Quantum Interactive Proof Systems

John Watrous
Department of Computer Science
University of Calgary

Classes of Problems

Computational problems can be classified in many different ways. Examples of classes:

- problems solvable in polynomial time by some deterministic Turing machine
- problems solvable by boolean circuits having a polynomial number of gates
- problems solvable in polynomial space by some deterministic Turing machine
- problems that can be reduced to integer factoring in polynomial time

Commonly Studied Classes

P	class of problems solvable in polynomial time on some deterministic Turing machine
NP	class of problems solvable in polynomial time on some nondeterministic Turing machine
	Informally: problems with efficiently checkable solutions
PSPACE	class of problems solvable in polynomial space on some deterministic Turing machine

Commonly Studied Classes

BPP — class of problems solvable in polynomial time on some probabilistic Turing machine (with "reasonable" error bounds)

L — class of problems solvable by some deterministic Turing machine that uses only logarithmic work space

SL, RL, NL, PL, LOGCFL, NC, SC, ZPP, R, P/poly, MA, SZK, AM, PP, PH, EXP, NEXP, EXPSPACE, . . .

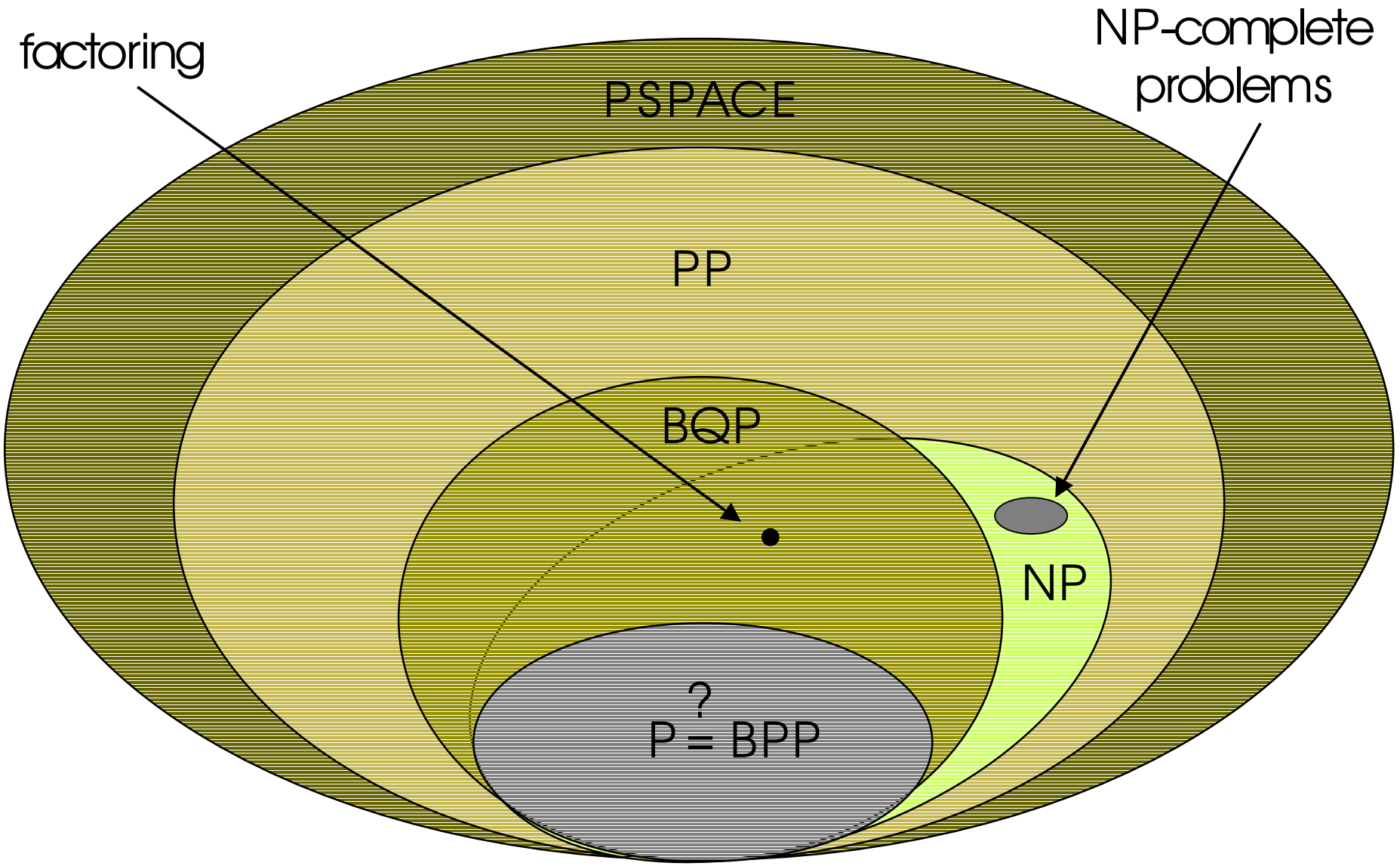
Quantum Polynomial Time

BQP

class of problems solvable in polynomial time on some quantum Turing machine (with "reasonable" error bounds)

equivalently: problems solvable by quantum circuits having a polynomial number of gates (again, with "reasonable" error bounds) plus technical restrictions on the circuits

Diagram of Complexity Classes



Interactive Proof Systems

- Introduced in 1985 by Babai and Goldwasser, Micali, and Rackoff.

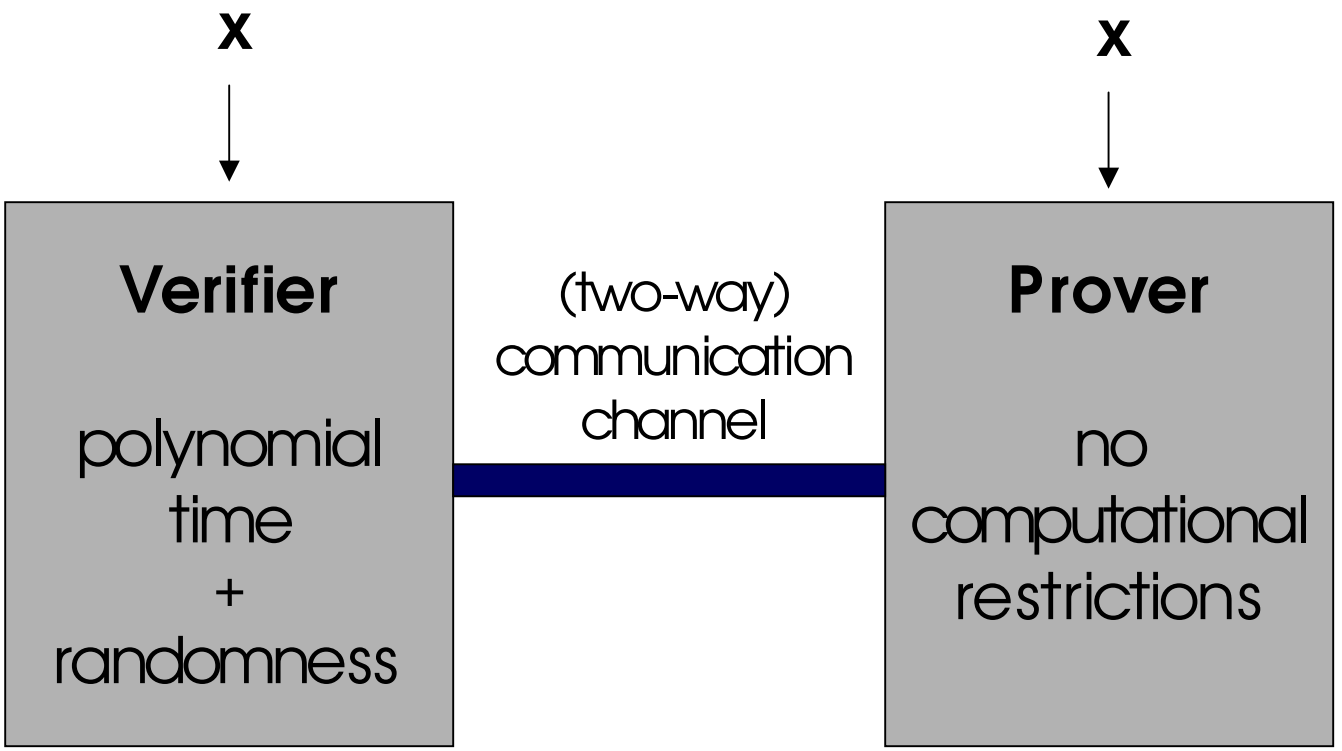
- Idea: two parties, called the **prover** and the **verifier**, have a conversation based on some common input string **x**.

  The prover has unlimited computation power.

  The verifier must run in polynomial time (and can flip coins).

  The prover wants the verifier to believe that the input **x** satisfies some fixed property. . . the verifier wants to verify the validity of this claim.

Interactive Proof Systems



Output: { accept — if the verifier believes the prover that **x** has property **A**

reject — otherwise

Which properties have interactive proof systems?

A property (or language) **A** has an interactive proof system if:

There exists a verifier V such that the following two conditions are satisfied.

1. (Completeness condition)

   If $x \in A$ then there exists a prover P that can convince V to accept x (with high probability).

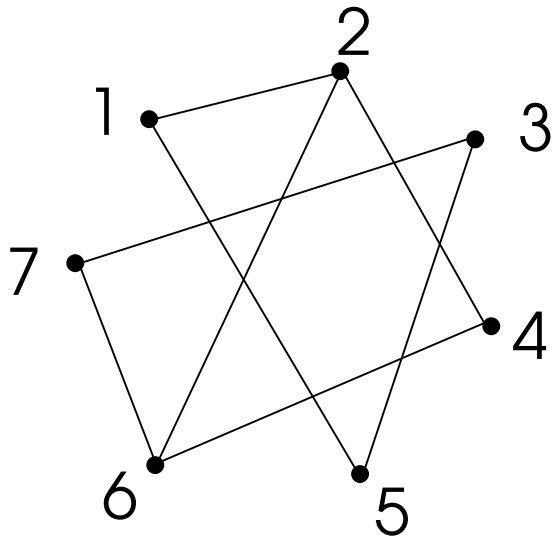
2. (Soundness condition)

   If $x \notin A$ then no prover can convince V to accept x (except with very small probability).

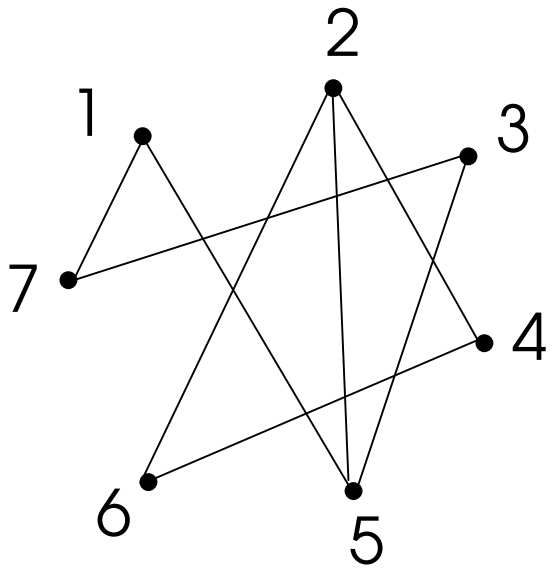
Example: Graph Non-Isomorphism

Suppose the input consists of two graphs: G_1 and G_2 .

The prover wants to convince the verifier that $G_1 \ncong G_2$.



G_1



G_2

Example: Graph Non-Isomorphism

The protocol:

1. The verifier randomly chooses $i \in \{1,2\}$, randomly permutes G_i, and sends the resulting graph H to the prover.
2. The prover is challenged to identify whether H is isomorphic to G_1 or G_2 (i.e, to determine i).

   The prover sends his guess to the verifier.
3. The verifier **accepts** if the prover correctly guesses i, and **rejects** otherwise.

Which properties have interactive proof systems?

Let IP denote the class of properties that have interactive proof systems.

(Lund, Fortnow, Karloff, and Nisan, 1990) + (Shamir, 1990):

$$IP = PSPACE$$

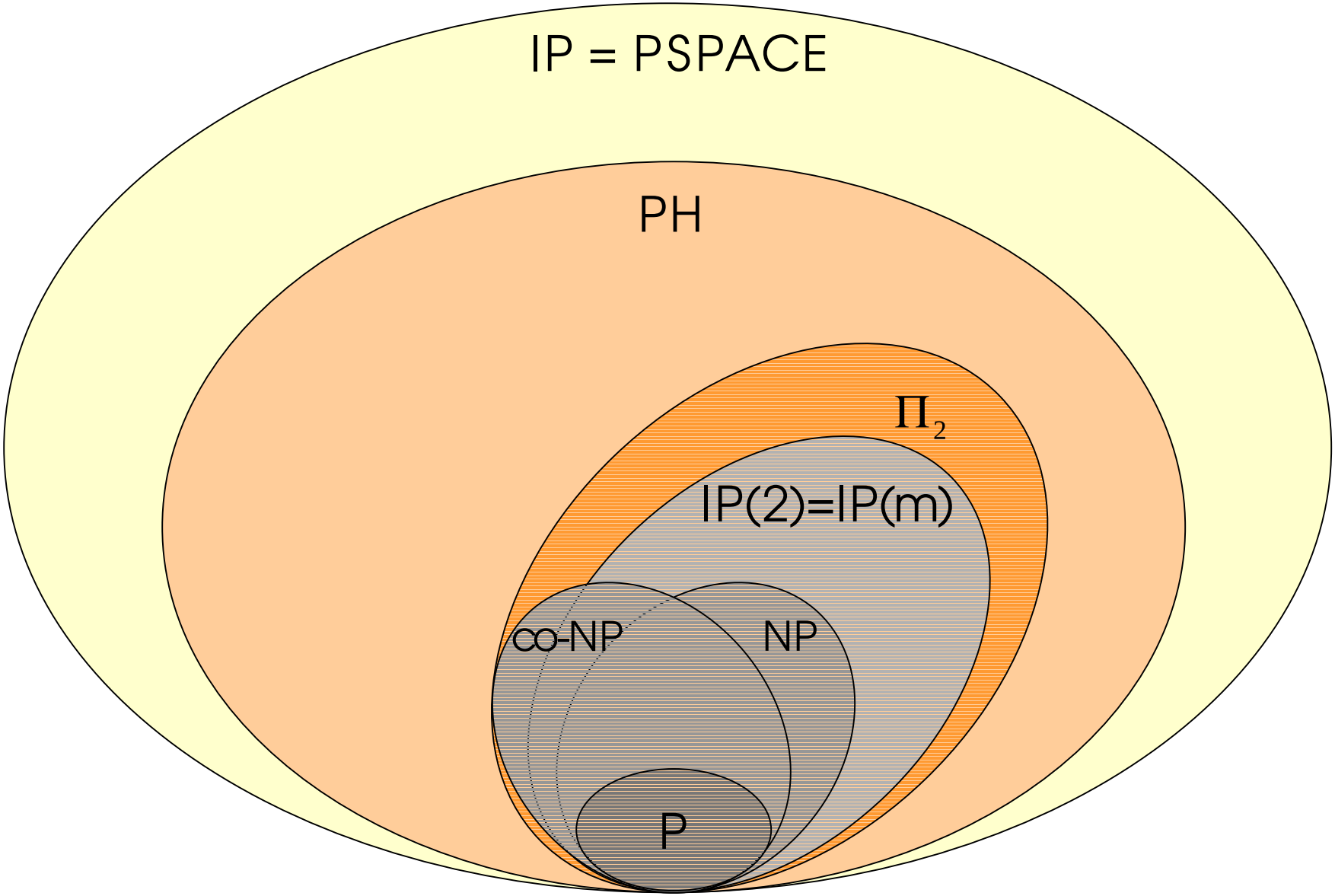
Let IP(m) denote the class of sets having interactive proof systems where the total number of messages sent is at most *m*.

(Babai, 1985) + (Goldwasser and Sipser, 1989):

$$IP(m) = IP(2) \subseteq \Pi_2$$

for any constant *m*.

Diagram of complexity classes



Why study interactive proof systems?

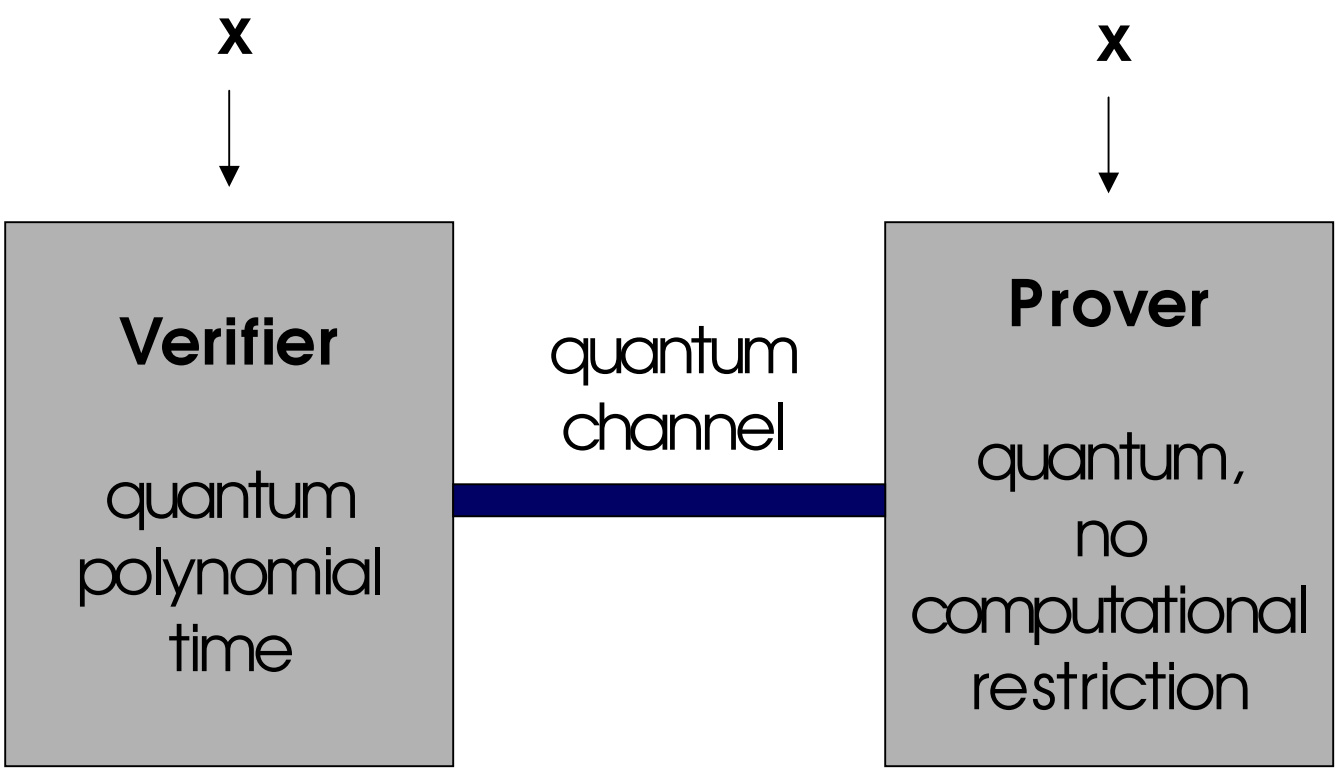
- Theoretical foundation for studying various cryptographic situations.

- Useful way to help classify problems.

- The study of interactive proof systems has had important (and unexpected) applications in complexity theory.

  Primary example: proving hardness of certain approximation problems.

Quantum Interactive Proof Systems

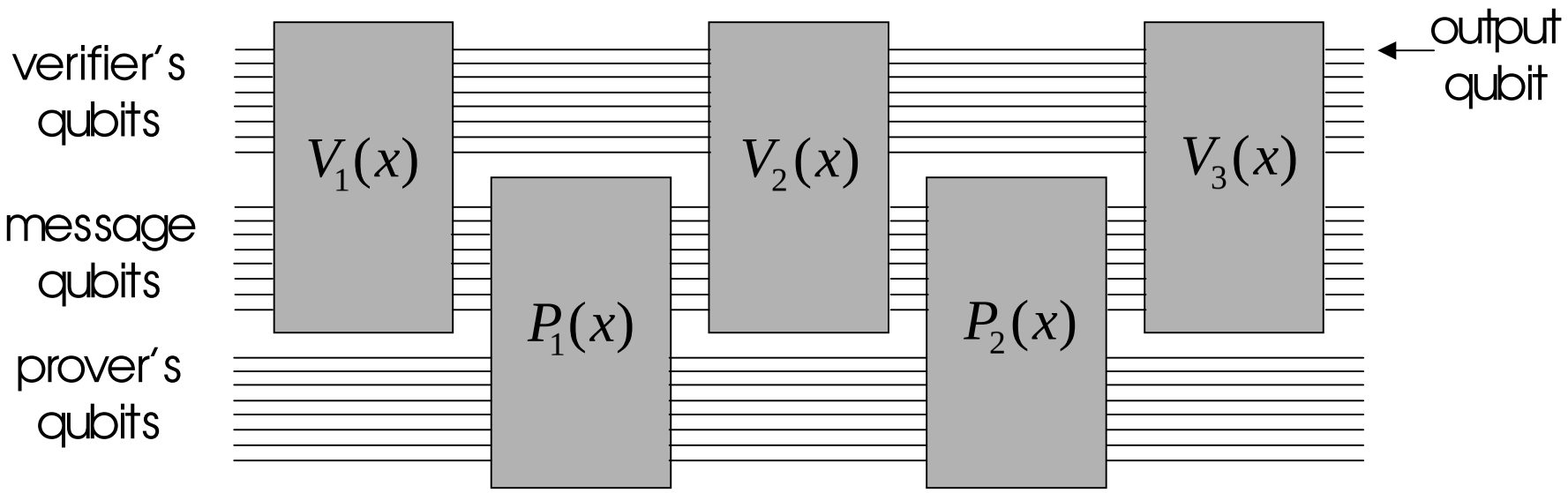


Output: { accept — if the verifier believes the prover that **x** has property **A**

reject — otherwise

Formalizing the model

We use the quantum circuit model. Example of a circuit for a 4-message quantum interactive proof system:



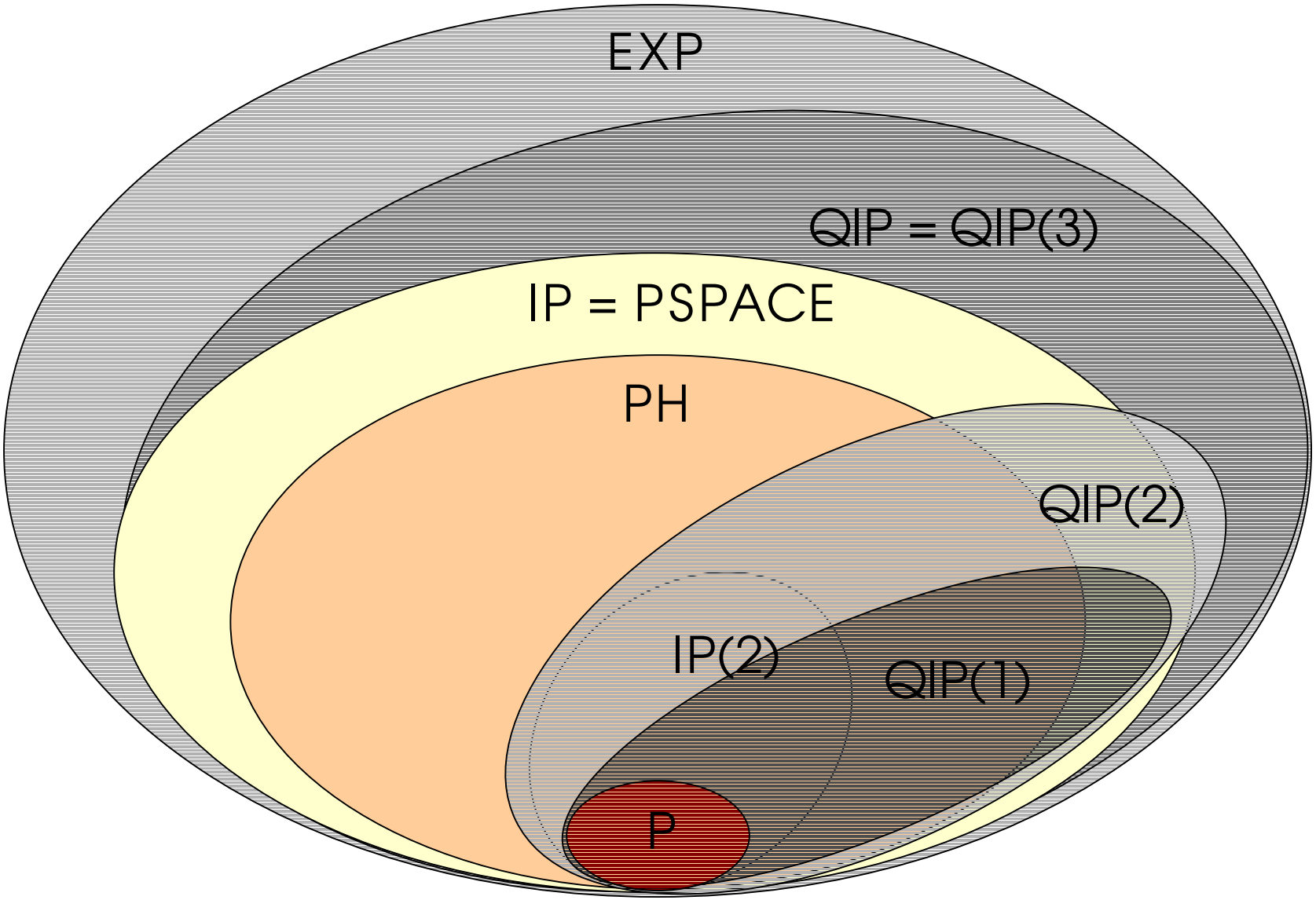
Quantum vs. Classical Interactive Proofs

- $IP \subseteq QIP$ and $IP(m) \subseteq QIP(m)$ for any m.
- $QIP = QIP(3)$

$$\Rightarrow \quad PSPACE \subseteq QIP(3)$$

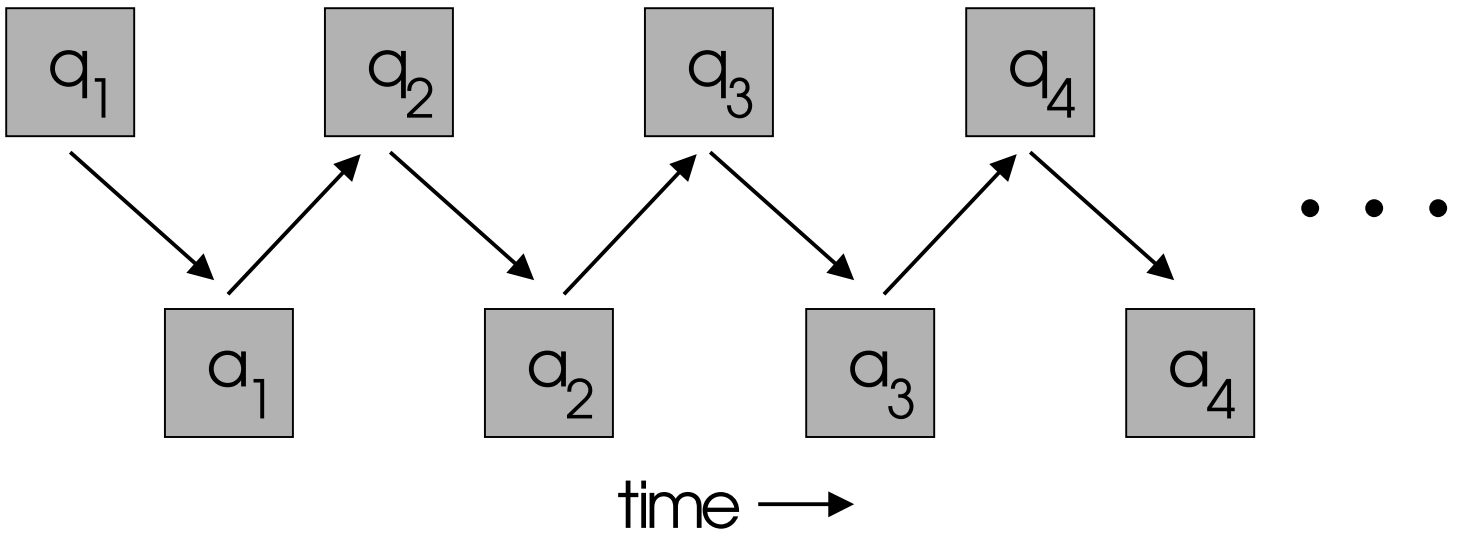
- $QIP \subseteq EXP$ (deterministic exponential time)
- $BQP \subseteq QIP(1) \subseteq PP \subseteq PSPACE$
- Can make completeness error 0 (cost: 2 extra messages).
- Parallel repetition works (for completeness error 0 case).

Diagram of complexity classes



Parallelizing classical proof systems

Suppose we have a classical interactive proof system consisting of many rounds:



It could be very important to the protocol that each question is answered before the next is asked. . .

Parallelizing classical proof systems

Naïve way to try and parallelize:

> Send all questions in one message, let the prover respond, and check the answers.

It doesn't work. . . if the prover can look at **all** the questions when giving his answers, he can (likely) cheat.

However, using quantum information, this method can be made to work. . .

Parallelizing classical proof systems

Idea:

The verifier asks all of his questions at once, **in superposition**.

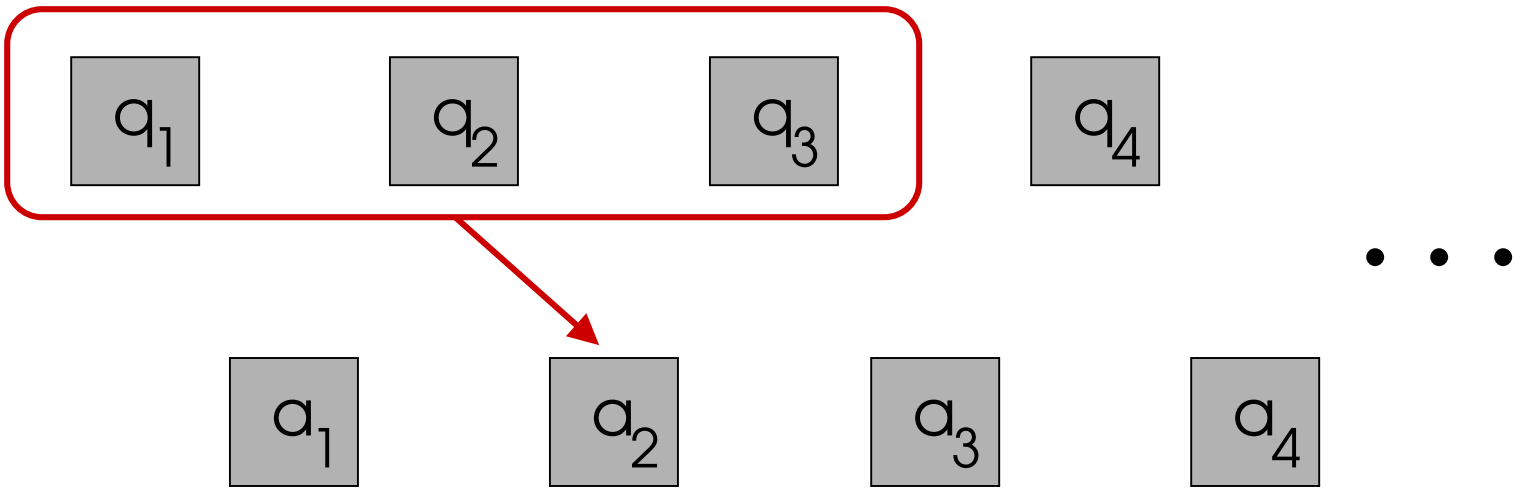
The prover answers.

The verifier checks that the answers were acceptable.

Now, the verifier needs to check that there are no "illegal correlations" among the questions and answers. (This will require 2 additional messages.)

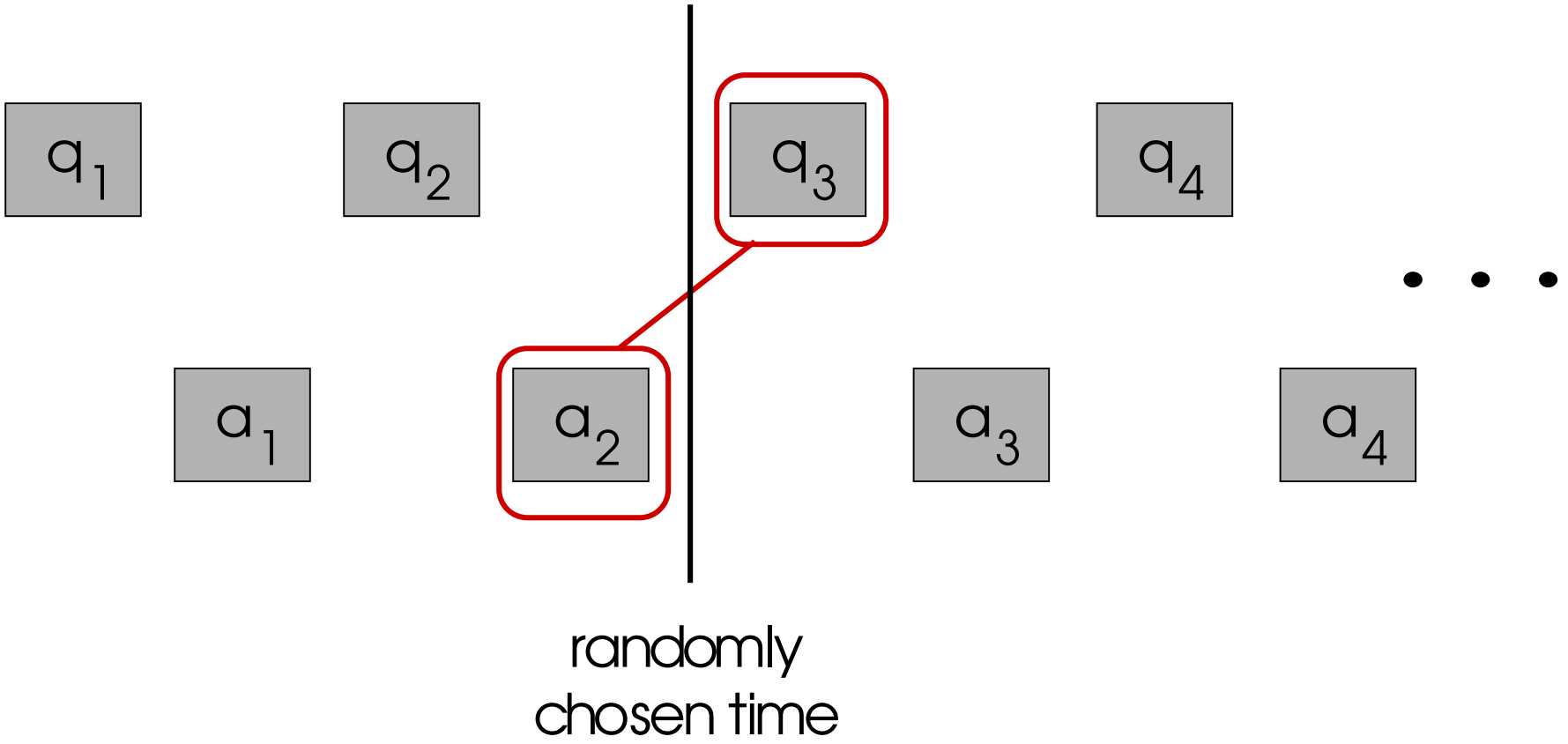
Parallelizing classical proof systems

Example of an "illegal correlation": answer 2 depends on question 3.

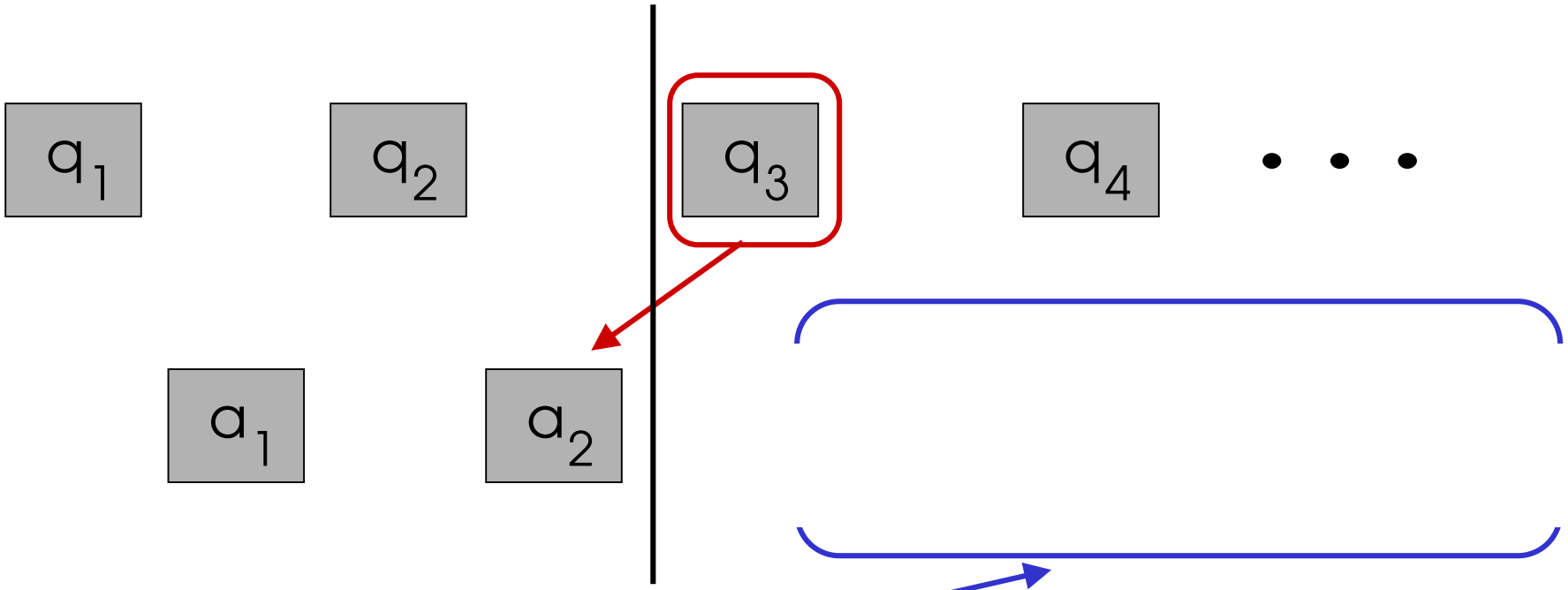


Parallelizing classical proof systems

The verifier will randomly choose a time in the protocol, and look for "illegal correlations" that cross this line.



Parallelizing classical proof systems



These answers are given back to the prover (so he has a chance to disentangle them from the questions occurring after the chosen time).

At this point, the register containing question 3 should not be entangled with anything.

The verifier can easily verify this by measuring this register appropriately.

Parallelizing classical proof systems

Notes:

- The number of messages can be reduced to 3.
- Completeness error is 0, soundness error can be made exponentially small (still 3 messages).
- Parallelization of quantum interactive proof systems uses somewhat different techniques. . .

Open Questions

1. There are many variants of (classical) interactive proof systems:

   - interactive proofs with stronger restrictions on the verifier (or on the prover).
   - multi-prover interactive proof systems
   - multiple competing provers
   - probabilistically checkable proofs
   - zero-knowledge

   General problem:

   How do quantum versions of these proof systems compare to the classical case?

   (Quantum versions of some of these have been studied.)

Open Questions

2. What else can be said about relations between quantum interactive proof system classes and other complexity classes?

   What can be said about $QIP(2)$?

3. Applications?