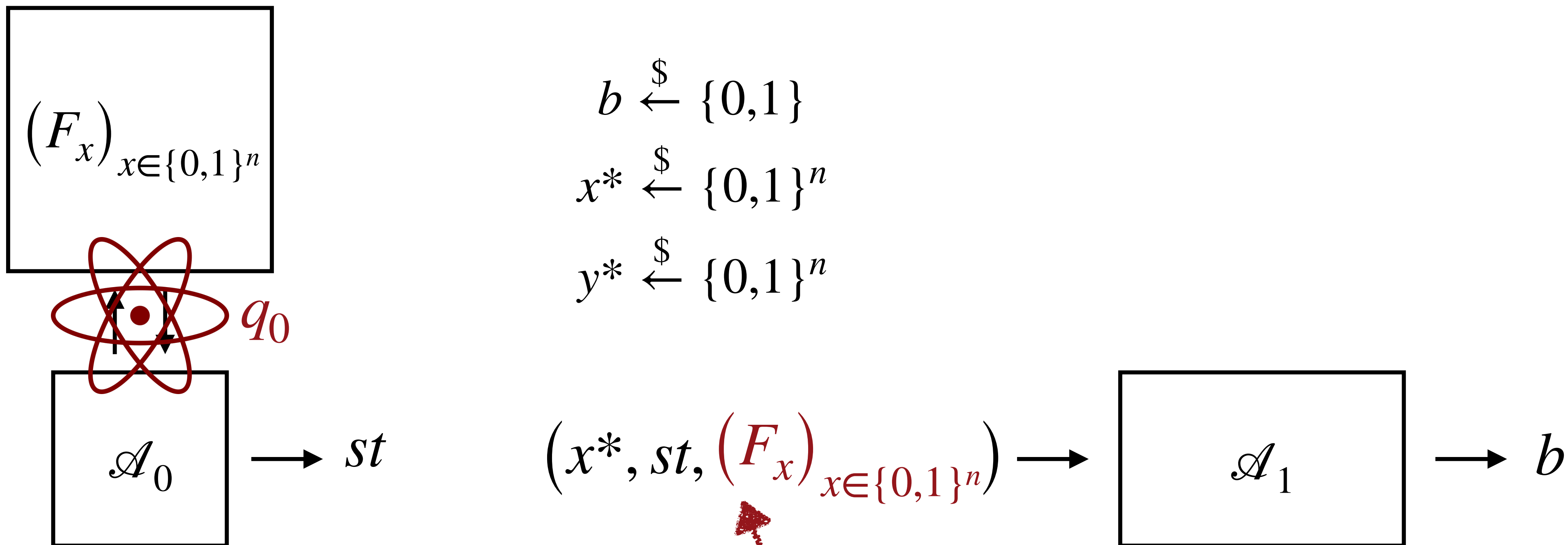


- Intuition: q_0 is limiting quantity
- simplification: allow $q_1 = 2^n$

For each $x \in \{0,1\}^n$:
Initialize n-qubit register F_x
in state $|\phi_0\rangle = |+\rangle^{\otimes n}$

Query unitary:
$U_H|x\rangle_X = \mathrm{CNOT}^{\otimes n}_{F_x:Y}$

Proof ideas



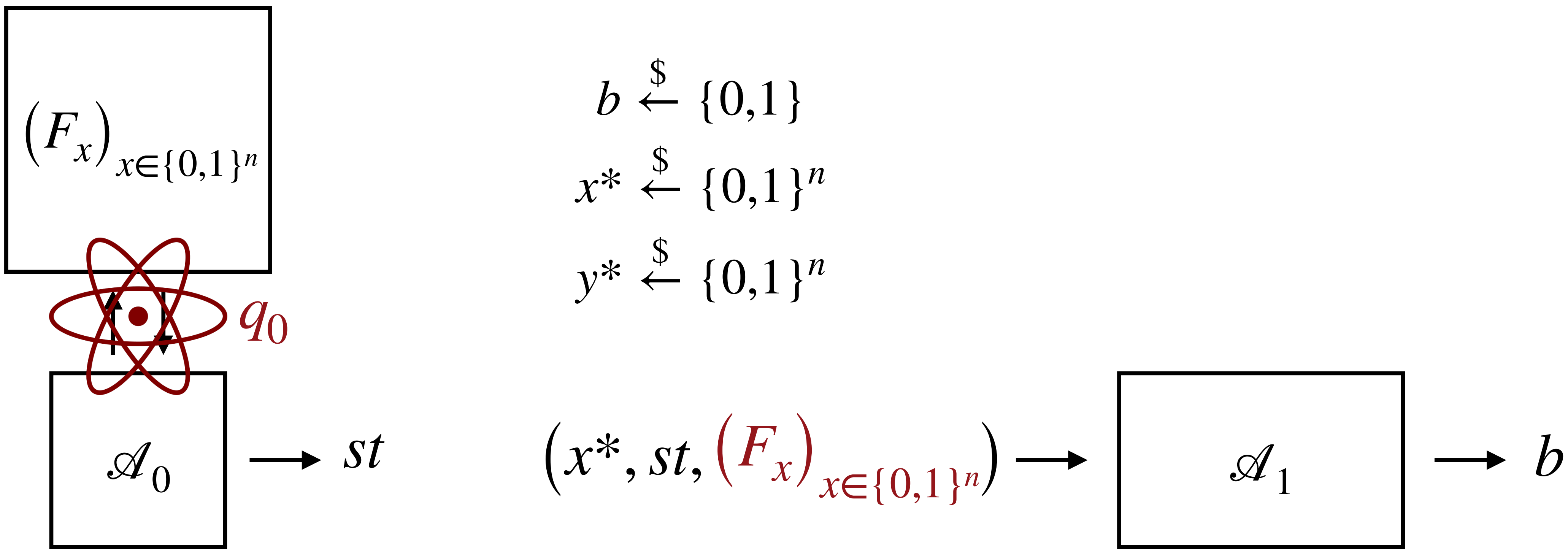
Hand over oracle's internal state after potentially reprogramming:

Reprogramming at x^*:

- Discard contents of F_{x^*}
- Prepare F_{x^*} in state $|\phi_0\rangle$

- Intuition: q_0 is limiting quantity
- simplification: allow $q_1 = 2^n$

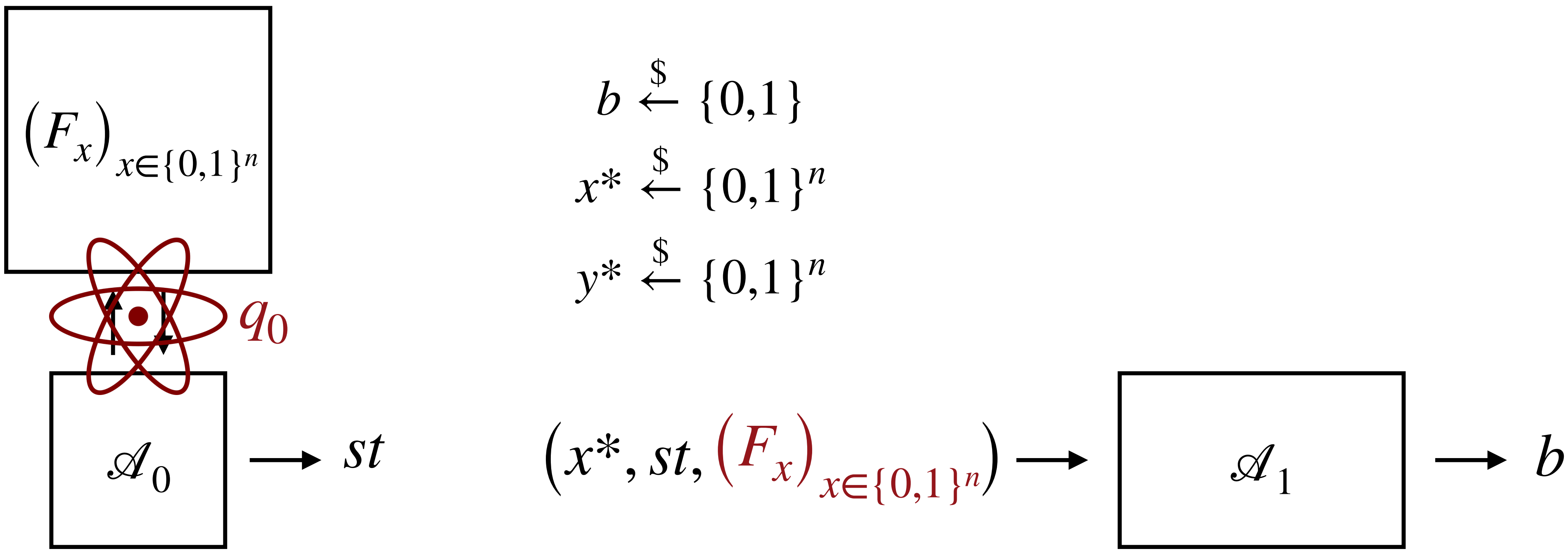
Proof ideas



- Intuition: q_0 is limiting quantity
- simplification: allow $q_1 = 2^n$

Oracle distinguishing → State discrimination!

Proof ideas



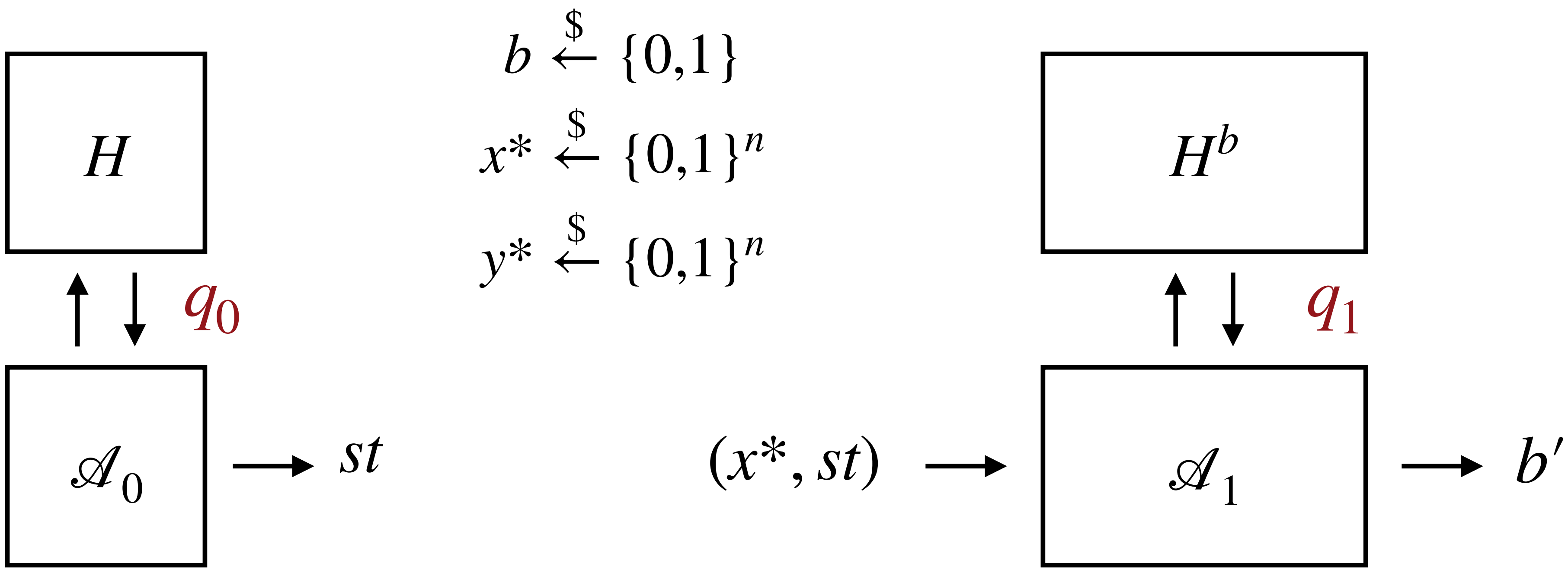
- Intuition: q_0 is limiting quantity
- simplification: allow $q_1 = 2^n$

Oracle distinguishing → State discrimination!

Suffices to bound a trace norm distance (for arbitrary $\mathscr{A}_0$).

A matching algorithm

Classical algorithm

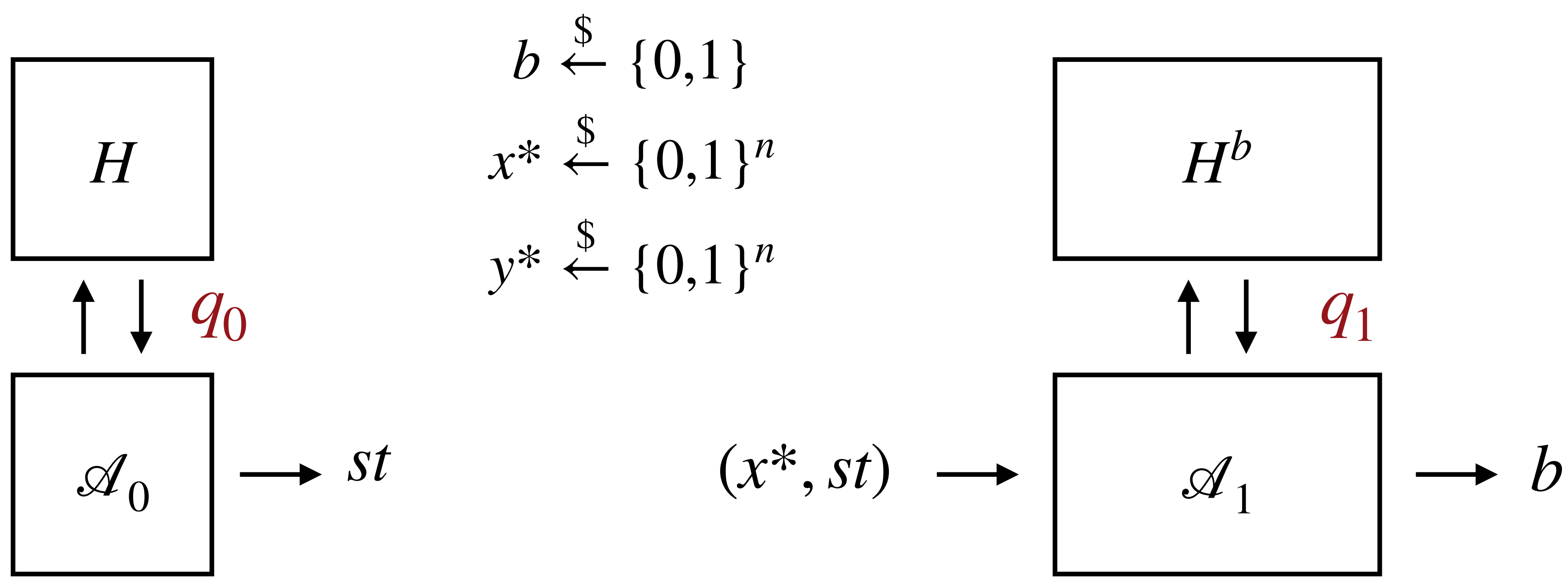


$\mathscr{A}$ wins if $b' = b$

Theorem:

$$\Pr[\mathscr{A} \text{ wins}] \leq \frac{1}{2}\left(1 + q_0 2^{-n}\right)$$

Classical algorithm



$\mathscr{A}$ wins if $b' = b$

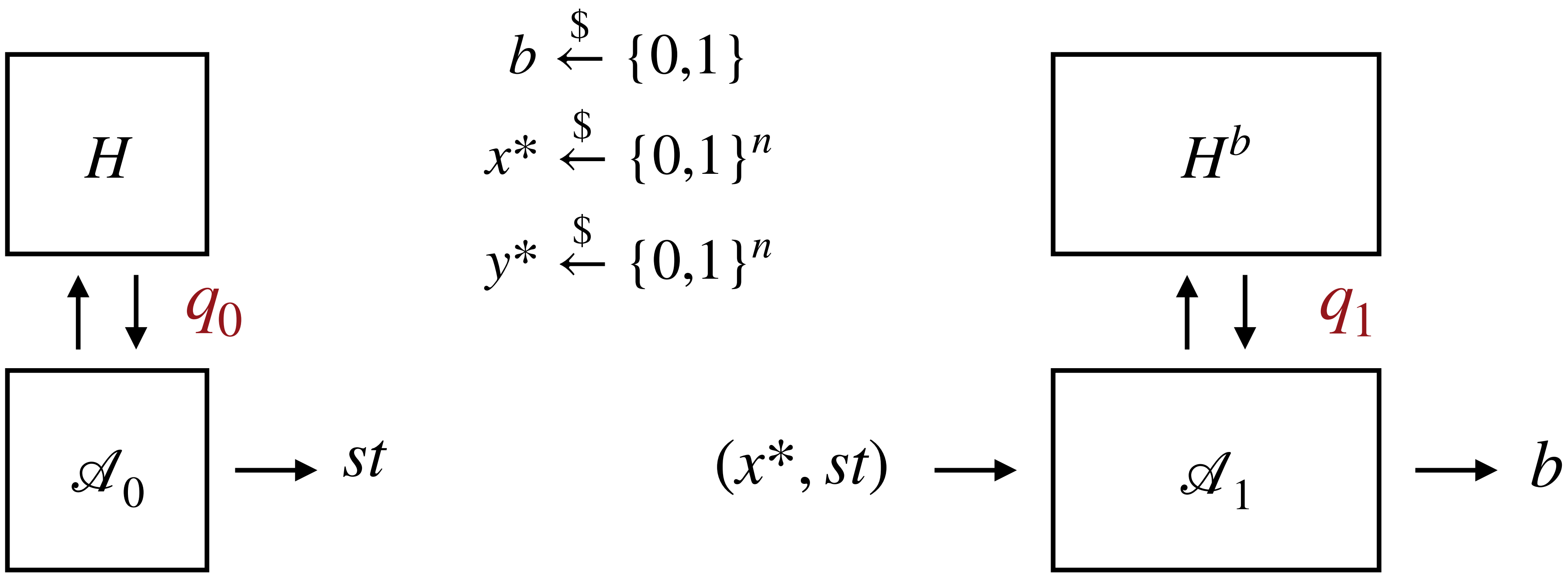
Theorem:

$$\Pr[\mathscr{A} \text{ wins}] \leq \frac{1}{2}\left(1 + q_0 2^{-n}\right)$$

Matching algorithms:

- Simple: query distinct inputs $x_1, \ldots, x_{q_0}$, store result, hope $x^* = x_i$ for some i

Classical algorithm



$\mathscr{A}$ wins if $b' = b$

Theorem:

$$\Pr[\mathscr{A} \text{ wins}] \leq \frac{1}{2}\left(1 + q_0 2^{-n}\right)$$

Matching algorithms:

- Simple: query distinct inputs $x_1, \ldots, x_{q_0}$, store result, hope $x^* = x_i$ for some i
- Constant space: $\mathscr{A}_0$ computes $H(x_0) \oplus H(x_1) \oplus \ldots \oplus H(x_{q_0-1})$, $\mathscr{A}_1$ checks

Quantum algorithm

Theorem: For classical $\mathscr{A}$,

$$\Pr[\mathscr{A} \text{ wins}] \leq \frac{1}{2}\left(1 + q_0 2^{-n}\right)$$

Theorem (Grilo, Hövelmanns, Hülsing, CM):
There exists a quantum algorithm that achieves

$$\Pr[\mathscr{A} \text{ wins}] = \frac{1}{2} + \Omega\left(\sqrt{q_0 2^{-n}}\right)$$

Quantum algorithm

Idea: use classical "checksum algorithm" for a superposition of sets of q_0 inputs

Quantum algorithm

Idea: use classical "checksum algorithm" for a superposition of sets of q_0 inputs

Classical algorithm:

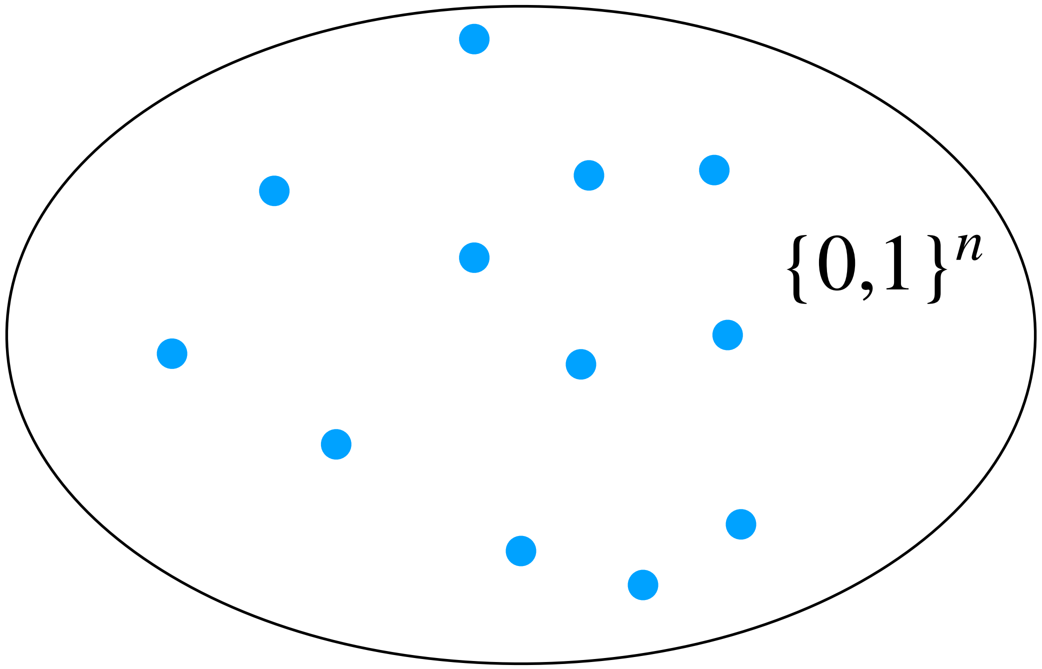
- $\mathscr{A}_0$ computes $z = H(x_0) \oplus H(x_1) \oplus \ldots \oplus H(x_{q_0-1})$
- $\mathscr{A}_1$ checks z

Quantum algorithm

Idea: use classical "checksum algorithm" for a superposition of sets of q_0 inputs

Classical algorithm:

- $\mathscr{A}_0$ computes $z = H(x_0) \oplus H(x_1) \oplus \ldots \oplus H(x_{q_0-1})$
- $\mathscr{A}_1$ checks z

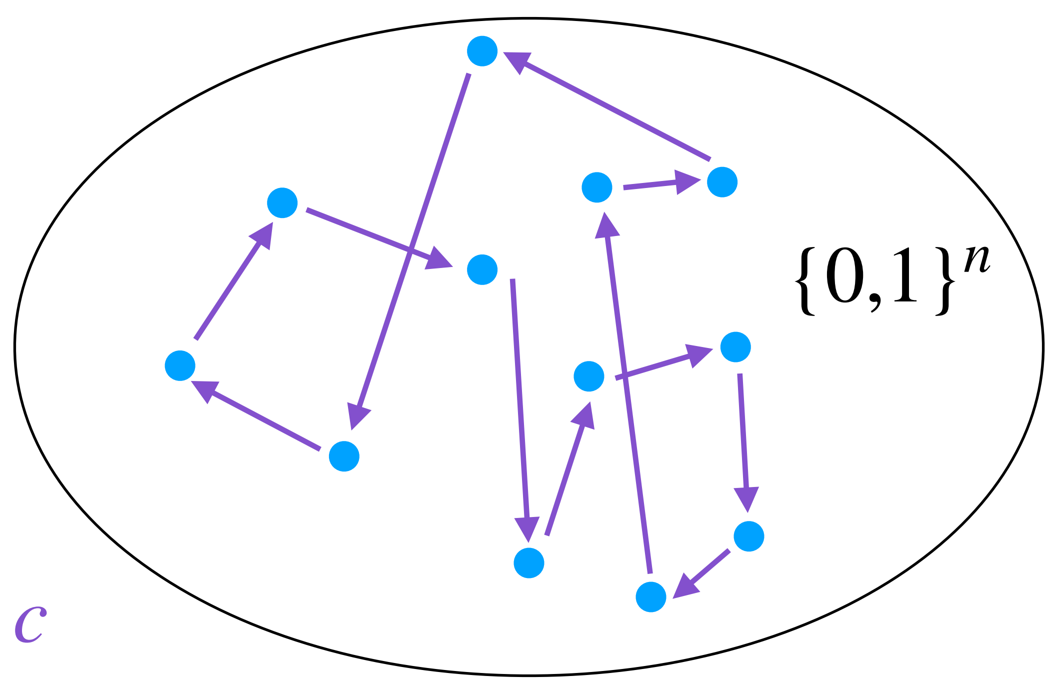


Quantum algorithm

Idea: use classical "checksum algorithm" for a superposition of sets of q_0 inputs

Classical algorithm:

- $\mathscr{A}_0$ computes $z = H(x_0) \oplus H(x_1) \oplus \ldots \oplus H(x_{q_0-1})$
- $\mathscr{A}_1$ checks z

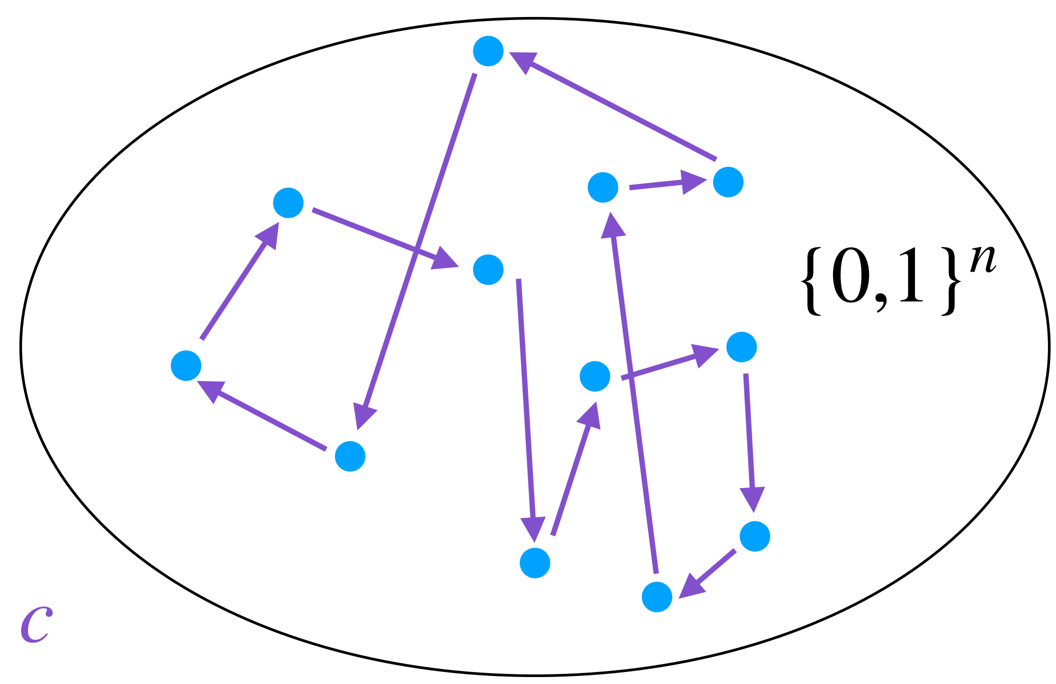


Quantum algorithm

Idea: use classical "checksum algorithm" for a superposition of sets of q_0 inputs

Classical algorithm:

- $\mathscr{A}_0$ computes $z = H(x_0) \oplus H(c(x_0)) \oplus \ldots \oplus H(c^{q_0-1}(x_0))$
- $\mathscr{A}_1$ checks z

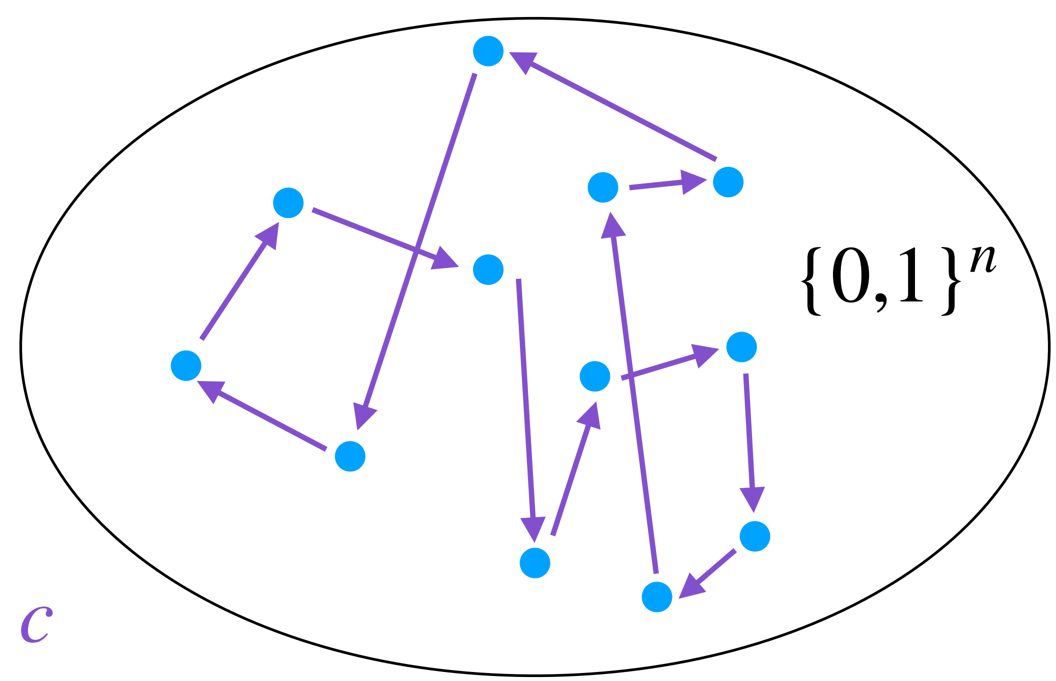


Quantum algorithm

Idea: use classical "checksum algorithm" for a superposition of sets of q_0 inputs

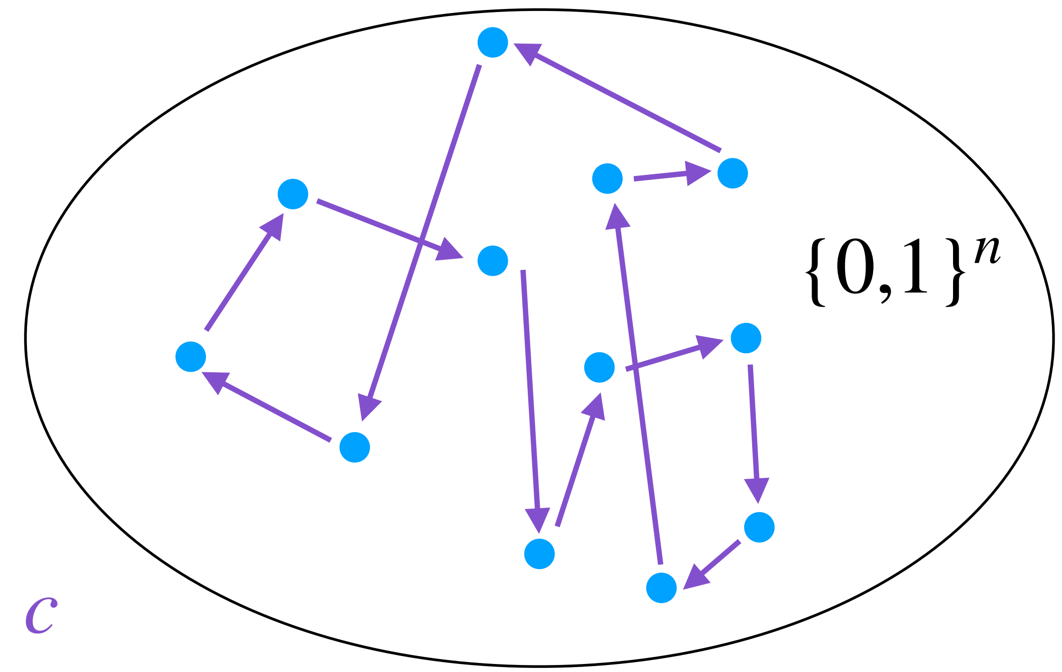
Classical algorithm:

- $\mathscr{A}_0$ computes $z = H(x_0) \oplus H(c(x_0)) \oplus \ldots \oplus H(c^{q_0-1}(x_0))$
- $\mathscr{A}_1$ tries to uncompute z, checks success



Quantum algorithm

Idea: use classical "checksum algorithm" for a superposition of sets of q_0 inputs

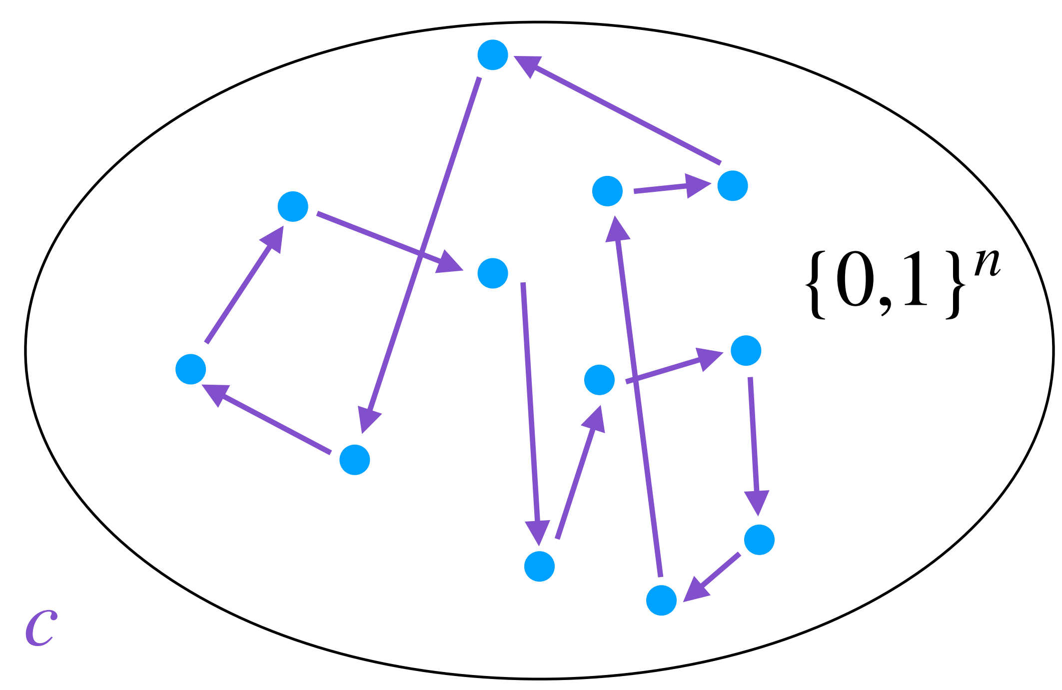


Quantum algorithm

Idea: use classical "checksum algorithm" for a superposition of sets of q_0 inputs

Quantum algorithm:

1. $\mathscr{A}_0$ prepares $|\phi_0\rangle = 2^{-\frac{n}{2}} \sum_{x \in \{0,1\}^n} |x\rangle_X |0\rangle_Y$

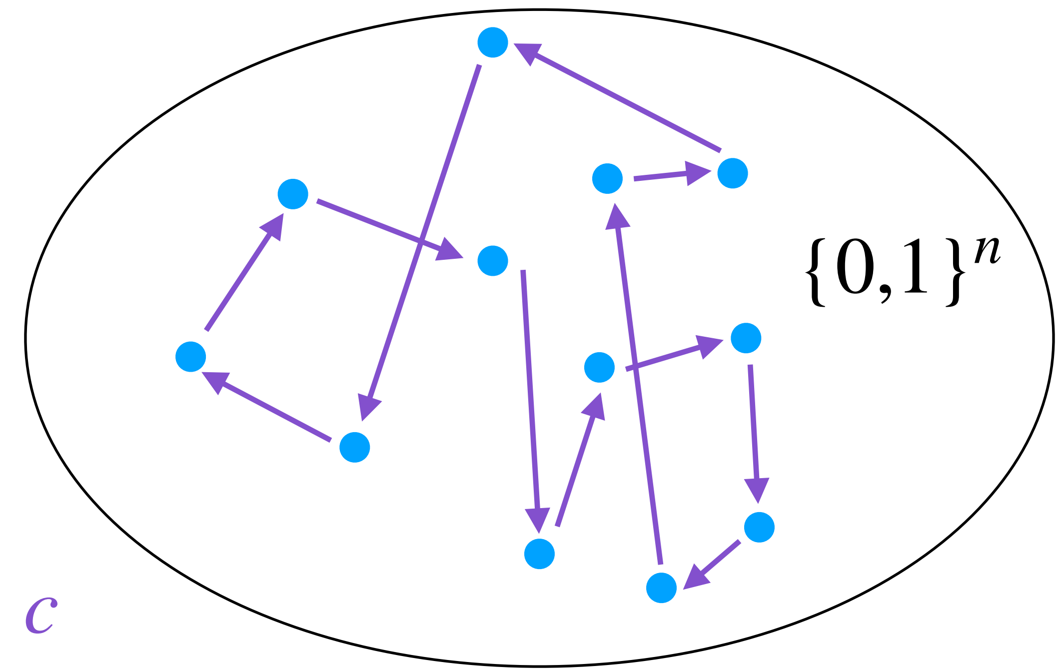


Quantum algorithm

Idea: use classical "checksum algorithm" for a superposition of sets of q_0 inputs

Quantum algorithm:

1. $\mathscr{A}_0$ prepares $|\phi_0\rangle = 2^{-\frac{n}{2}} \sum_{x \in \{0,1\}^n} |x\rangle_X |0\rangle_Y$
2. $\mathscr{A}_0$ repeats q_0 times:

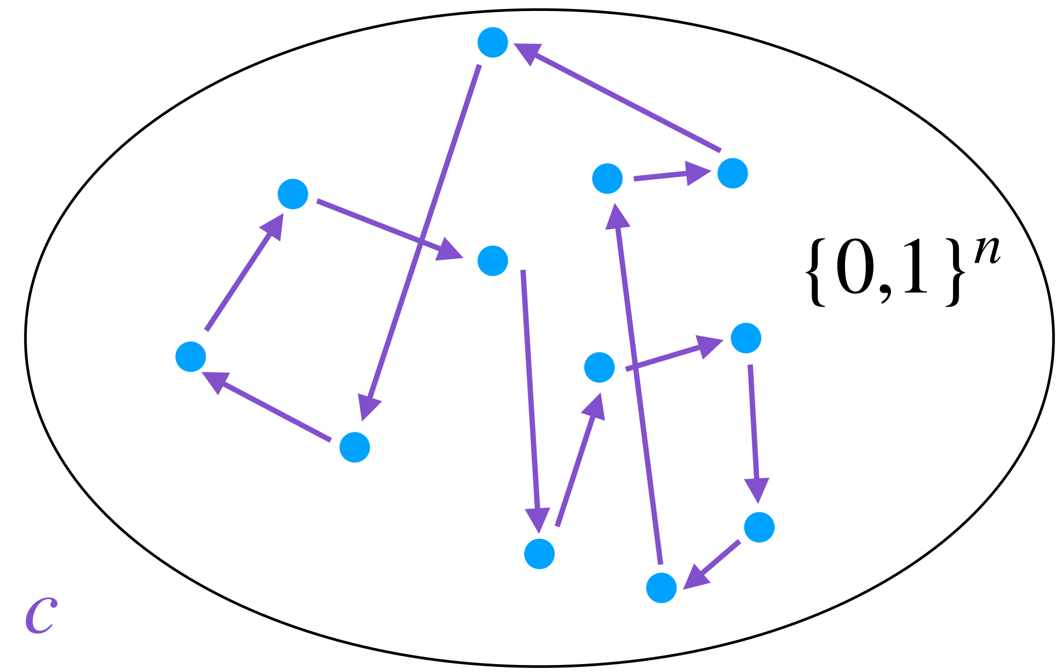


Quantum algorithm

Idea: use classical "checksum algorithm" for a superposition of sets of q_0 inputs

Quantum algorithm:

1. $\mathscr{A}_0$ prepares $|\phi_0\rangle = 2^{-\frac{n}{2}} \sum_{x \in \{0,1\}^n} |x\rangle_X |0\rangle_Y$
2. $\mathscr{A}_0$ repeats q_0 times:
   - query H

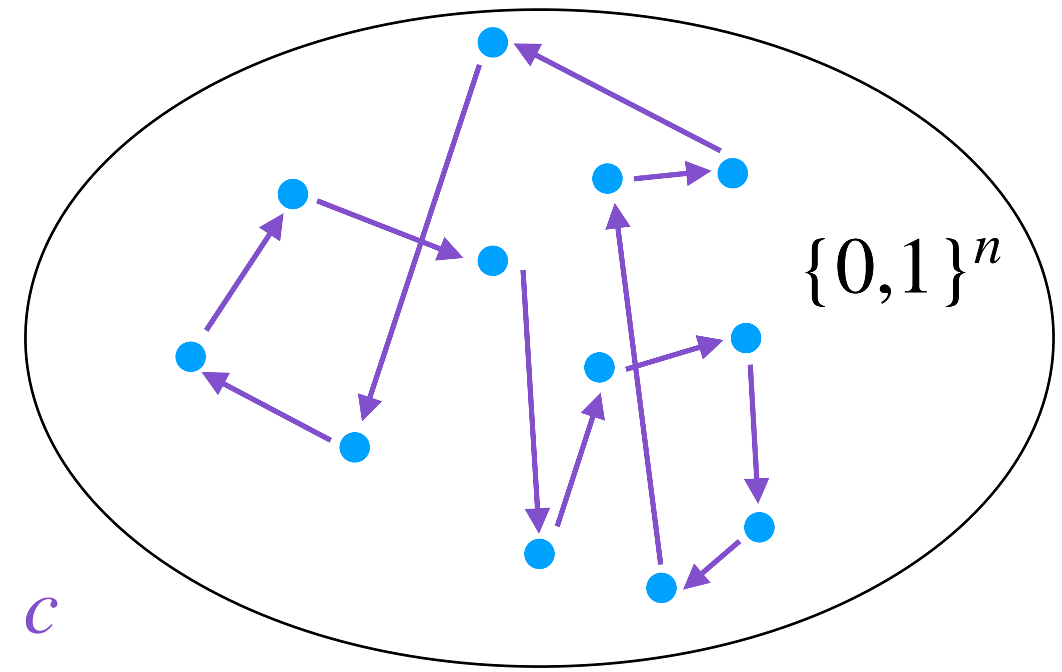


Quantum algorithm

Idea: use classical "checksum algorithm" for a superposition of sets of q_0 inputs

Quantum algorithm:

1. $\mathscr{A}_0$ prepares $|\phi_0\rangle = 2^{-\frac{n}{2}} \sum_{x \in \{0,1\}^n} |x\rangle_X |0\rangle_Y$
2. $\mathscr{A}_0$ repeats q_0 times:
   - query H
   - apply c to X

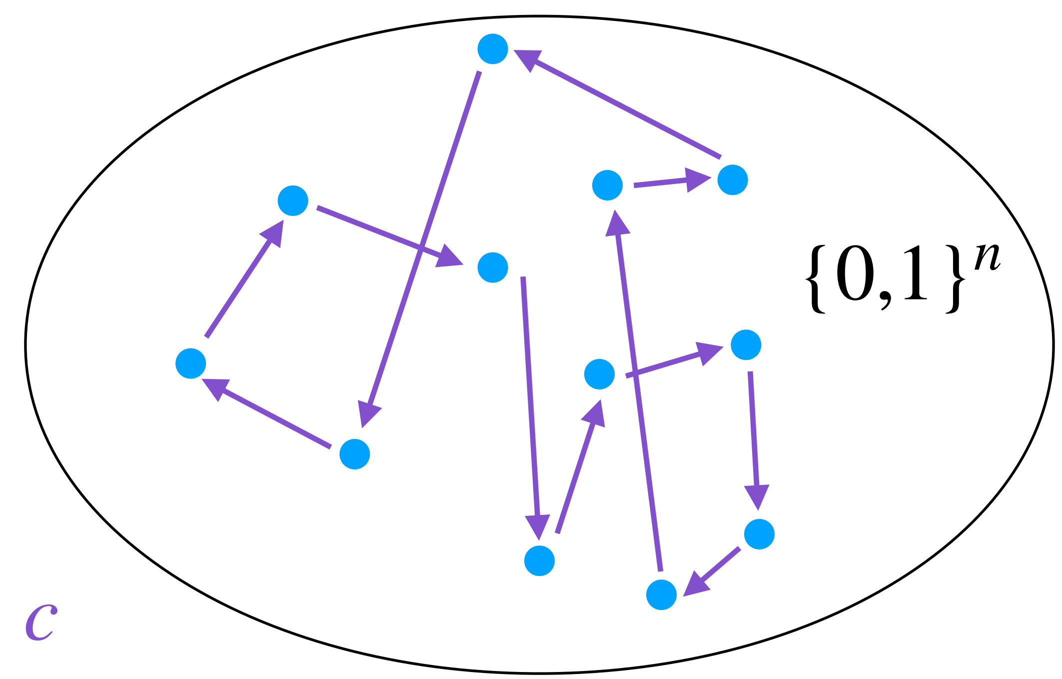


Quantum algorithm

Idea: use classical "checksum algorithm" for a superposition of sets of q_0 inputs

Quantum algorithm:

1. $\mathscr{A}_0$ prepares $|\phi_0\rangle = 2^{-\frac{n}{2}} \sum_{x\in\{0,1\}^n} |x\rangle_X |0\rangle_Y$
2. $\mathscr{A}_0$ repeats q_0 times:
   - query H
   - apply c to X
3. $\mathscr{A}_1$ tries to undo 2.

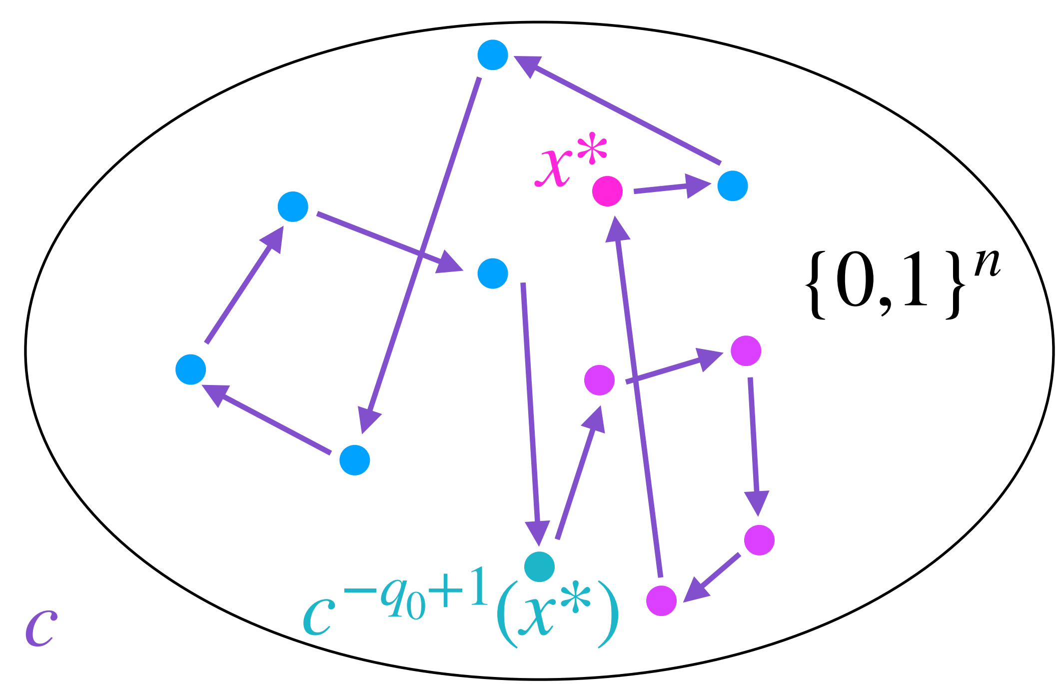


Quantum algorithm

Idea: use classical "checksum algorithm" for a superposition of sets of q_0 inputs

Quantum algorithm:

1. $\mathscr{A}_0$ prepares $|\phi_0\rangle = 2^{-\frac{n}{2}} \sum_{x \in \{0,1\}^n} |x\rangle_X |0\rangle_Y$
2. $\mathscr{A}_0$ repeats q_0 times:
    - query H
    - apply c to X
3. $\mathscr{A}_1$ tries to undo 2.

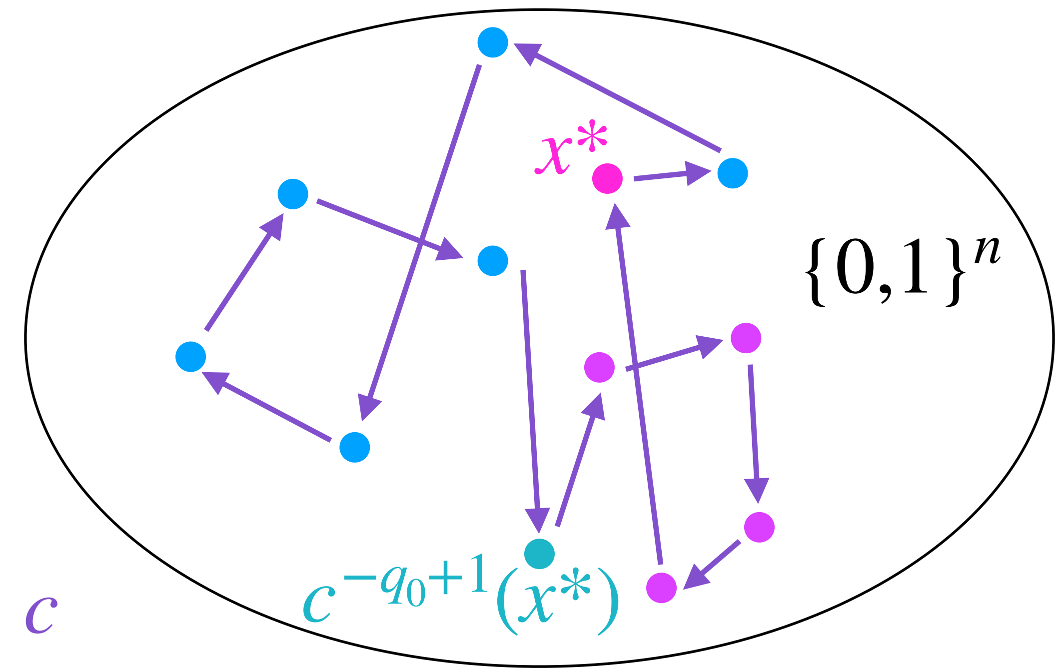


Quantum algorithm

Idea: use classical "checksum algorithm" for a superposition of sets of q_0 inputs

Quantum algorithm:

1. $\mathscr{A}_0$ prepares $|\phi_0\rangle = 2^{-\frac{n}{2}} \sum_{x \in \{0,1\}^n} |x\rangle_X |0\rangle_Y$
2. $\mathscr{A}_0$ repeats q_0 times:
    - query H
    - apply c to X
3. $\mathscr{A}_1$ tries to undo 2.



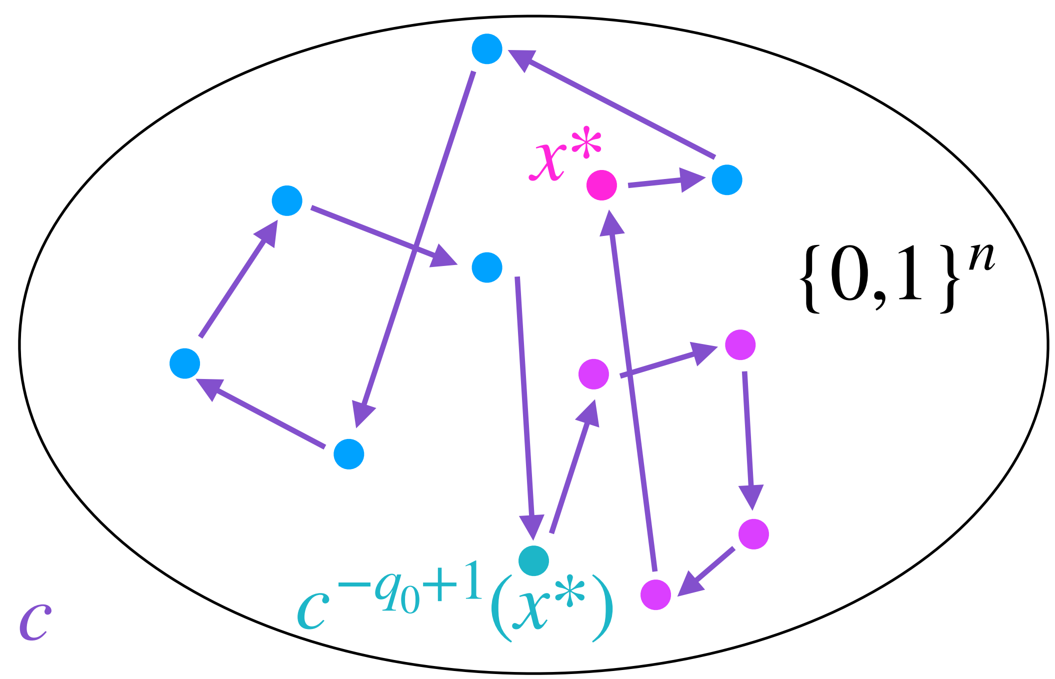
$S = \{ \bullet, \bullet, \bullet, \bullet, \bullet, \bullet \}$

Quantum algorithm

Idea: use classical "checksum algorithm" for a superposition of sets of q_0 inputs

Quantum algorithm:

1. $\mathscr{A}_0$ prepares $|\phi_0\rangle = 2^{-\frac{n}{2}} \sum_{x\in\{0,1\}^n} |x\rangle_X |0\rangle_Y$
2. $\mathscr{A}_0$ repeats q_0 times:
   - query H
   - apply c to X
3. $\mathscr{A}_1$ tries to undo 2.



Result: $|\phi_b\rangle$, with

$$|\phi_1 > = 2^{-\frac{n}{2}} \left(\sum_{x\in S} |x\rangle |H(x^*) \oplus y^*\rangle + \sum_{x\notin S} |x\rangle |0\rangle \right)$$

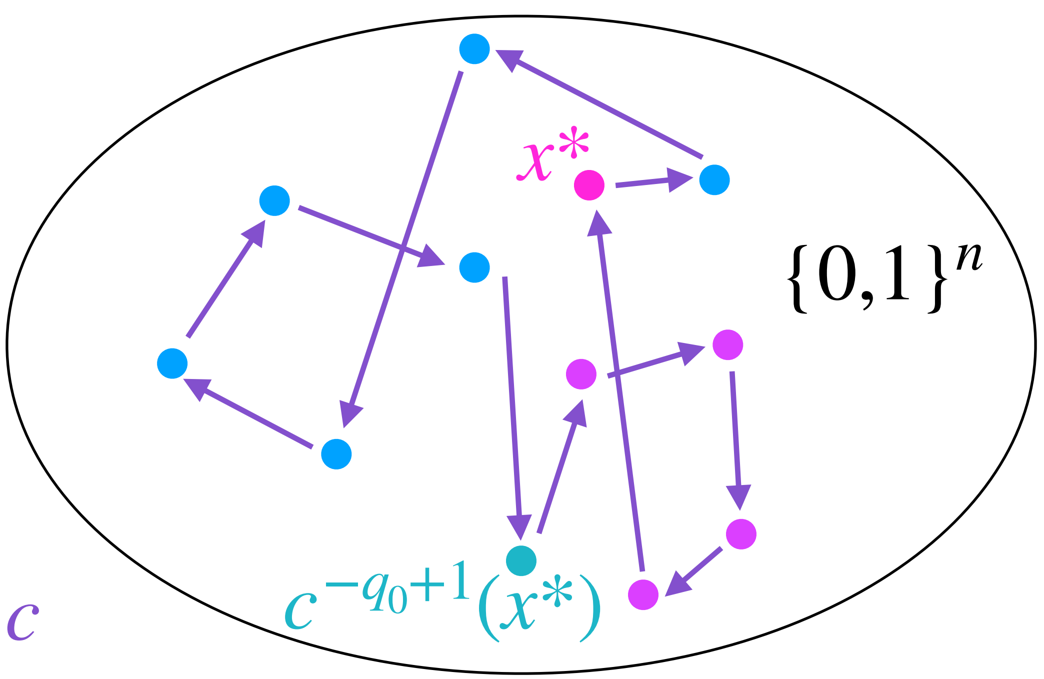
$S = \{\bullet, \bullet, \bullet, \bullet, \bullet, \bullet\}$

Quantum algorithm

Idea: use classical "checksum algorithm" for a superposition of sets of q_0 inputs

Quantum algorithm:

1. $\mathscr{A}_0$ prepares $|\phi_0\rangle = 2^{-\frac{n}{2}} \sum_{x \in \{0,1\}^n} |x\rangle_X |0\rangle_Y$
2. $\mathscr{A}_0$ repeats q_0 times:
   - query H
   - apply c to X
3. $\mathscr{A}_1$ tries to undo 2.



c

Result: $|\phi_b\rangle$, with

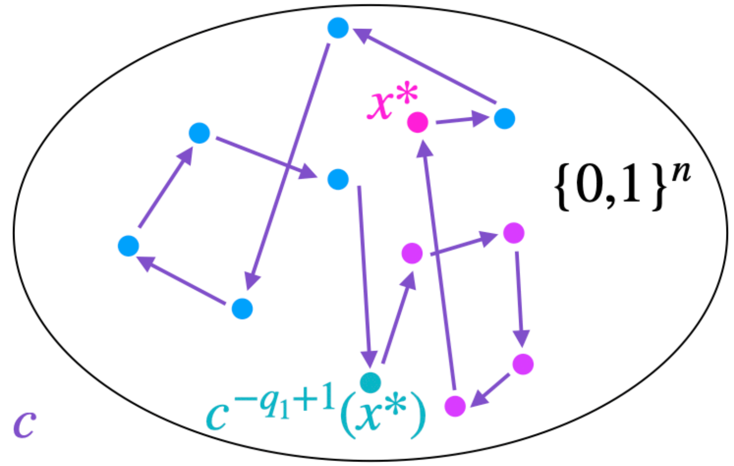
$$|\phi_1> = 2^{-\frac{n}{2}} \left(\sum_{x \in S} |x\rangle |H(x^*) \oplus y^*\rangle + \sum_{x \notin S} |x\rangle |0\rangle \right)$$

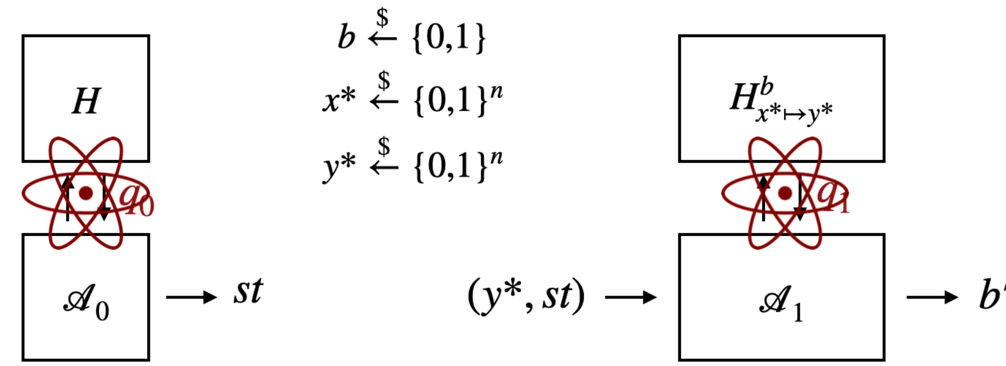
$$\left\| |\phi_0\rangle - |\phi_1\rangle \right\| = \sqrt{2q_0 2^{-n}}$$

$S = \{ \bullet, \bullet, \bullet, \bullet, \bullet, \bullet \}$

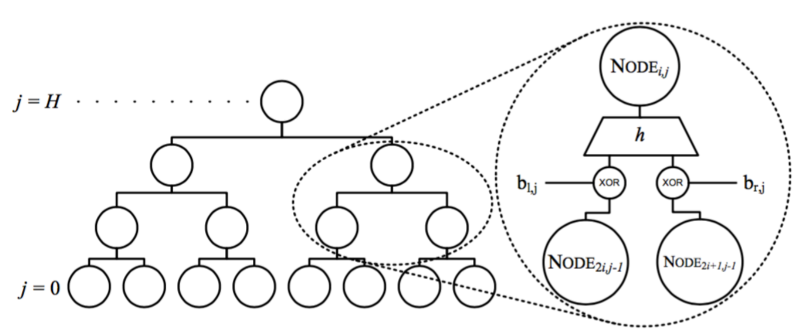
Summary

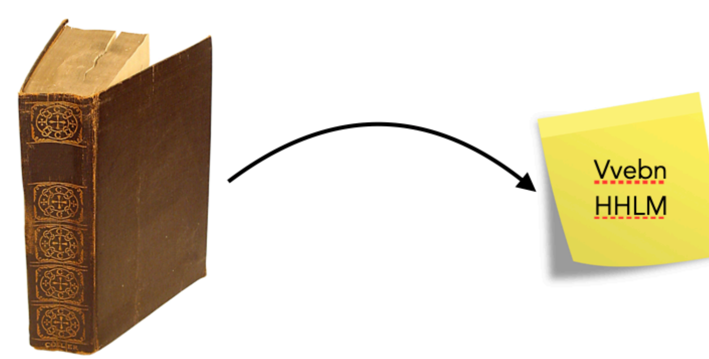
- Tight characterization of "adaptive reprogramming" oracle distinguishing task in the quantum setting
- Informs NIST competition for post-quantum crypto schemes
- Proof based on simplest version of Zhandry's superposition oracle
- Efficient algorithm matching the bound.


x^*
$\{0,1\}^n$
c
$c^{-q_1+1}(x^*)$


H
q_0
$\mathscr{A}_0$
$\to st$
$b \xleftarrow{\$} \{0,1\}$
$x^* \xleftarrow{\$} \{0,1\}^n$
$y^* \xleftarrow{\$} \{0,1\}^n$
$H^b_{x^* \mapsto y^*}$
q_1
$(y^*, st) \to$
$\mathscr{A}_1$
$\to b'$

Thanks!


$j = H$
$j = 0$
NODE$_{i,j}$
h
$b_{l,j}$
XOR
XOR
$b_{r,j}$
NODE$_{2i,j-1}$
NODE$_{2i+1,j-1}$


Vvebn
HHLM