Quantum Complexity Theory II:
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 a 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 a deterministic Turing machine

Commonly Studied Classes

BPP — class of problems solvable in polynomial time on a 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, . . .

The list goes on...

…, #P, #L, AC, SPP, SAC, WPP, NE, AWPP, FewP, CZK, PCP(r(n),q(n)), D#P, NPO, GapL, GapP, LIN, ModP, NLIN, k-BPB, $P^{NP[log]}$, P^{PP}, $Pr_H SPACE(s)$, S_2P, $C_=P$, APX, DET, DisNP, EE, ELEMENTARY, mL, NISZK, OptP, UP, UL, W[SAT], symP, SO-E, SF_k, NEE, mNL, MaxSNP, MA-EXP, FOLL, BP_HL, AH , $+SAC_1$, …

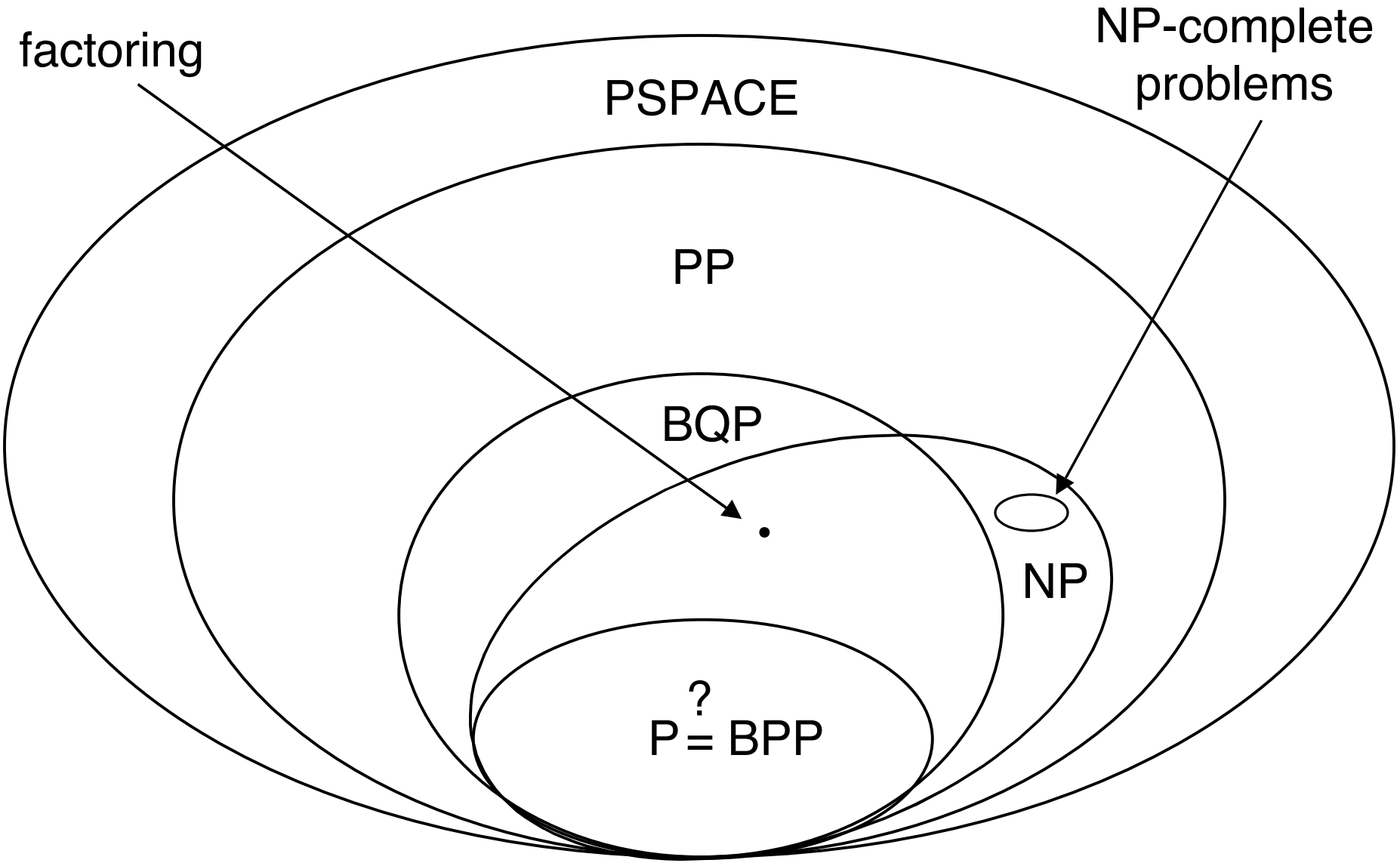
Quantum Polynomial Time

BQP

class of problems solvable in polynomial time on a 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

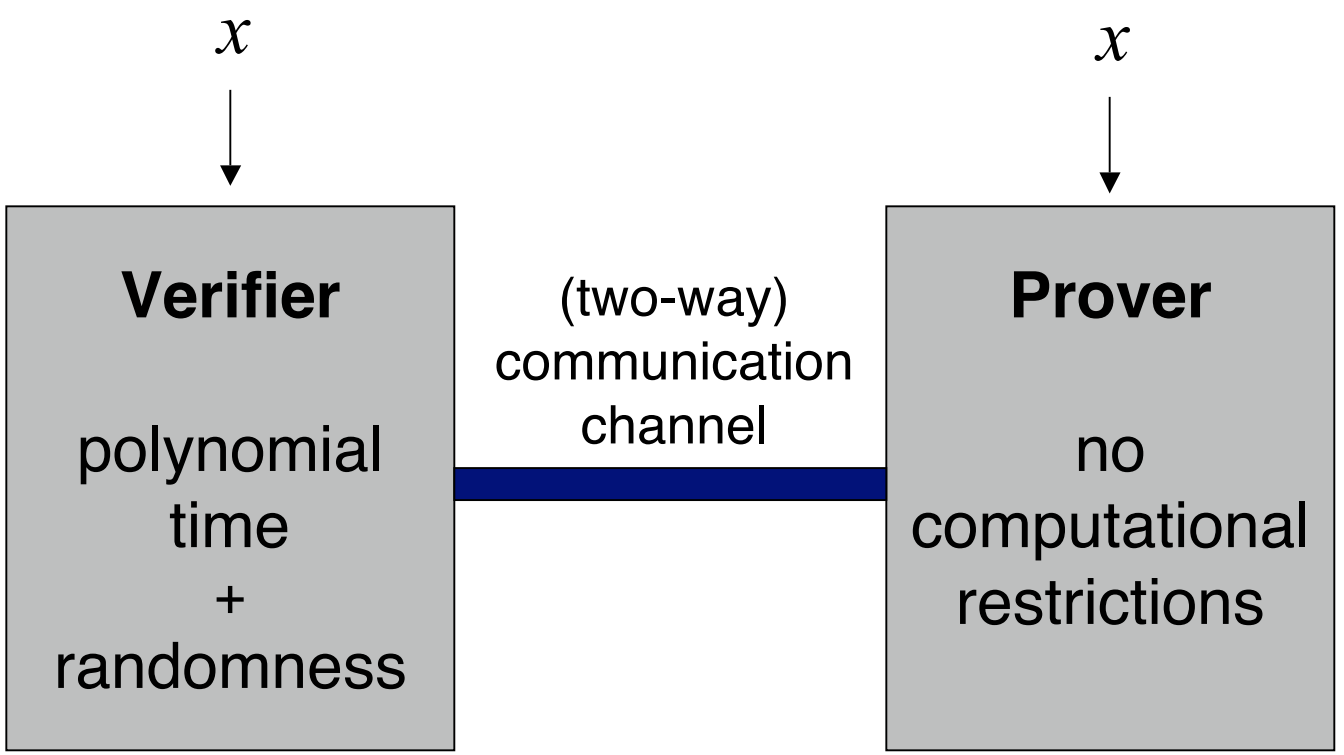
- Introduced in 1985 by Babai and Goldwasser, Micali, and Rackoff.
- Idea: two parties, called the **prover** and the **verifier**, have a conversation based on some common input string x.

  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 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

Properties with 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).

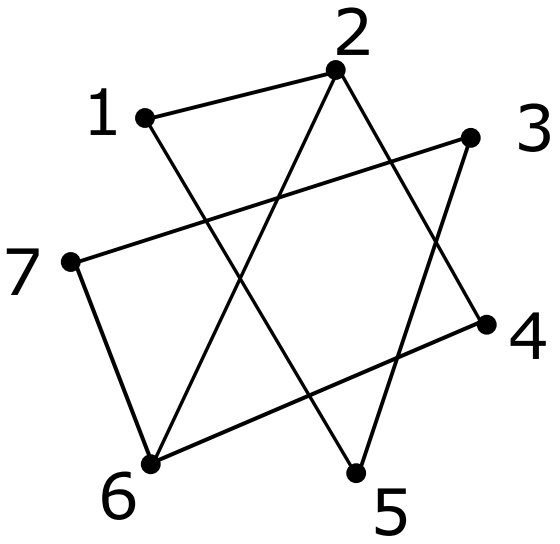
2. (Soundness condition)

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

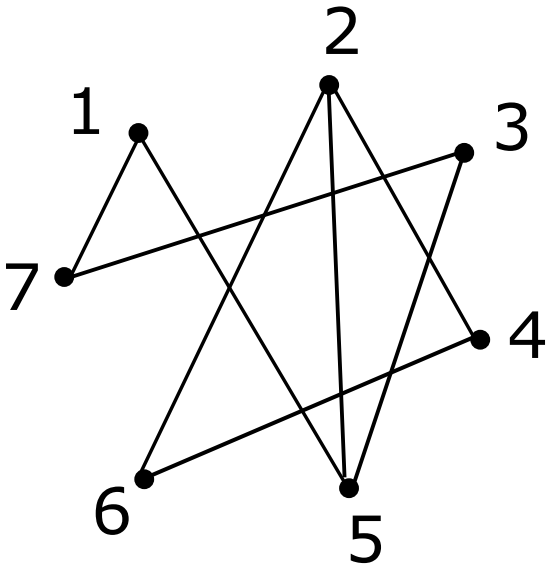
Example: Graph Non-Isomorphism

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

The prover wants to convince the verifier that
$G_1 \not\cong G_2$



G_1



G_2

Example: Graph Non-Isomorphism

The protocol:

1. The verifier randomly chooses one of the two graphs, randomly permutes it, and sends it to the prover.

2. The prover is challenged to identify whether the graph send by the verifier is isomorphic to the first or second input graph.

   The prover sends his guess to the verifier.

3. The verifier **accepts** if the prover correctly guesses the correct graph 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]:

$$IP = PSPACE$$

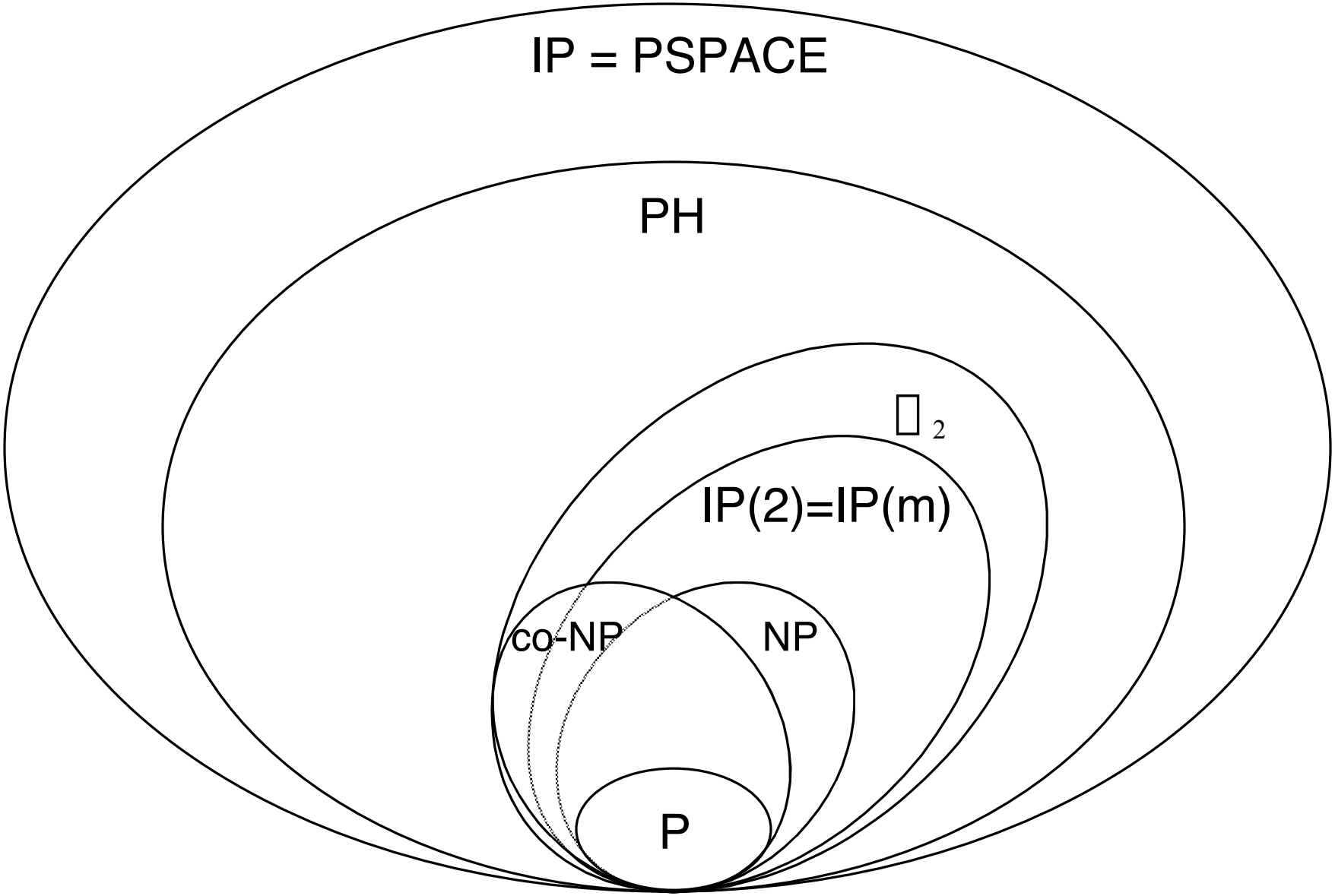
Let $IP(m)$ denote the class of sets having interactive proof systems where the total number of messages sent is at most m.

[Babai, 1985] + [Goldwasser and Sipser, 1989]:

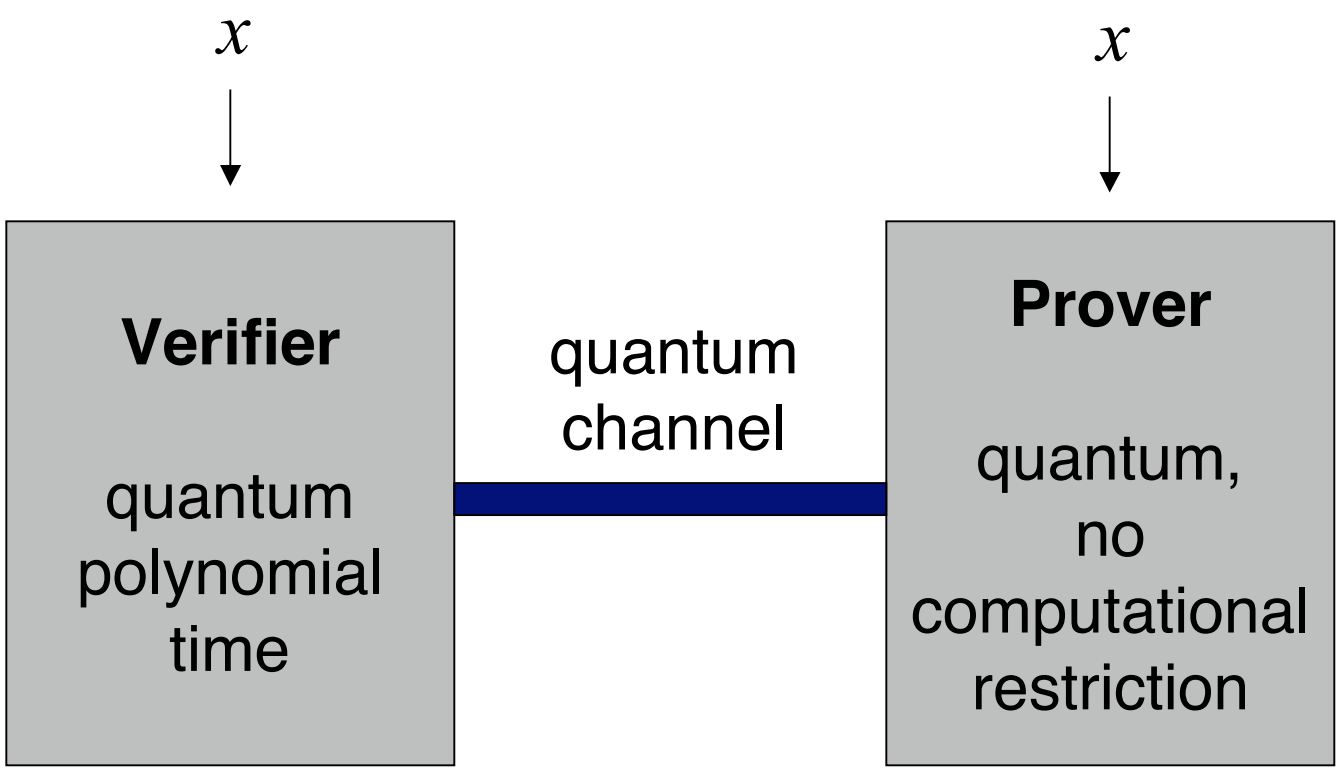
$$IP(m) = IP(2)$$

for any constant m.

Diagram of complexity classes



Quantum Interactive Proof Systems

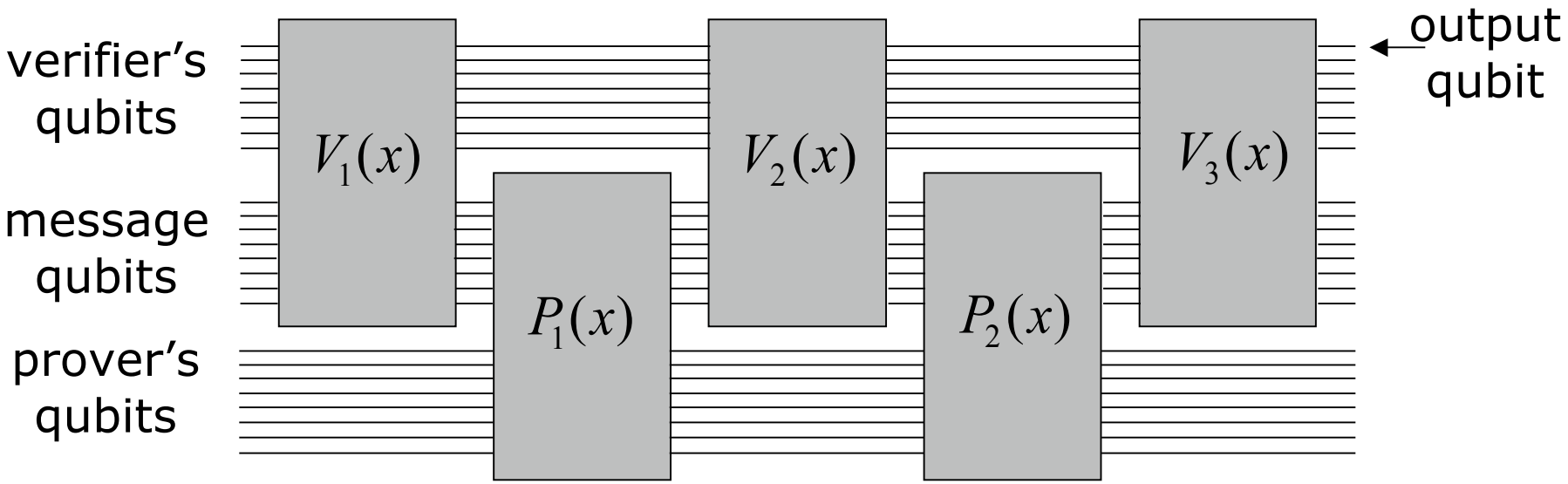


Output: { accept — if the verifier believes the prover that x has property A

reject — otherwise

Formalizing the model

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



Facts about quantum interactive proofs

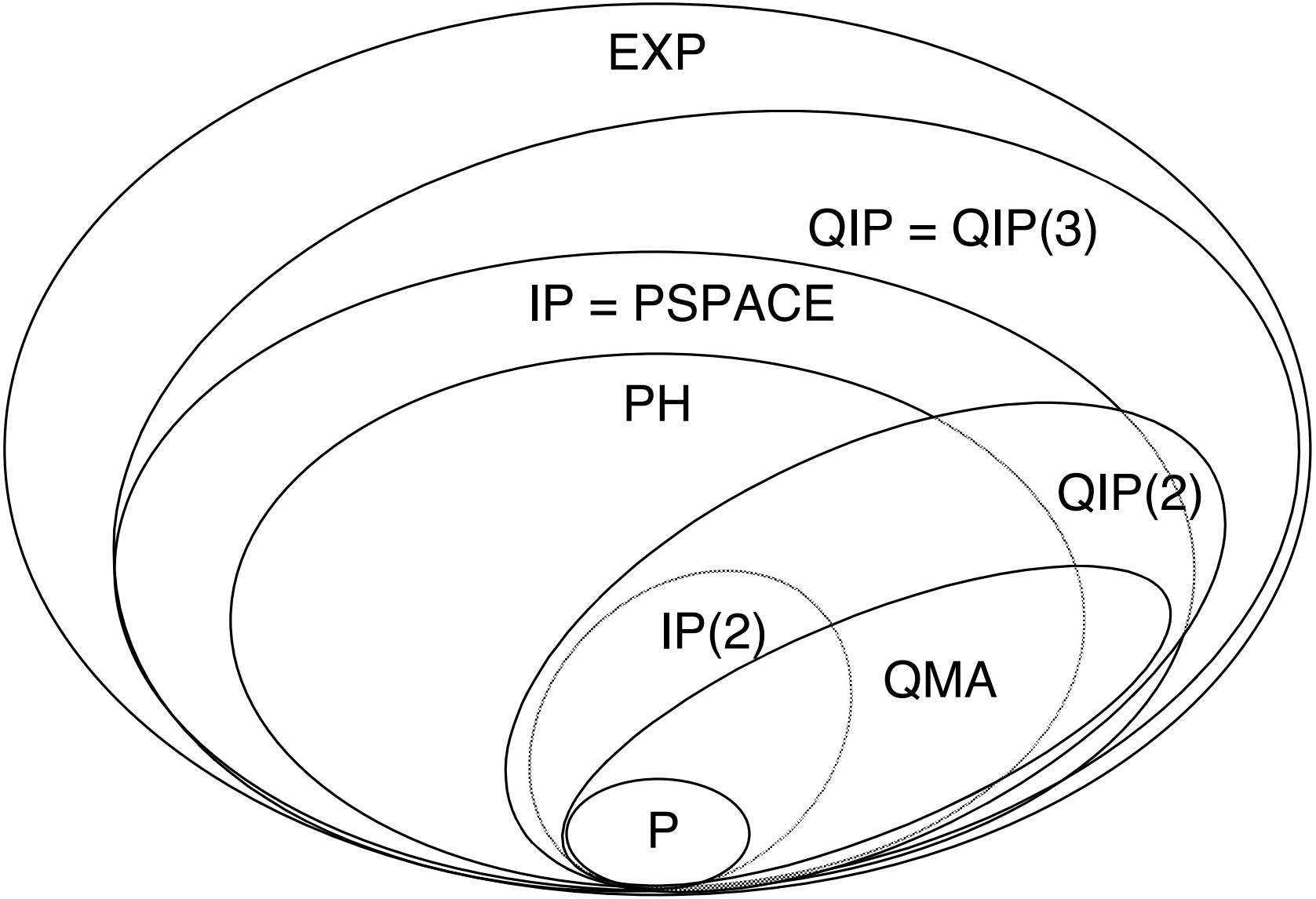
- $IP \subseteq QIP$ and $IP(m) \subseteq QIP(m)$ for any m.
- $QIP = QIP(3)$

$$= PSPACE \subseteq QIP(3)$$

- $QIP \subseteq EXP$ (deterministic exponential time)
- $BQP \subseteq QIP(1) \subseteq PP \subseteq PSPACE$

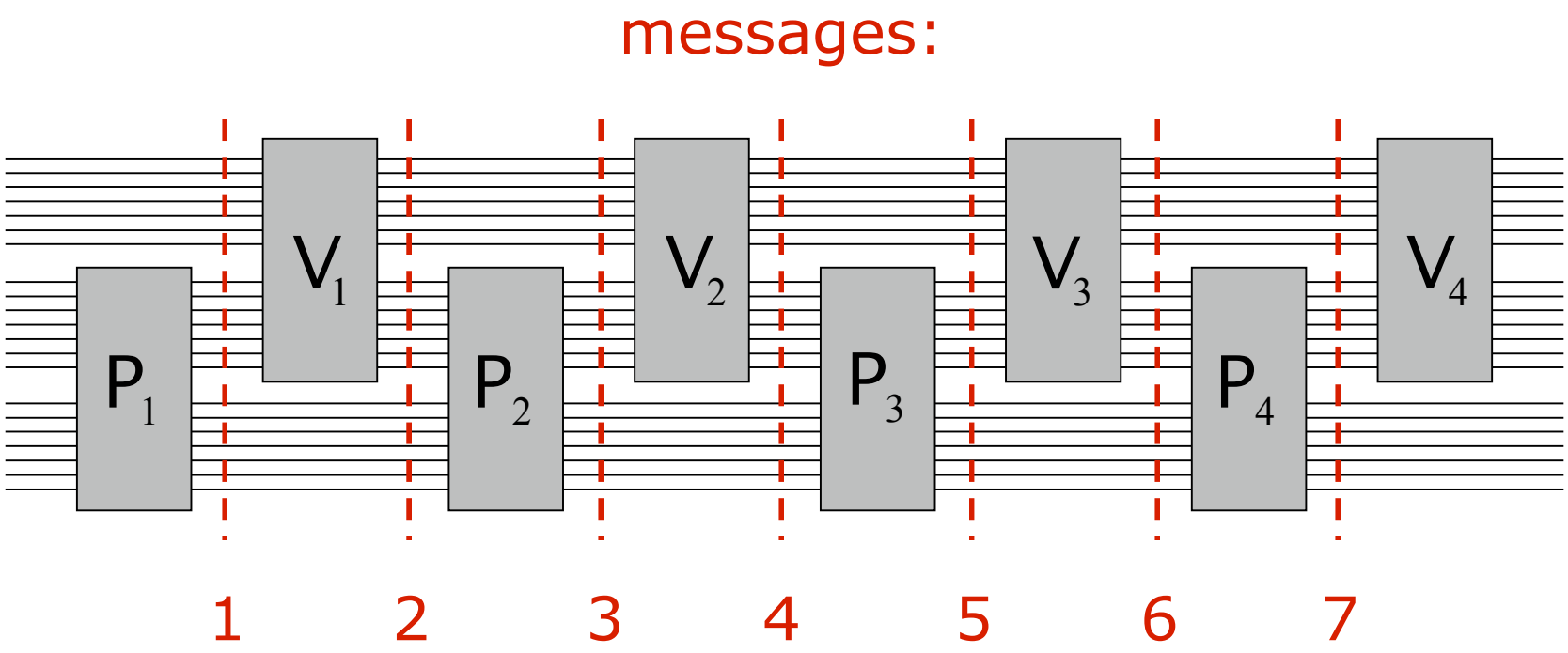
same as QMA

Diagram of complexity classes



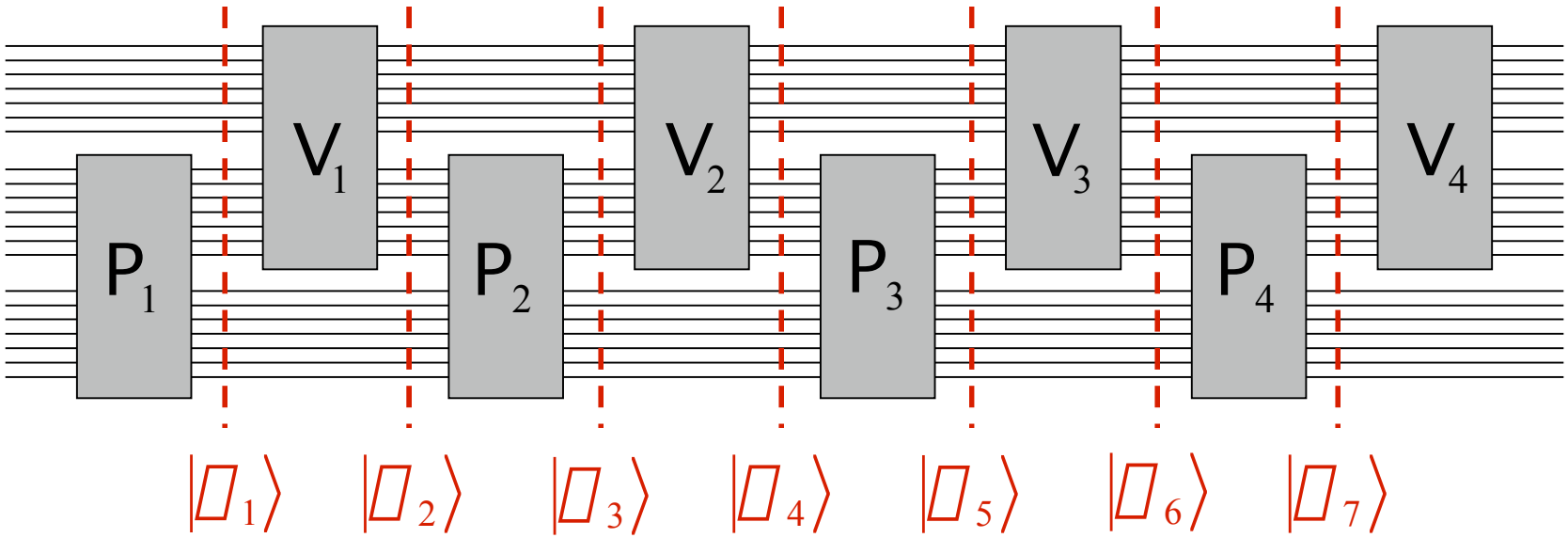
Parallelizing quantum interactive proofs

Suppose we have a quantum interactive proof consisting of several rounds:



Parallelizing quantum interactive proofs

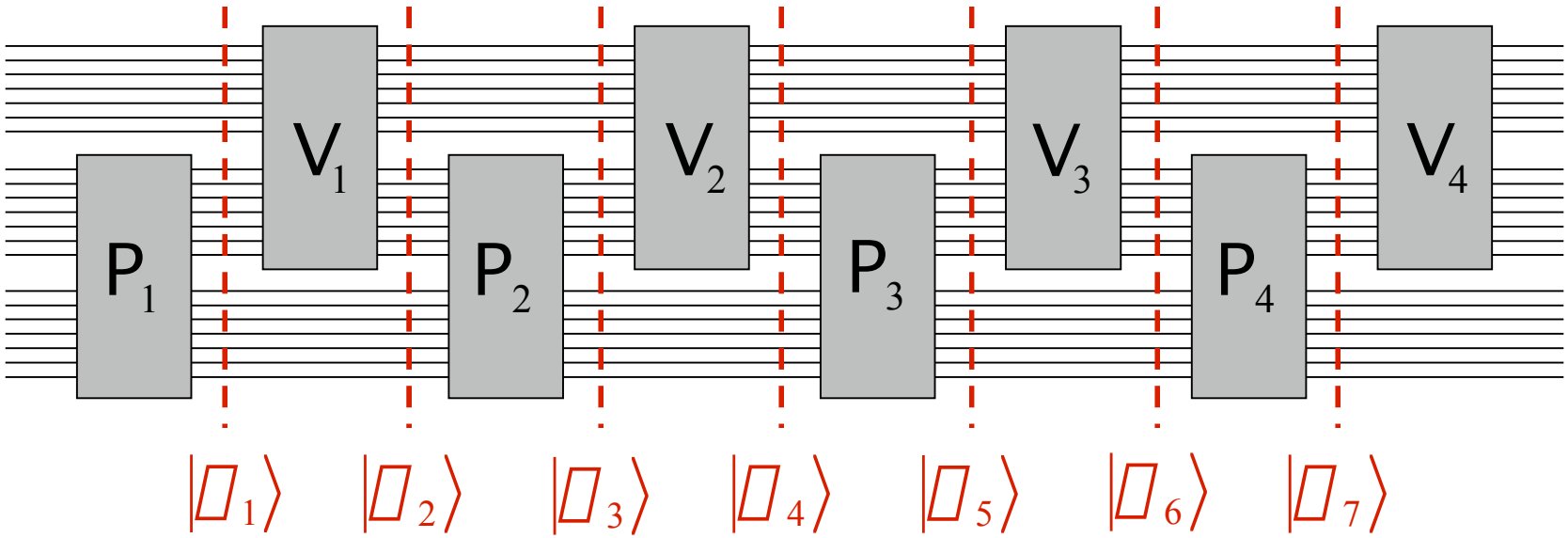
Consider the states of the system during some execution (optimal for the prover):



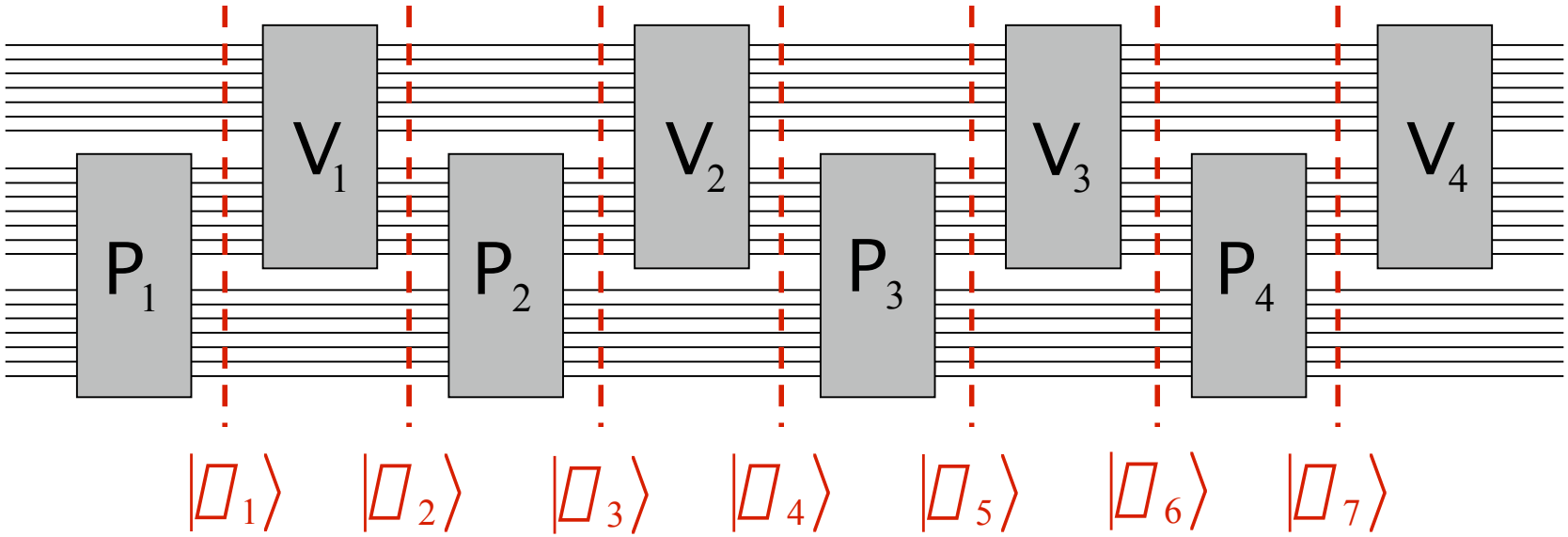
Parallelizing quantum interactive proofs

Message 1 (of parallelized protocol):

The prover sends $|\square_1\rangle$, K , $|\square_m\rangle$ to the verifier.



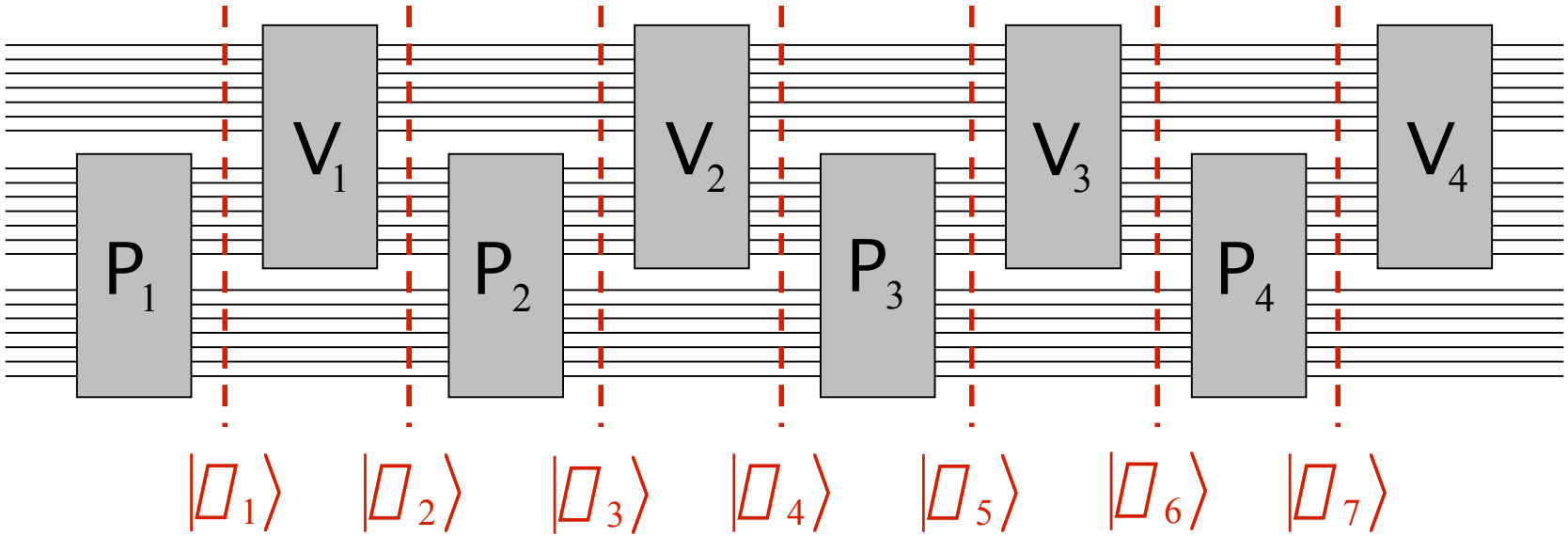
Parallelizing quantum interactive proofs



The verifier now needs to check that these states are consistent with one another…

… this will require 2 additional messages.

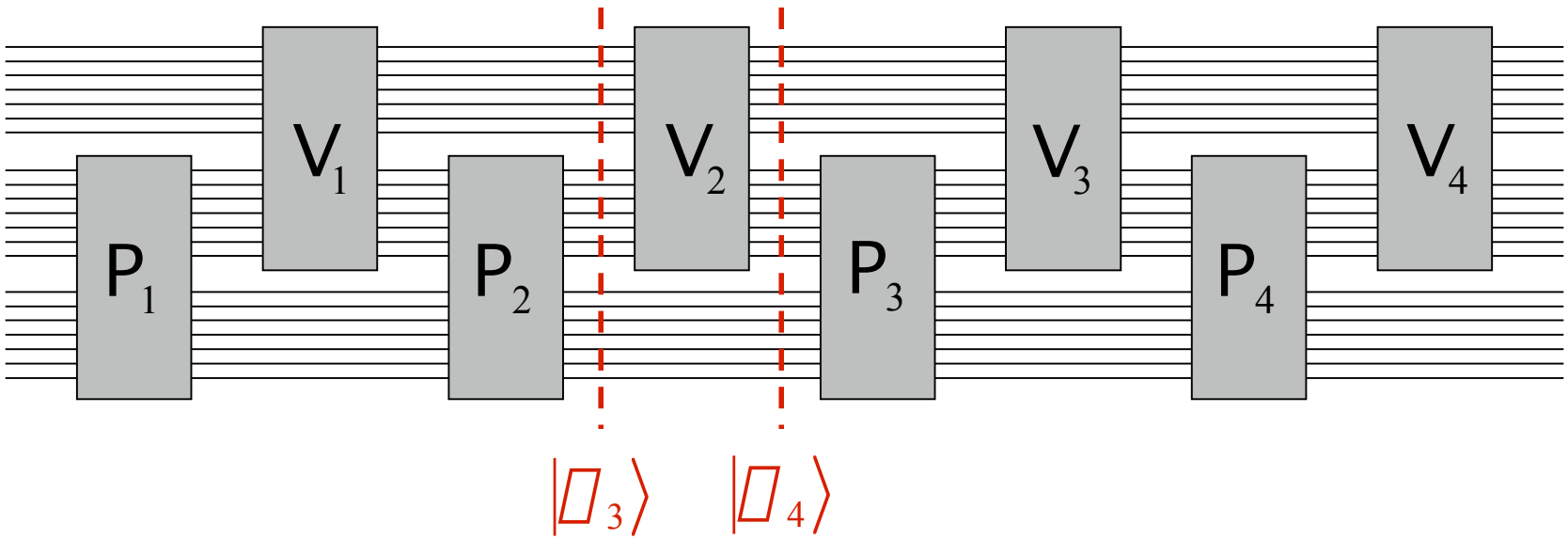
Parallelizing quantum interactive proofs



The verifier randomly chooses 2 consecutive states to test for consistency.

Case 1: states are separated by a verifier transformation.

Parallelizing quantum interactive proofs



The verifier randomly chooses 2 consecutive states to test for consistency.

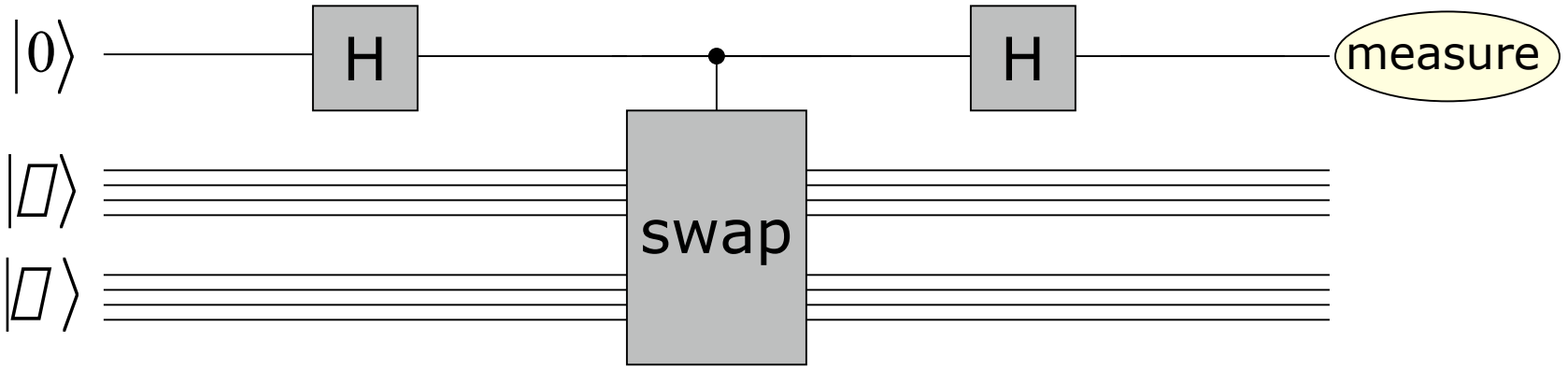
Case 1: states are separated by a verifier transformation. Easy

Swap test

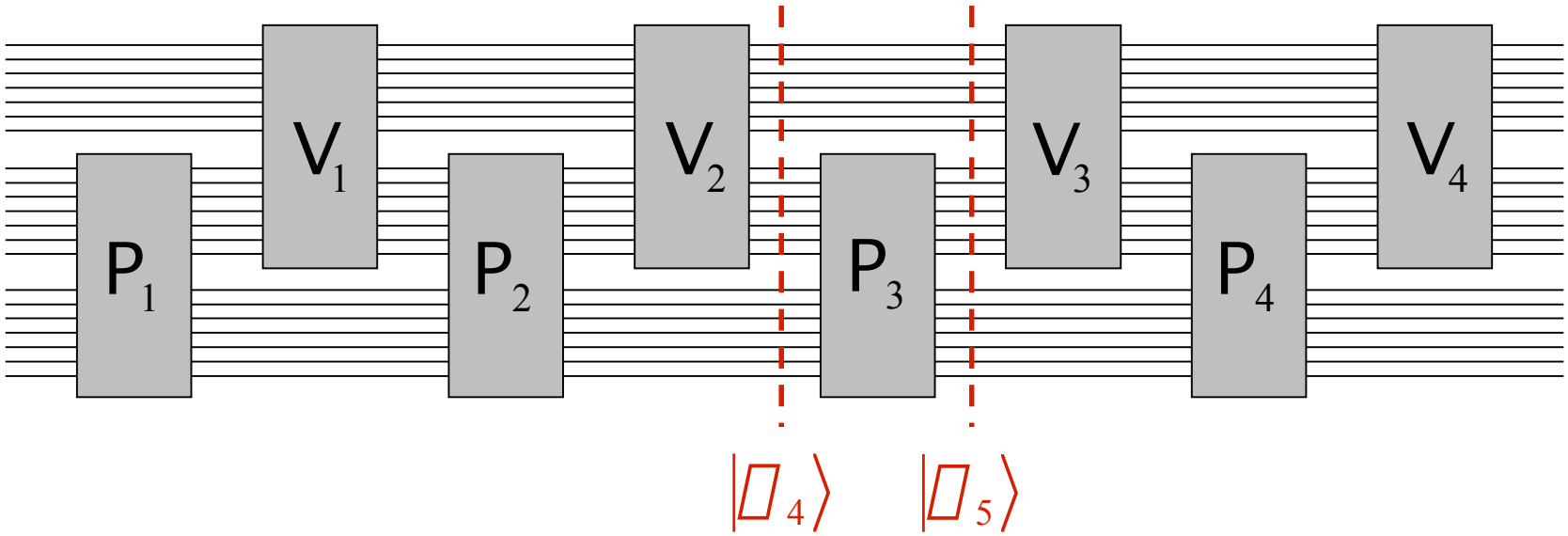
Suppose we have two (pure) quantum states:

$$|\square\rangle \quad \text{and} \quad |\square\rangle$$

Want to know if they are close together or far apart.

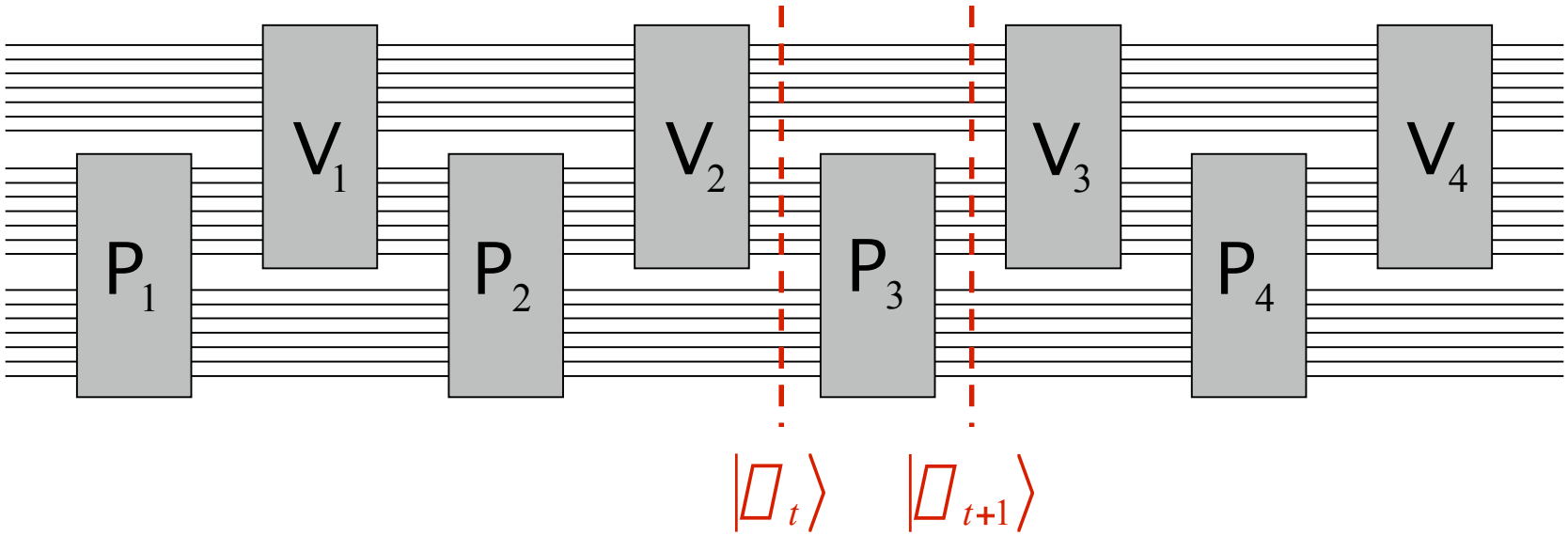


Parallelizing quantum interactive proofs



Case 2: states are separated by a <u>prover</u> transformation.

Parallelizing quantum interactive proofs



Messages 2 and 3 (of parallelized protocol):

Verifier sends the message and private prover qubits of $|\Box_t\rangle$ to the prover… the prover is challenged to convert $|\Box_t\rangle$ to $|\Box_{t+1}\rangle$.

Parallelizing quantum interactive proofs

It turns out that this works…

A cheating prover will be caught with probability at least

$$\frac{c}{m^2}$$

for some constant c.

Proof is highly nontrivial—must take into account cheating prover strategies that use entanglement.

Parallel repetition can be used to reduce soundness error to be exponentially small… still only use 3 messages.

Parallelizing quantum interactive proofs

So what is the difference between quantum and classical that allows this to work?

It seems that it is because the verifier can check that two mixed quantum states are close together (with the help of the prover)…

… not possible to do classically with probability distributions (at least in the way that would be required).

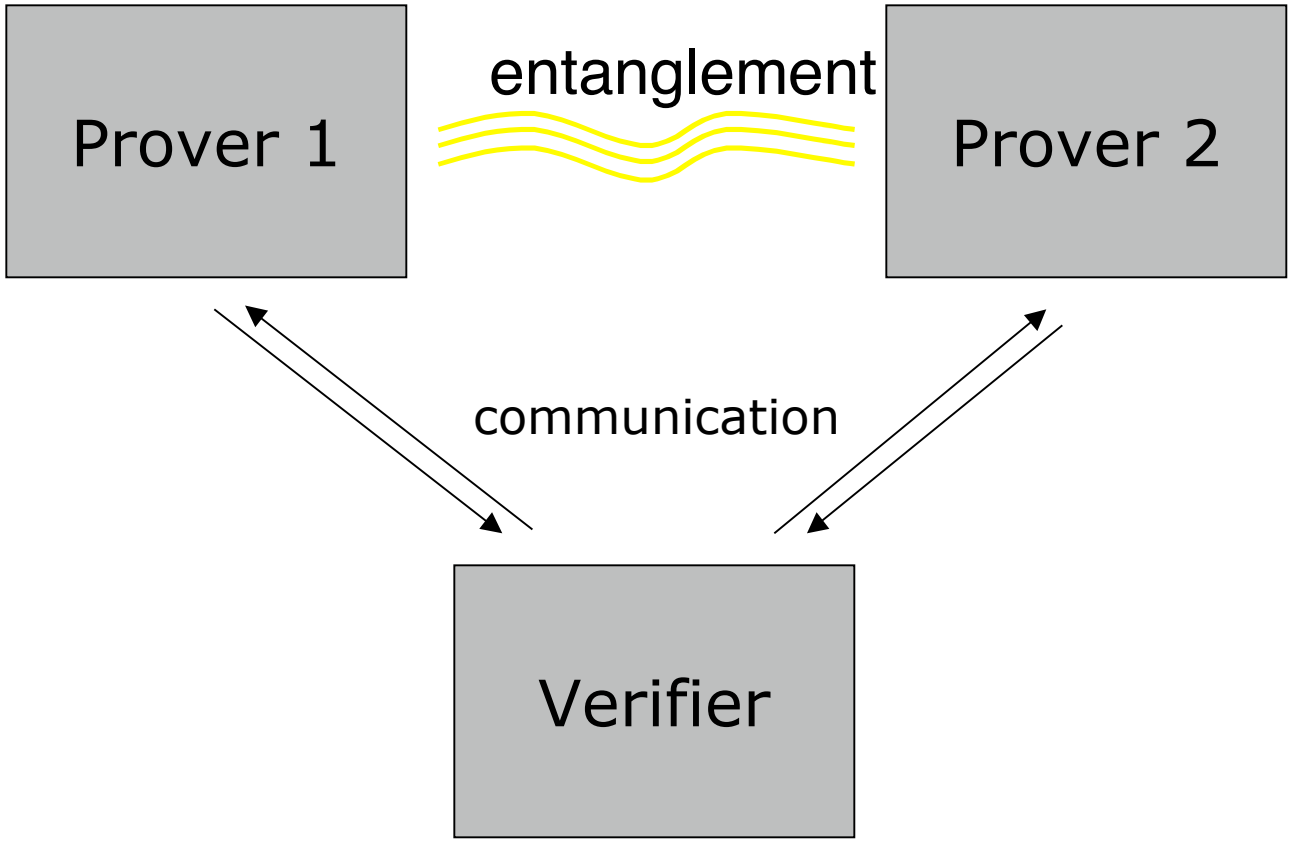
Multi-prover quantum interactive proofs

Just about everything is open…

Classical case: MIP = NEXP, only 1 round is needed, parallel repetition works, …

We do not know any of these things for the quantum case, even when the verifier is classical.

Multi-prover quantum interactive proofs



Bell Inequalities and Multiple Provers

Any upper bound on the probability with which multiple provers can convince a verifier to accept a given input is analogous to a Bell inequality.

Such upper bounds are not valid if the provers initially share entanglement.

Bell Inequalities and Multiple Provers

What we need are upper bounds on the probability that entangled provers can convince the verifier to accept.

These are analogous to Tsirelson Inequalities. These tend to be much harder to prove…

Allowing entangled quantum provers, we could have any of the following:

MIP = NEXP
MIP strictly weaker than NEXP
MIP strictly stronger than NEXP
MIP incomparable with NEXP

Similar for the quantum verifier case.

Example: 3SAT

An instance of 3SAT is an AND of ORs where each OR consists of 3 literals (variables or negations of variables).

Example:

$$(x_1 \quad x_3 \quad x_4) \square (x_2 \quad \bar{x}_3 \quad x_4) \square (\bar{x}_1 \quad \bar{x}_2 \quad x_4) \square (x_1 \quad \bar{x}_2 \quad x_3)$$

Problem: determine if there exists a boolean assignment to the variables that causes the formula to evaluate to true.

This problem is **NP-complete**.

Simple proof system for 3SAT

Randomly choose a clause (OR of 3 literals):

$$(\bar{x}_1 \quad \bar{x}_2 \quad x_4)$$

Ask prover 1 to give an assignment to the variables appearing in the clause:

$$s = (x_1, x_2, x_4) \longrightarrow a \in \{0,1\}^3$$

Randomly select one of these variables, and ask prover 2 to give an assignment:

$$t = x_2 \longrightarrow b \in \{0,1\}$$

Accept if clause is satisfied and answers are consistent, reject otherwise.

Simple proof system for 3SAT

This proof system works, assuming the provers play classically:

If the formula is satisfiable, they can win with certainty.

If the formula is not satisfiable, their probability of winning is bounded away from 1.

This is an example of a general technique sometimes called **oracularization**: randomly ask prover 2 one of the questions asked to prover 1 to force prover 1 to play non-adaptively.

Simple proof system for 3SAT

Quantum case: the proof system does not work!

We can find a 3SAT formula that is not satisfiable, but for which the provers can win the associated game with certainty.

Based on the “Magic Square” [Aravind, 2002], [Mermin, 1990].

How Powerful are Multi-prover Interactive Proofs?

Oracularization is one of the starting points for the study of (classical) multi-prover interactive proof systems…

… but the method is not sound against quantum strategies.

We currently do not know how to fix this problem, and we know very little about the power of multi-prover interactive proofs when the provers can use quantum strategies.

Other Variants of Interactive Proofs

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

- interactive proofs with stronger restrictions on the verifier (or on the prover).
- competing provers
- probabilistically checkable proofs

General problem:

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