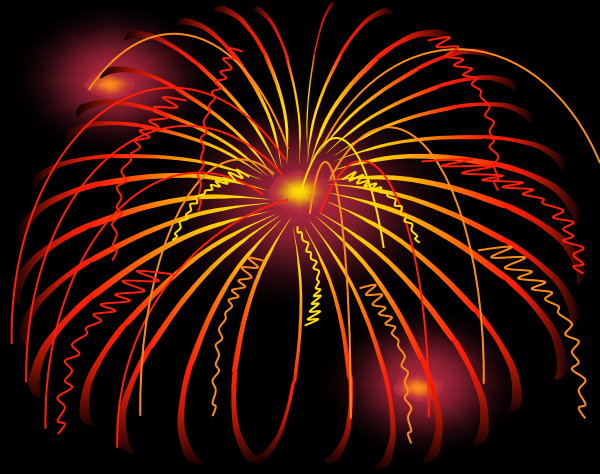


15th Week

Quantum Interactive Proofs and Quantum Optimization

Synopsis.

- Quantum Optimization Problems
- Quantum Interactive Proofs
- Quantum Zero-Knowledge Proofs
- Arthur-Merlin/Merlin-Arthur Games
- Multi-Prover QIP

July 12, 2018. 23:59

Course Schedule: 16 Weeks

Subject to Change

- Week 1: Basic Computation Models
- Week 2: NP-Completeness, Probabilistic and Counting Complexity Classes
- Week 3: Space Complexity and the Linear Space Hypothesis
- Week 4: Relativizations and Hierarchies
- Week 5: Structural Properties by Finite Automata
- Week 6: Stype-2 Computability, Multi-Valued Functions, and State Complexity
- Week 7: Cryptographic Concepts for Finite Automata
- Week 8: Constraint Satisfaction Problems
- Week 9: Combinatorial Optimization Problems
- Week 10: Average-Case Complexity
- Week 11: Basics of Quantum Information
- Week 12: BQP, NQP, Quantum NP, and Quantum Finite Automata
- Week 13: Quantum State Complexity and Advice
- Week 14: Quantum Cryptographic Systems and Quantum Functions
- Week 15: Quantum Interactive Proofs and Quantum Optimization
- Week 16: Final Evaluation Day (no lecture)

YouTube Videos

- This lecture series is based on numerous papers of T. Yamakami. He gave conference talks (in English) and invited talks (in English), some of which were video-recorded and uploaded to YouTube.
- Use the following keywords to find a playlist of those videos.
- YouTube search keywords:

  Tomoyuki Yamakami conference invited talk playlist

Conference talk video

Main References by T. Yamakami I

- T. Yamakami. Quantum optimization problems. In Proc. of UMC 2002, LNCS, Vol.2509, pp.300-314 (2002)
- T. Yamakami. Multiple quantum zero-knowledge proofs with constant space verifiers. An oracle presentation at CEQIP 2007 (2007).
- H. Kobayashi, K. Matsumoto, and T. Yamakami. Quantum Merlin-Arthur proof systems: are multiple Merlins more helpful to Arthur? Chicago Journal of Theoretical Computer Science, Vol. 2009, Article 3 (2009)
- H. Nishimura and T. Yamakami. An application of quantum finite automata to interactive proof systems. Journal of Computer and System Sciences 75, 255-269 (2009)

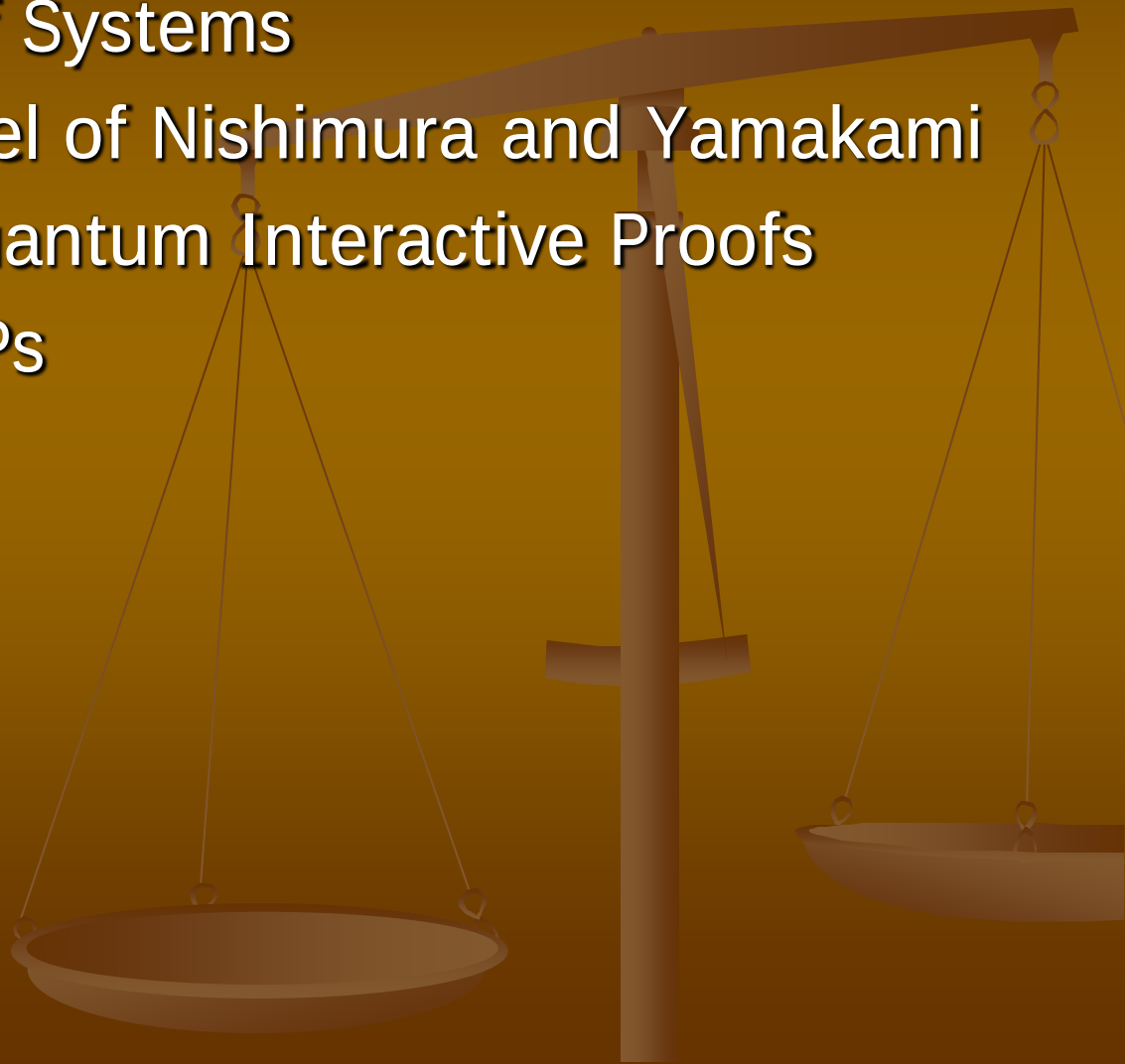
(To be continued)

Main References by T. Yamakami II

- M. Villagra and T. Yamakami. Quantum and reversible verification of proofs using constant space. In Proc. of TPNC 2014, LNCS, Vol. 8890, pp.144-156 (2014)
- T. Yamakami. Constant-space quantum interactive proof systems against multiple provers. Information Processing Letters 114, 611-619 (2014)
- H. Nishimura and T. Yamakami. Interactive proofs with quantum finite automata. Theoretical Computer Science 568, 1-18 (2015)

I. Quantum Interactive Proofs

1. Interactive Proof Systems
2. A Quantum Model of Nishimura and Yamakami
3. Single-Prover Quantum Interactive Proofs
4. Properties of QIPs



Interactive Proof Systems I (revisited)

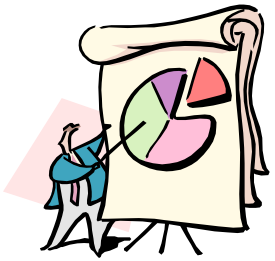
- We have already discussed a classical case of interactive proof systems in Week 7.

- A language L has an IP system $\Leftrightarrow$ there exists a verifier V that satisfies the following two conditions: for a certain constant $\varepsilon \in [0,1/2)$,
  1. For every $x \in L$, there exists a honest prover P such that V accepts a proof from P with probability at least $1/2+\varepsilon$; and
  2. For every $x \notin L$, V rejects any proof from any (possibly malicious) prover with probability at least $1/2+\varepsilon$.

A proof is a piece of information.

Interactive Proof Systems II (revisited)

- A language L has an IP system $\Leftrightarrow$ there exists a verifier V that satisfies the following two conditions: for a certain $\varepsilon \in [0,1/2)$,
  1. For every $x \in L$, there exists a honest prover P such that V accepts a proof from P with probability at least $1/2+\varepsilon$; and
  2. For every $x \notin L$, V rejects any proof from any (possibly malicious) prover with probability at least $1/2+\varepsilon$.



prover

$x \in L$

$x \notin L$

Accepts with probability $\geq 1/2+\varepsilon$

Rejects with probability $\geq 1/2+\varepsilon$

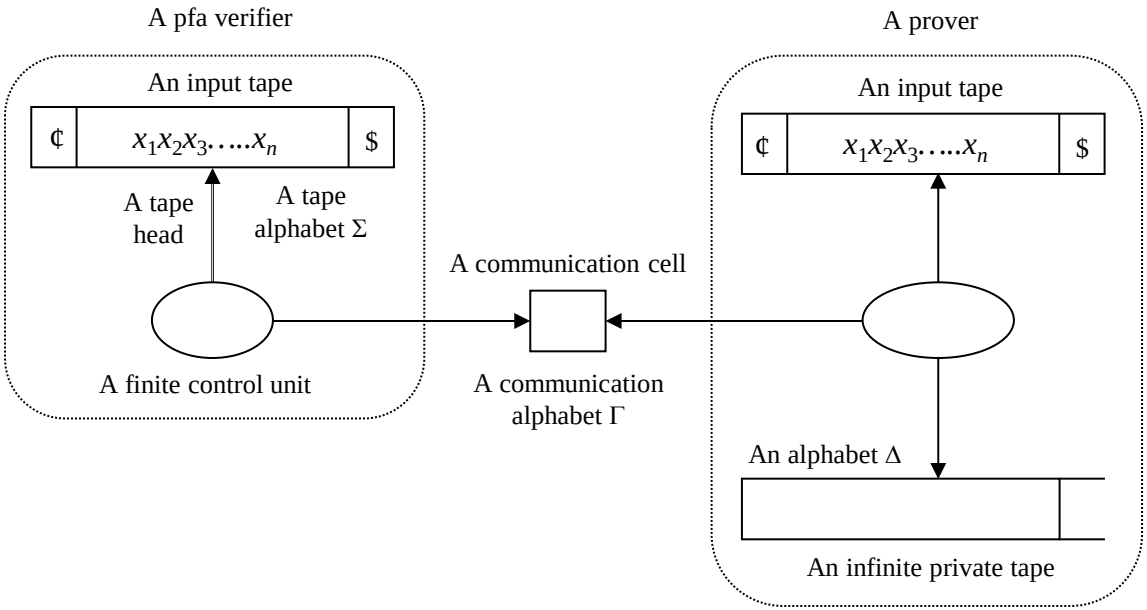
verifier

Believe me. This is a correct proof.

Let me judge the correctness of your proof.

An Illustration of IP System (revisited)

- Dwork-Stockmeyer IP system, based on probabilistic finite automata (pfa's), is illustrated as follows.



Constant-Space Interactive Proofs (revisited)

- Dwork and Stockmeyer (1992) considered interactive proof (IP) systems with 2-way probabilistic finite automata (2pfa's).
- Major advantages: we can prove certain separation results that are impossible (at least at present) to obtain for polynomial-time or logarithmic-space bounded IP systems.
- Finite automata can be viewed as constant-space Turing machines with read-only input tapes and work tapes.

- IP(⟨restrictions⟩) = the class of all languages that have IP systems satisfying the restrictions given in ⟨restrictions⟩.
  - IP(2pfa,poly-time) = the class of all languages that have IP systems with 2pfa verifiers running in expected polynomial time.

Private Coins vs. Public Coins (revisited)

- In an IP system, a verifier is considered to choose the next transition probabilistically by flipping coins.
- When the outcomes of such coin flips are hidden from a prover, we say that the verifier uses private coins.
- By contrast, if the verifier reveals the outcomes, the verifier is said to use public coins.
- A public-coin version of IP systems are called Arthur-Merlin proof systems or Arthur-Merlin games.

- AM(⟨restrictions⟩) = the class of all languages that have public-coin IP systems satisfying the restrictions given in ⟨restrictions⟩
  - ➢ AM(2pfa) = the class of all languages that have public-coin IP systems with 2pfa verifiers

A Quantum Model of Nishimura and Yamakami

- What if quantum computation is used for IP systems?

- Nishimura and Yamakami (2009) presented a quantum analogue of IP(2pfa) and IP(2pfa,poly-time) by replacing probabilistic finite automata (pfa's) with Kondacs-Watrous quantum finite automata (qfa's).
- Moreover, Nishimura and Yamakami used quantum provers in place of classical provers.

Single-Prover Quantum Interactive Proofs

- Nishimura and Yamakami (2009) introduced the notation QIP(⟨restrictions⟩) in a way similar to define IP(⟨restrictions⟩).

- QIP(⟨restrictions⟩) = the class of all languages that have single-prover QIP systems between quantum provers and quantum verifiers with restrictions given in ⟨restrictions⟩.
  - QIP(1qfa) = a quantum analogue of IP(1pfa) with 1pfa verifiers
  - QIP(2qfa,poly-time) = similar with 2qa verifiers running in expected polynomial time

Examples of QIP Systems I

- Let us construct a simple QIP system for the marked even-length palindromes:

$$\mathrm{Pal}_{\#} = \{\, x\#x^{R} \mid x \in \{0,1\}^{*} \,\}.$$

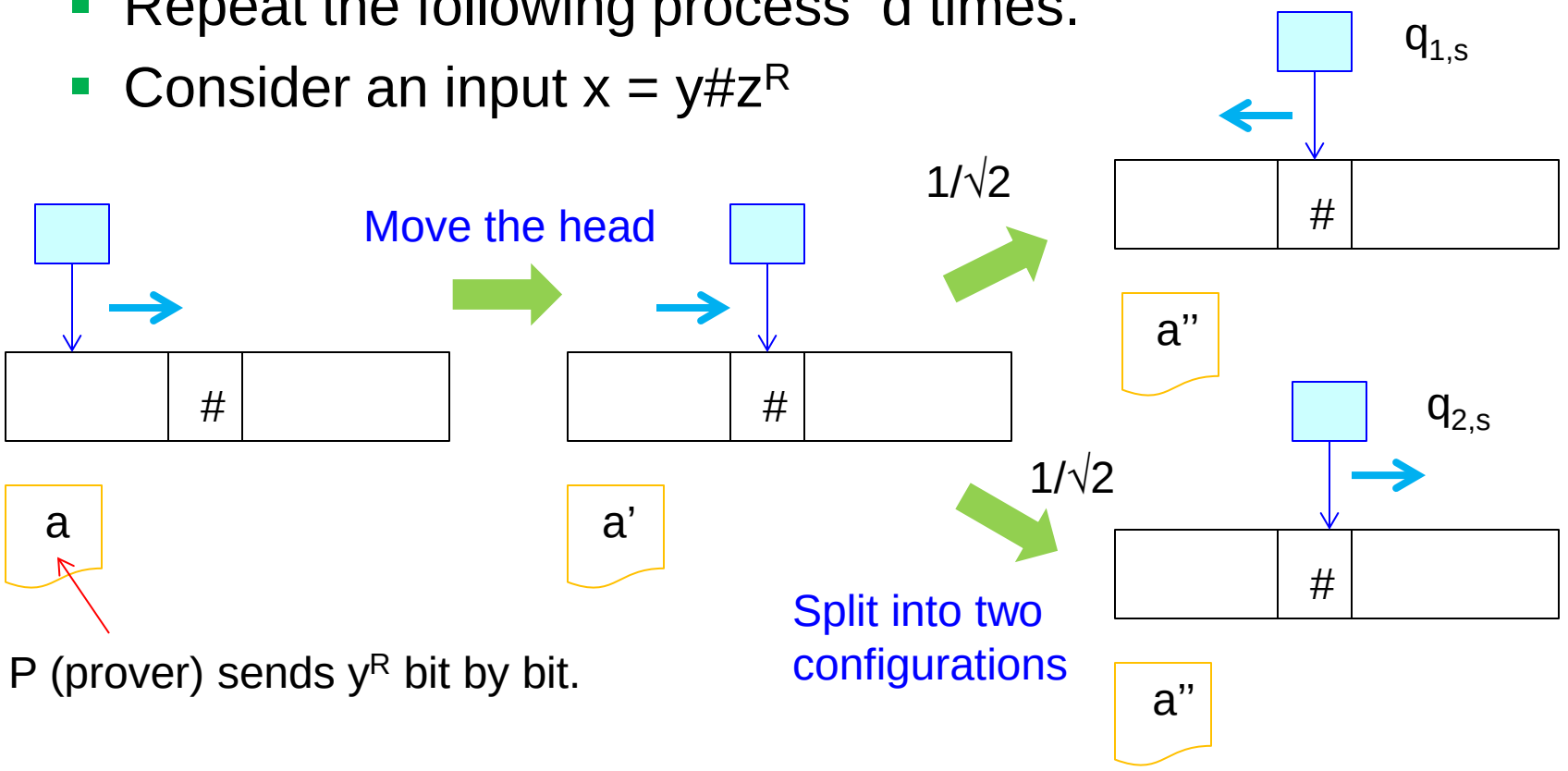
A verifier works as follows.

$$V_{¢}|q_{0,s}\rangle|\#\rangle = |q'_{0,s}\rangle|\#\rangle \qquad V_{¢}|q_{1,s}\rangle|\#\rangle = |q_{0,s0}\rangle|\#\rangle$$
$$V_{¢}|q_{i,s}\rangle|a\rangle = |r_{i,s}\rangle|a\rangle \qquad V_{\$}|q_{i,s}\rangle|a\rangle = |r_{i,s}\rangle|a\rangle$$
$$V_{\$}|q'_{0,s}\rangle|\#\rangle = |r_{0,s}\rangle|\#\rangle \qquad V_{\$}|q_{2,s}\rangle|\#\rangle = |q_{0,s1}\rangle|\#\rangle$$
$$V_{\#}|q'_{0,s}\rangle|\#\rangle = \frac{1}{\sqrt{2}}(|q_{1,s}\rangle|\#\rangle + |q_{2,s}\rangle|\#\rangle) \qquad V_{\#}|q_{i,s}\rangle|b\rangle = |r_{i,s}\rangle|b\rangle$$
$$V_{a}|q'_{0,s}\rangle|\#\rangle = |q'_{0,s}\rangle|\#\rangle$$
$$V_{a}|q_{i,s}\rangle|a\rangle = |q_{i,s}\rangle|a\rangle \qquad V_{a}|q_{i,s}\rangle|a'\rangle = |r_{i,s}\rangle|a'\rangle$$

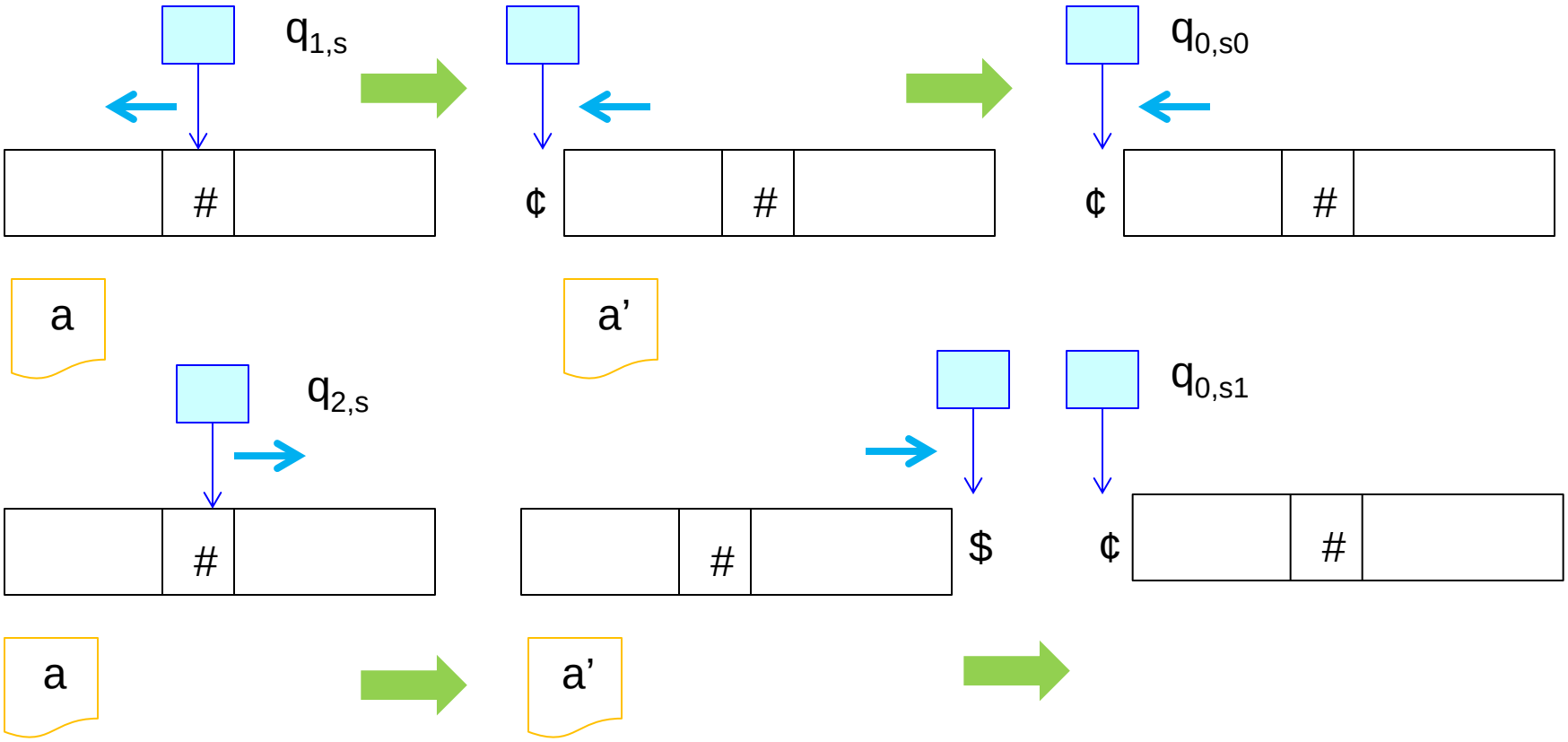
$$D(q_{1,s}) = -1 \qquad D(q_{2,s}) = 1$$
$$D(q'_{0,s}) = 1 \qquad D(q_{0,s0}) = 0$$
$$D(q_{0,s1}) = 1$$

Examples of QIP Systems II

- (P,V) protocol for $Pal_{\#}$
  - Let $\varepsilon \in (0,1/2)$ be any constant and let $d = \lceil \log_2(1/\varepsilon) \rceil$.
  - Let $s = \lambda$ (empty string) and start with $q_{0,s}$.
  - Repeat the following process d times.
  - Consider an input $x = y\#z^R$



Examples of QIP Systems III



Move the head in opposite directions

The head jumps from $ to ¢ because the tape is circular.

Examples of QIP Systems III

- Consider the case where x is of the form $y\#y^R$.
  - When honest P provides y^R to V, V correctly checks x is of the from $y\#y^R$ with probability 1.

- Consider the case where x is of the from $y\#z^R$ with $y \neq z$.
  - No matter what a dishonest P* provides, V mistakenly rejects with probability at least 1/2.
  - Since we repeat the process d times, the total rejection probability is at most $\sum_{i=1}^{d} 2^{-i} = 1 - 2^{-d} \geq 1 - \varepsilon$.

Properties of QIPs I

- Single-prover QIP systems with 1-qfa verifiers.
- Theorem: [Nishimura-Yamakami (2009)]
  1QFA $\subseteq$ QIP(1qfa) = REG.
- Since 1QFA $\neq$ REG, we obtain 1QFA $\neq$ QIP(1qfa).

❑ Proof Sketch:

- It suffices to show that QIP(1qfa) = REG.
- REG $\subseteq$QIP(1qfa) is shown by using the honest prover as an eraser to guarantee the reversibility of verifiers.
- QIP(1qfa)$\subseteq$REG requires a notion of 1-tiling complexity.

QED

Properties of QIPs II

- Single-prover QIP systems with 2-qfa verifiers.

- Theorem: [Nishimura-Yamakami (2009)]
  1. REG $\subseteq$ QIP(2qfa,poly-time) $\subseteq$ NP.
  2. QIP(2qfa,poly-time) $\not\subseteq$ AM(2pfa).

❑ Proof Sketch:

- (1) The last inclusion follows from a direct simulation.
- (2) This comes from the fact that $Pal_{\#} \in$ QIP(2qfa,poly-time) but $Pal_{\#} \notin$ AM(2pfa) [Dwork-Stockmeyer (1992)]

QED

Properties of QIPs III

- In our single-prover QIP systems, provers behave quantumly.
- When those provers behave classically (i.e., they use classical deterministic moves), we use the term "c-prover".
- Let Center = { x1y | x,y$\in${0,1}*, |x|=|y| }.
- Theorem: [Nishimura-Yamakami (2015)]
  1. AM(2pfa) $\subseteq$ QIP(2qfa,c-prover).
  2. Center $\in$ QIP(2qfa,poly-time,c-prover).
- (Open Problem)

  Is it true that Center $\in$ QIP(2qfa,poly-time)?

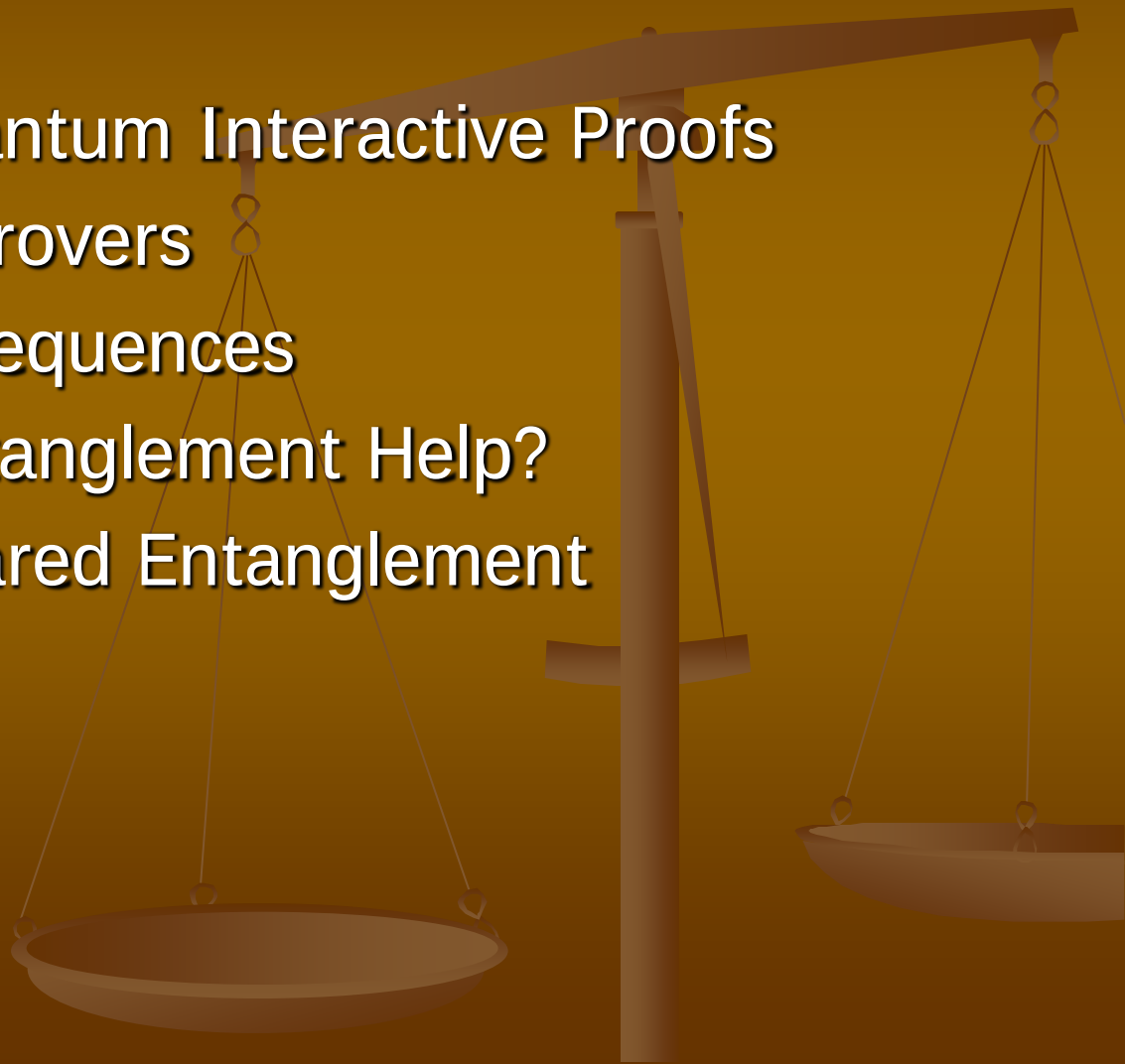
Open Problems

- Consider the following issues associated with QIP systems.

1. What if we restrict prover's ability?
2. Compare between public coins and private coins.
3. Consider the case of communicating through classical channels.
4. Minimize the number of interactions.
5. Study the effect of using prior entanglement.

II. Multiple-Prover QIP Systems

1. Multiple Provers
2. Multi-Prover Quantum Interactive Proofs
3. Power of Multi-Provers
4. Immediate Consequences
5. Does Shared Entanglement Help?
6. Polynomially-Shared Entanglement



Multiple Provers

- A standard model of IP system uses one single prover communicating with a verifier.
- Instead, Feige and Shamir (1992) considered a multiple-prover model of weak-verifier IP system.
- 2IP(2pfa,poly-time) = the language class defined by IP systems with 2 provers
- (Claim) [Feige-Shamir (1992)]
  1. 2IP(2pfa,poly-time) = NEXP
  2. 2IP(2pfa) = RE
- RE stands for the set of recursively enumerable languages.

Multi-Prover Quantum Interactive Proofs

- Similar to the classical case, we can consider a QIP system with multiple provers who do not communicate with each other.

- QMIP(⟨restrictions⟩) = similar to QIP but with multiple provers and restrictions given in ⟨restrictions⟩.
  - QMIP(2qfa,poly-time) = the language class defined by QIP systems with multiple provers and a verifier running in expected polynomial time

Power of Multi-Provers

- We can show the following statements.
- In what follows, all amplitudes are limited to polynomial-time approximable amplitudes.

- Theorem: [Yamakami (2014)]
  1. CFL $\subseteq$ QMIP(1qfa) $\subseteq$ NE
  2. QMIP(2qfa) = RE
  3. QMIP(2qfa,poly-time) = NEXP.

- In the following slides, we will give the proof sketches of this theorem.

Proof Ideas I

- We want to show that QMIP(2qfa) = RE.

- Lemma:
  1. QMIP(2qfa) $\subseteq$ RE.
  2. QMIP(2qfa,poly-time) $\subseteq$ NEXP.

❑ Proof Sketch:

- Note that quantum provers can use any amount of quantum memory in their computation.
- The claim follows from the fact that the quantum-prover's private memory can be reduced to polynomial size.

QED

Proof Ideas II

- Next, we want to show that QMIP(2qfa,poly-time) = NEXP.

- Lemma:
  1. 2IP(2pfa) $\subseteq$ 2QIP(2qfa).
  2. 2IP(2pfa,poly-time) $\subseteq$ 2QIP(2qfa,poly-time).

❑ Proof Sketch:

- By an “almost” straightforward simulation except for the use of one prover as an eraser who stores all the information discarded from the other provers and the verifier.

QED

Immediate Consequences

- The previous theorem leads to the following consequences.

- Theorem: [Yamakami (2014)]
  1. QIP(2qfa,poly-time) $\neq$ QMIP(2qfa,poly-time).
  2. QIP(1pfa) $\neq$ QMIP(1qfa).

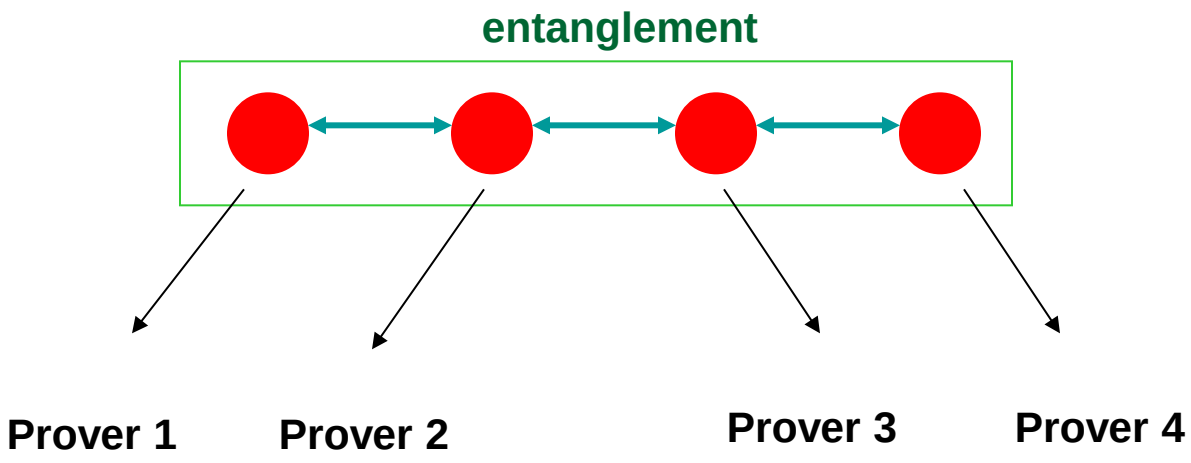
❑ Proof Ideas:

- Recall that QIP(2qfa,poly-time) $\subseteq$ NP.
- Note that QIP(1qfa) = REG.

QED

Does Shared Entanglement Help?

- Entanglement is an essence of quantum computation.
- What if multiple provers share entangled qubits?
- In a one-way communication model, such entanglement does not help. How about two-way communication model?

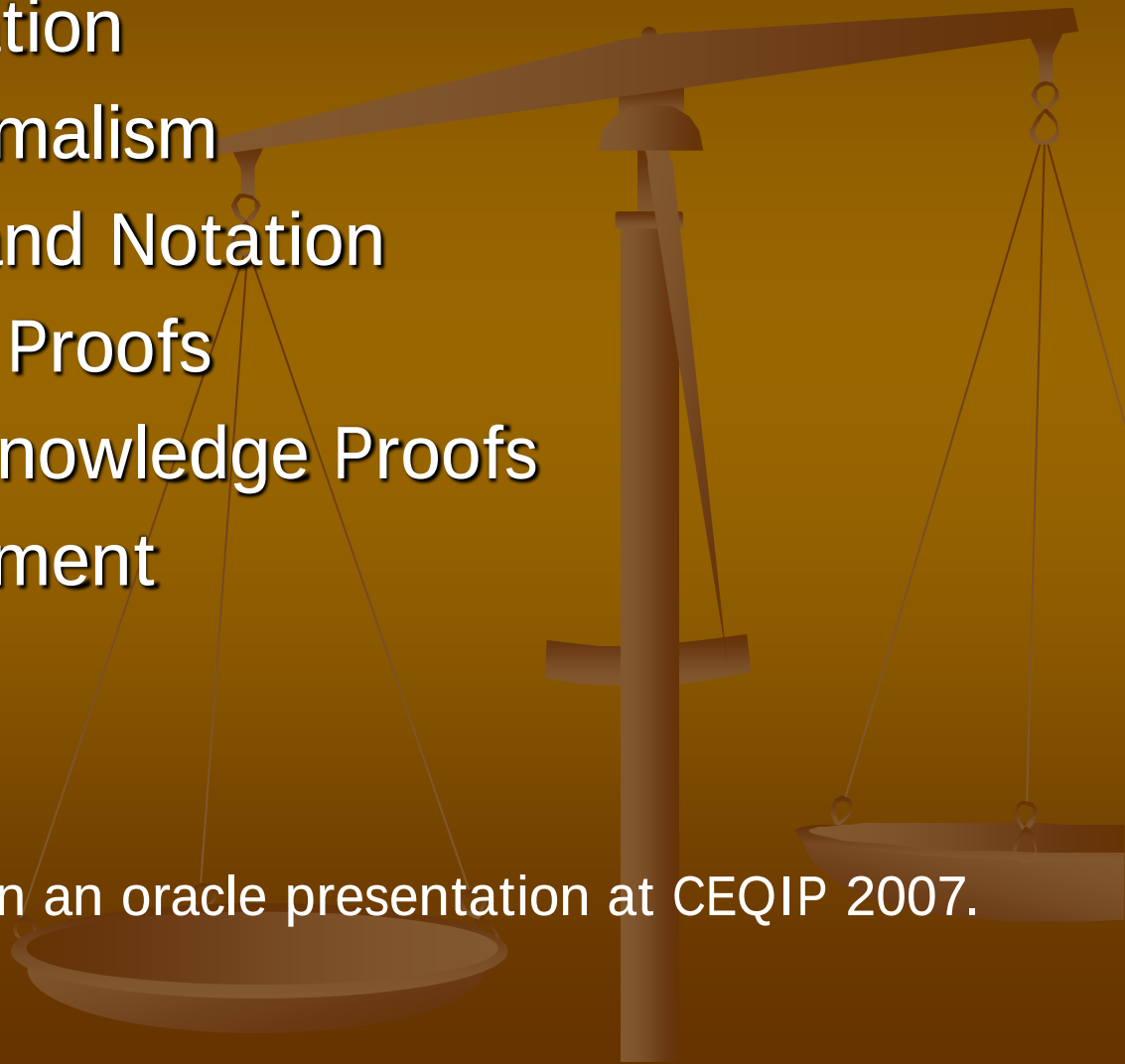


Polynomially Shared Entanglement

- With a polynomial amount of shared entanglement, we expand our multiple-prover QIP.
- We use the new restriction:
  - poly-shared = certain polynomially-many entangled qubits being shared among multiple provers
- QMIP(2qfa,poly-shared,poly-time) = the language class by expected-polynomial-time QMIP systems whose provers share polynomially-many entangled qubits
- There is an unsettling question of how useful shared entanglement really is in a various setting of quantum computation.
- (Open Problem) Does QMIP(2qfa,poly-time) coincide with QMIP(2qfa,poly-shared,poly-time)?

III. Zero-Knowledge Proofs

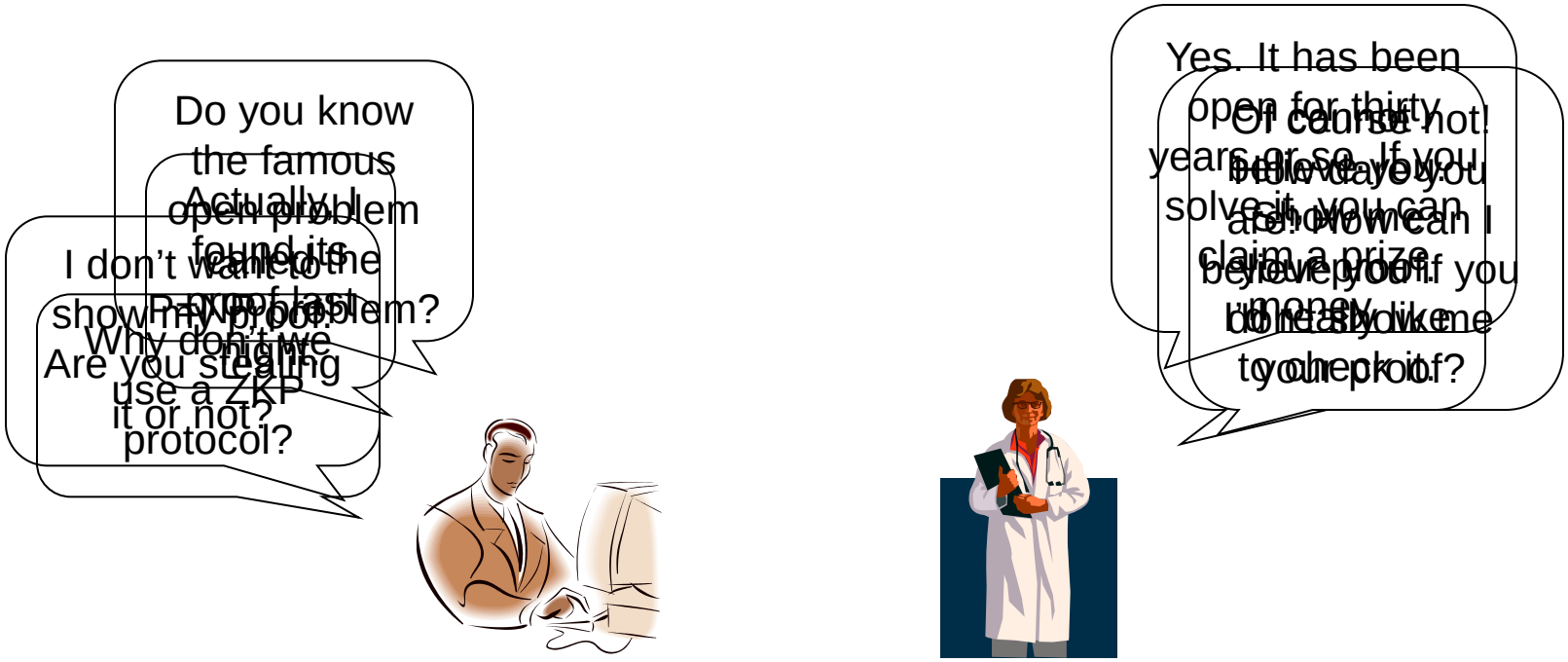
1. Intuitive Explanation
2. Returning to Formalism
3. Basic Concepts and Notation
4. Zero-Knowledge Proofs
5. Quantum Zero-Knowledge Proofs
6. Shared Entanglement



(*) This section is based on an oracle presentation at CEQIP 2007.

Intuitive Explanation

- Roughly, a zero-knowledge property is about a technical way to convince people that your claim is true without telling it.
- Is this a magic or hoax? What do ordinary people say?



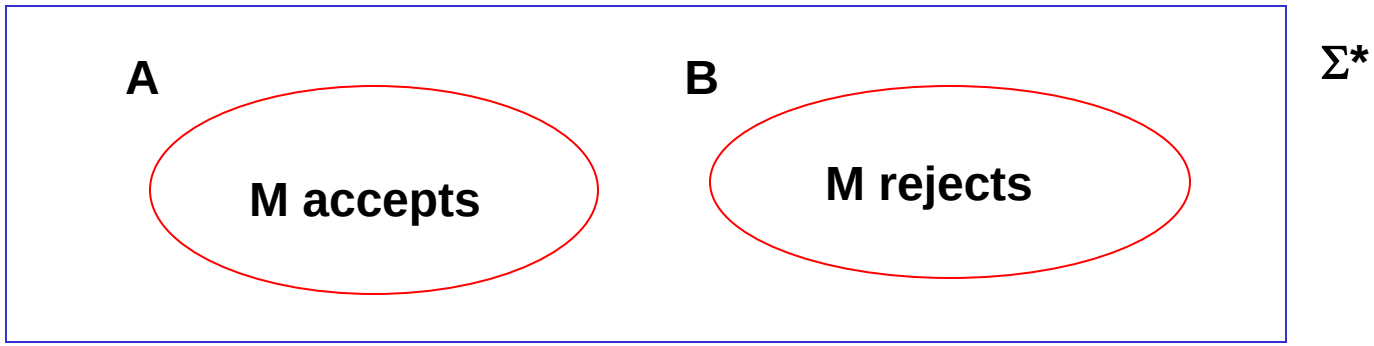
Returning to Formalism

- Let’s go back to the formal definition of zero-knowledge proofs.
- We are focused on a restricted model with constant-space verifiers.
- A core framework of our zero-knowledge proof system is the same as our interactive proof system.

- How can we formulate the zero-knowledge property using finite automata?
- Apparently, we cannot take a standard definition of zero-knowledge property for polynomial-time verifiers.

Basic Concepts and Notation

- A partial problem is a pair (A,B) of sets over a common alphabet such that $A \cap B = \varnothing$.
- A MIP protocol or a machine M recognizes (A,B) with error probability $\varepsilon \Leftrightarrow$ for any input x,
  1. if $x \in A$, M accepts x with probability $\geq 1 - \varepsilon$, and
  2. if $x \in B$, M rejects x with probability $\geq 1 - \varepsilon$.



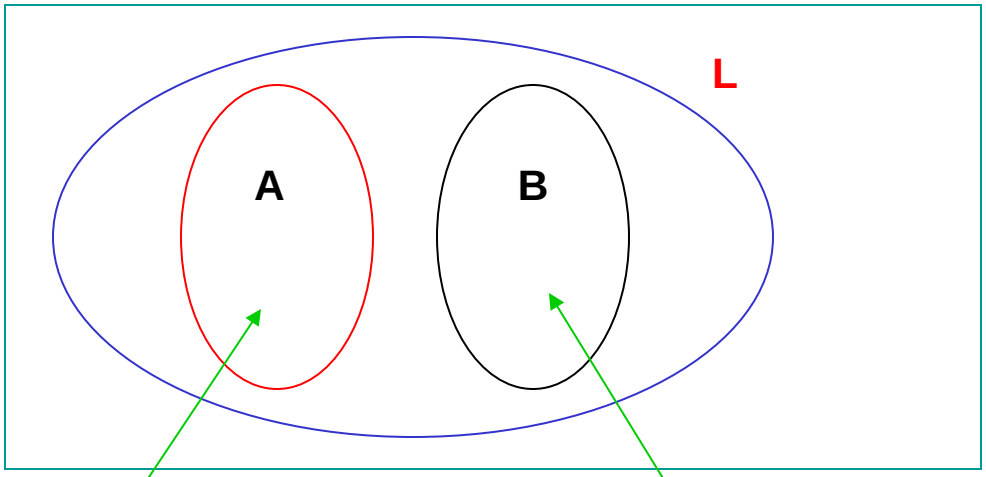
Zero-Knowledge Proofs

- Dwork and Stockmeyer (1992, in another paper) defined weak-verifier zero-knowledge proof (ZKP) systems.
- Zero-knowledge property: any (untrusted) verifier can learn nothing more than the validity of the outcome from the interaction with a honest prover.
- Dwork and Stockmeyer used the (recognition) zero-knowledge property, explained below.

- Recognition zero-knowledge property:

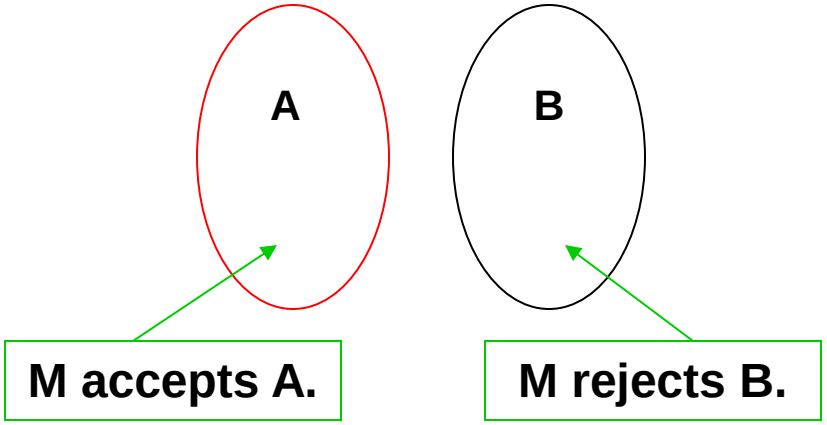
  A k-IP system $(P_1,P_2,\ldots,P_k,V)$ satisfies the (recognition) zero-knowledge for language L over class C of verifiers $\Leftrightarrow$ for every partial problem (A,B) with $A \cup B \subseteq L$, every verifier $V^* \in C$, and every constant $\varepsilon<1/2$, if $(P_1,P_2,\ldots,P_k,V^*)$ recognizes (A,B) with error probability at most ε, then there exists a constant $\varepsilon'<1/2$ and a 2pfa M such that M recognizes (A,B) with error probability at most ε'.

Recognition Zero-Knowledge Property



V* accepts A with help of (P_1,P_2).

V* rejects B with help of (P_1,P_2).



M accepts A.

M rejects B.

(P_1,P_2,V^*) recognizes (A,B) with error probability < ε, where V* is a cheater.

Σ^*

Assuming that provers are all honest and helping a verifier.

A 2pfa M recognizes (A,B) with error probability < ε' without any help of (P_1,P_2).

Every information V* can obtain from (P_1,P_2) can be obtained without (P_1,P_2).

(Classical) ZKP Notation

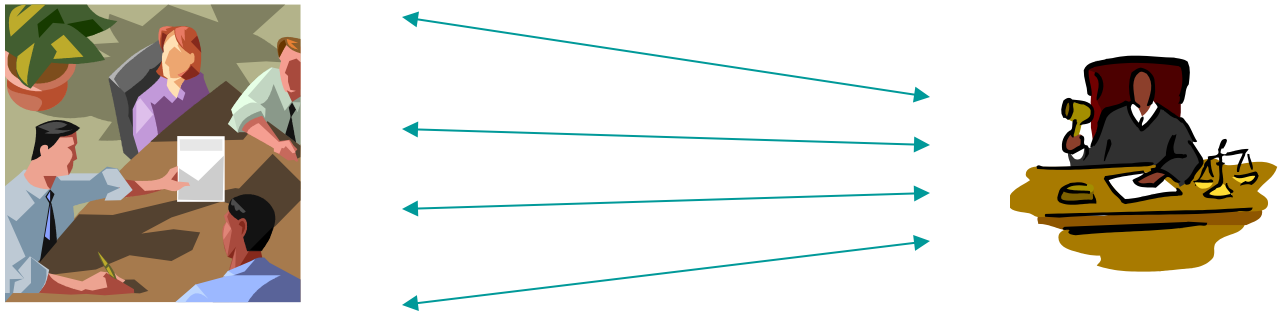
- In the classical case, Dwork and Stockmeyer (1992) introduced the following notation.

- ZKP(⟨restrictions⟩) = the class of all languages that have IP systems satisfying the recognition zero-knowledge property for a verifier and a prover with restrictions given in ⟨restrictions⟩.

- For example:
  - ZKP(2pfa,poly-time) = the class of all languages that have ZKP systems with 2pfa verifiers running in expected polynomial time.

Quantum Zero-Knowledge Proofs

- Earlier, Watrous (2002) and Kobayashi (2003) studied polynomial-time quantum zero-knowledge proof (QZKP) systems.
- Here, we discuss a quantum analogue of automata-based ZKP(2pfa,poly-time) with multiple provers.



multiple provers

a verifier

QMZKP Notation

- QMZKP(⟨restrictions⟩) = the class of all languages that have QZKP (⟨restrictions⟩) systems with the restrictions given in ⟨restrictions⟩.

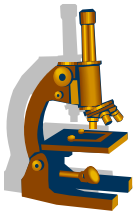
- For example:
  - QMZKP(2qfa,poly-time) = the language class by QMIP(2qfa,poly-time) systems that satisfy the recognition zero-knowledge property.
  - QMZKP(2qfa,poly-shared,poly-time) = the language class by QMIP(2qfa,poly-shared,poly-time) systems with polynomially-many shared entangled qubits

QMIP vs. QMZKP

- Yamakami (2007) obtained the following statement.

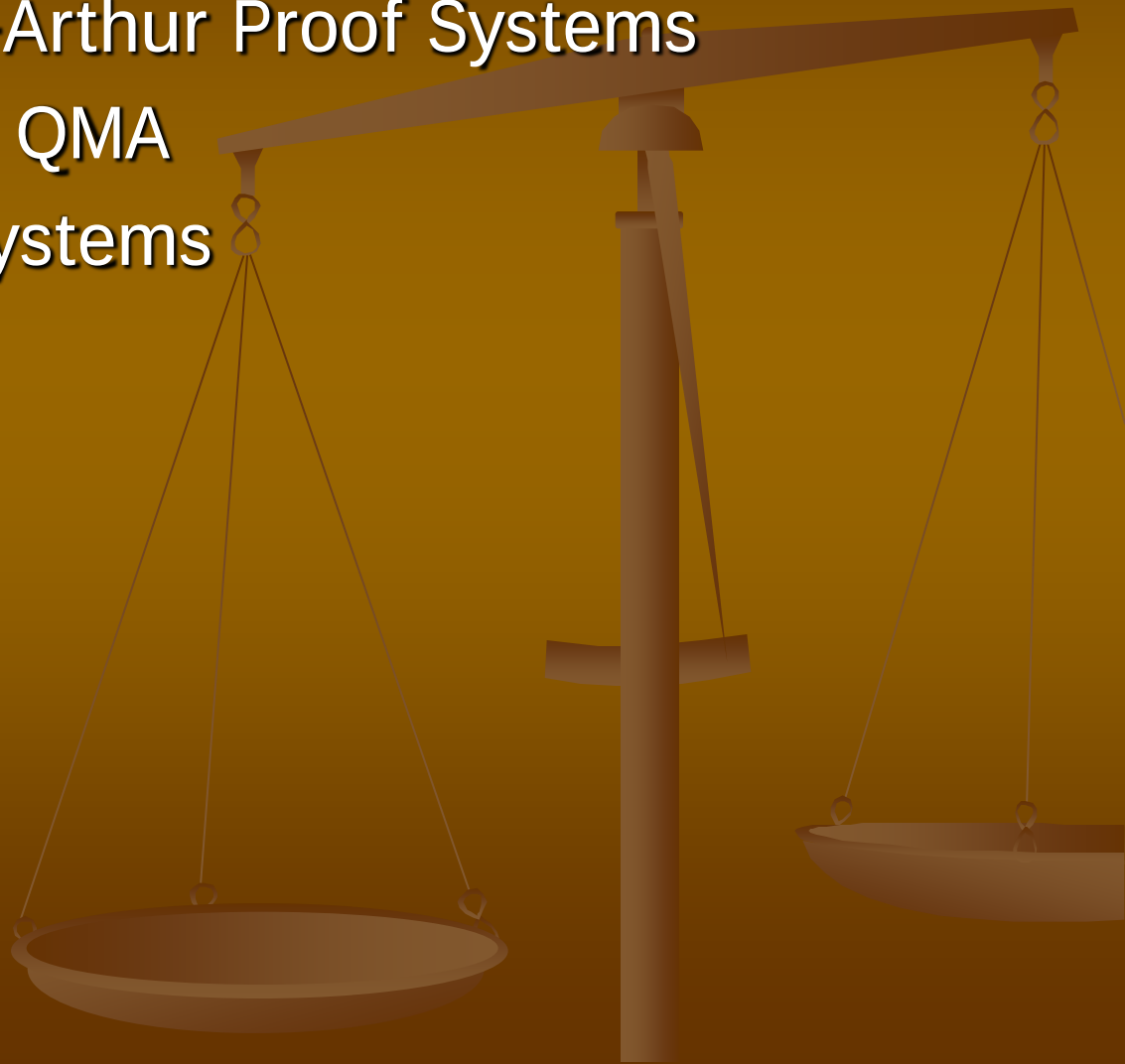
- Theorem: [Yamakami (2007)]

  QMIP(2qfa,poly-shared,poly-time) $\subseteq$ QMZKP(2qfa,poly-shared,poly-time).



IV. Polynomial-Time QMA Proof Systems

1. Quantum Merlin-Arthur Proof Systems
2. Polynomial-Time QMA
3. Power of QMA Systems



Quantum Merlin-Arthur Proof Systems

- Kobayashi, Matsumoto, and Yamakami (2009) studied polynomial-time QMA proof systems.
- Here, we take quantum Turing machines (QTMs) as verifiers.

- A language L has a (k,c,s)-quantum Merlin-Arthur proof (QMA) system (P,V) $\Leftrightarrow$ there exists a polynomial-time quantum verifier (i.e., QTM) V s.t., for every input x,
  - $x \in L \rightarrow$ there exists a set of k(|x|) quantum proofs that makes V accept x with prob $\geq$ c(|x|), and
  - $x \notin L \rightarrow$ for any set of k(|x|) quantum proofs, V accept x with prob $\leq$ s(|x|).

Polynomial-Time QMA

- A language L is in QMA(k,c,s) $\Leftrightarrow$ L has a (k,c,s)-quantum Merlin-Arthur proof (QMA) system

- Theorem: [Kobayashi-Matsumoto-Yamakami (2009)]

  The following statements are equivalent.

  1. QMA(k,c,s) = QMA(2,2/3,1/3) for any k $\geq$2 and any two-sided bounded-error completeness-soundness pair (c,s).
  2. QMA(2,c,s) = QMA(2,2/3,1/3) for any two-sided bounded-error completeness-soundness pair (c,s).

Power of QMA Systems

- Here, we present two results obtained by Kobayashi, Matsumoto, and Yamakami (2009).

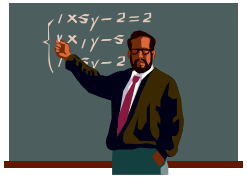
- Theorem: [Kobayashi-Matsumoto-Yamakami (2009)]

  For any p-bounded function k and any function c : $N^+ \to (0,1]$, it follows that QMA(k,c,0) = QMA(1,c,0).

- Theorem: [Kobayashi-Matsumoto-Yamakami (2009)]

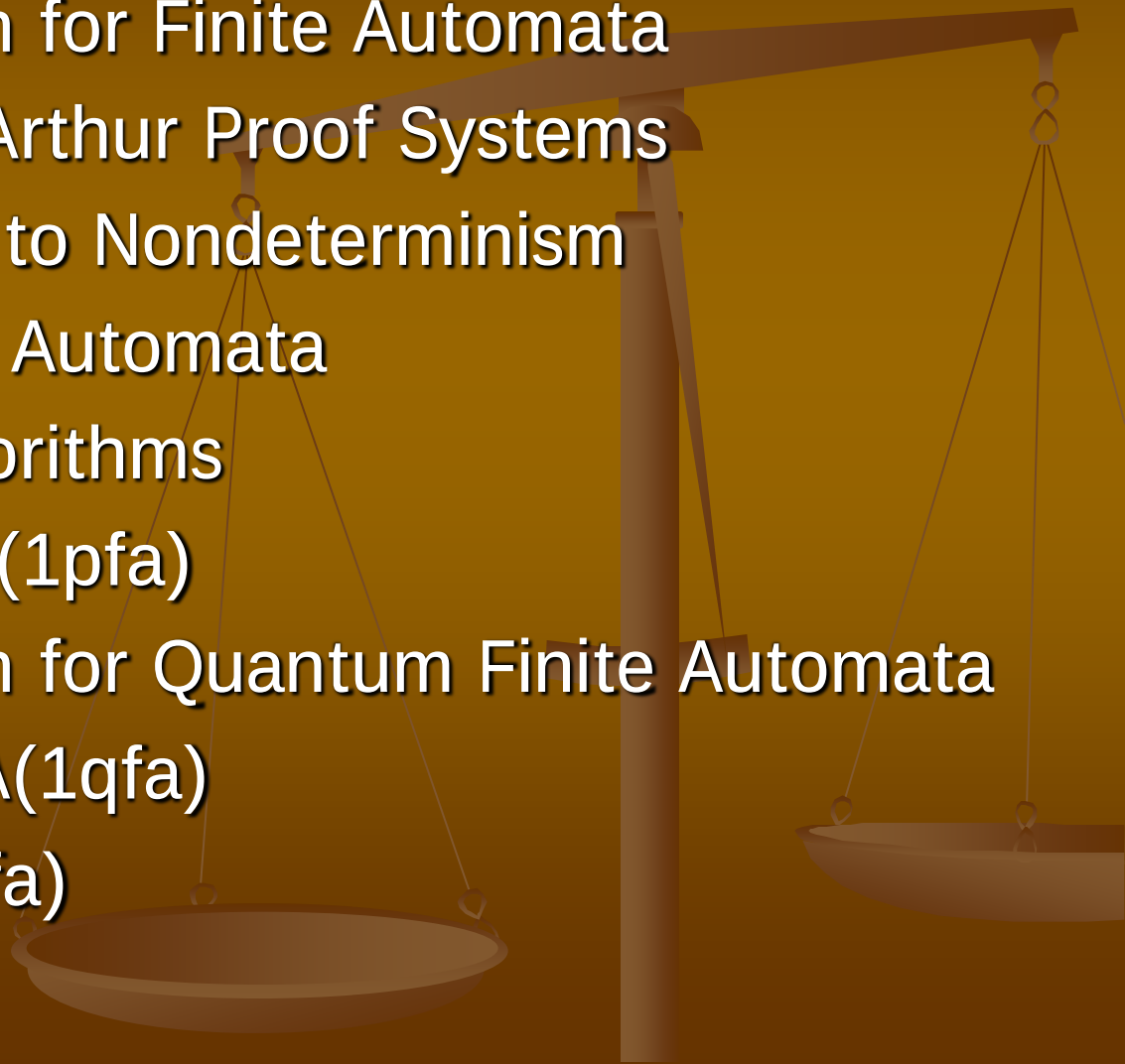
  NQP = U_c QMA(1,c,0),

  where c : $N^+ \to (0,1]$ be any function.



V. Proof-Verification Processes

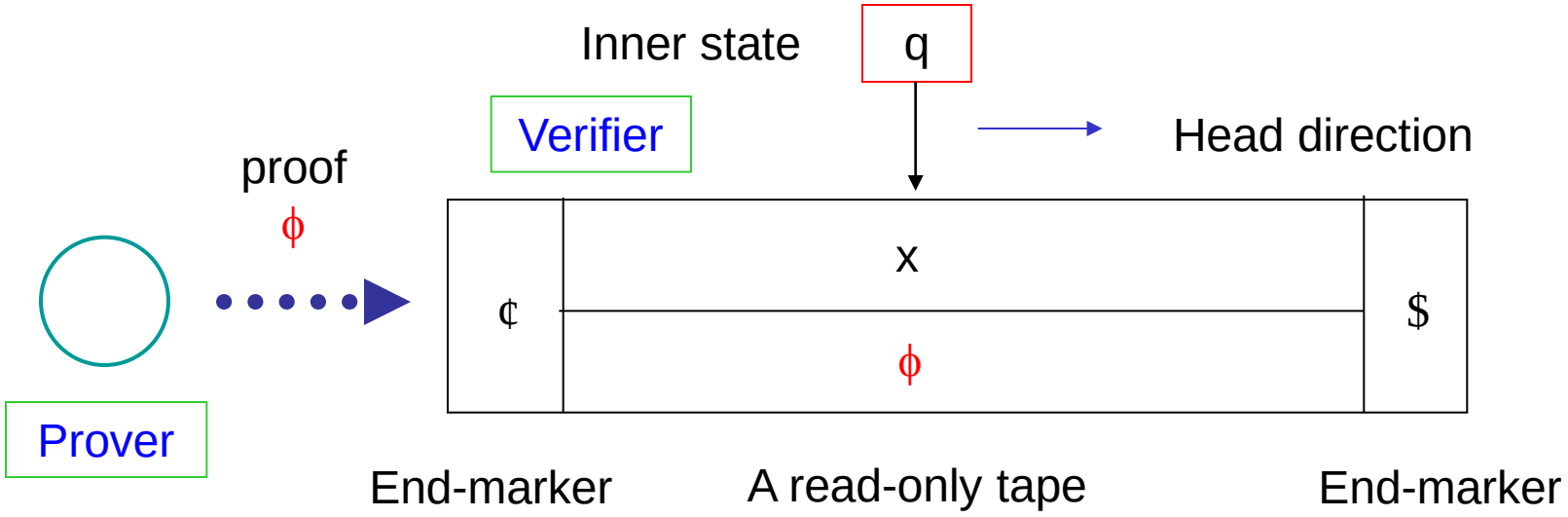
1. Proof-Verification for Finite Automata
2. Classical Merlin-Arthur Proof Systems
3. Correspondence to Nondeterminism
4. Reversible Finite Automata
5. Randomized Algorithms
6. properties of MA(1pfa)
7. Proof-Verification for Quantum Finite Automata
8. MA(1rfa) and MA(1qfa)
9. 1QFA vs. MA(1qfa)



Proof-Verification for Finite Automata

- Verifier is now uses one-way finite automata.

- Merlin-Arthur Proof-Verification Condition:
  - $x \in L \rightarrow \exists\, \phi$:proof [V accepts $[x,\phi]^T$]
  - $x \notin L \rightarrow \forall\, \phi$:proof [V rejects $[x,\phi]^T$]



Classical Merlin-Arthur Proof Systems

- Our proof-verification systems are called Merlin-Arthur games.

- Notationally, we write:

  MA(fa) = languages L s.t. there is a Merlin-Arthur proof system with fa for L

- If we use determinsitic advice, we write

  MA(fa)/n = languages L s.t. there is a Merlin-Arthur proof system with fa with advice for L

Correspondence to Nondeterminism

- Proof-verification process naturally corresponds to "nondeterminism."
- For example, we can prove that MA(1dfa) = 1NFA.
- However, it is known that 1DFA = 1NFA = REG.
- Therefore, we obtain the following conclusion.
- (Claim) MA(1dfa) = REG.
- Similarly, by adding advice, we can show that
  - MA(1dfa)/n = 1NFA/n.
  - 1DFA/n = 1NFA/n = REG/n.

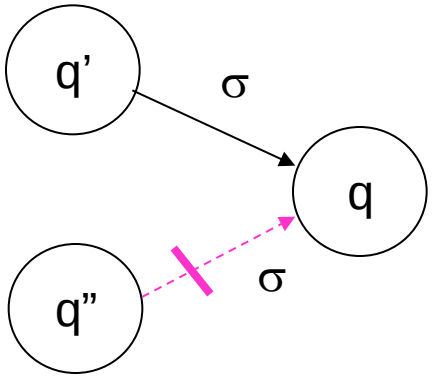
Reversible Finite Automata (revisited)

A one-way (deterministic) reversible finite automaton (or 1rfa) is defined as follows:

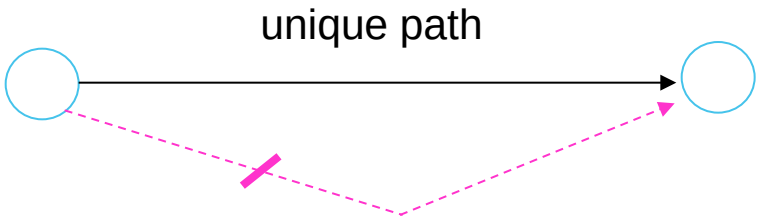
$M = (Q,\Sigma,\delta,q_0,Q_{acc},Q_{rej})$ Σ = input alphabet

Reversibility condition:

$\forall q \in Q\ \forall \sigma \in \Sigma\ \exists$ at most one $q' \in Q$ s.t. $\delta(q',\sigma) = q$.



Property: If there is a computation path from q_0 to $q \in Q_{acc}$ (or Q_{rej}), such a path should be unique.



Nondeterminism for Reversible Finite Automata

- Proof-verification procedures could define non-determinism.
- Hence, MA(1rfa) could define one-way nondeterministic reversible finite automata (or 1nrfa's)

- A language L is in MA(1rfa) $\Leftrightarrow$

  $\exists M = (Q,\Sigma,\delta,q_0,Q_{acc},Q_{rej})$: 1rfa s.t., for any x,

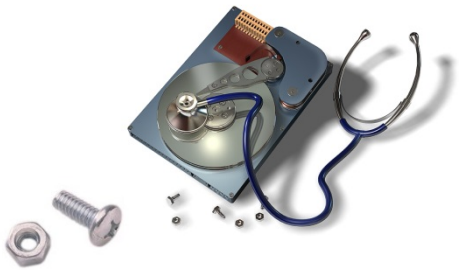
  1. $x \in L \rightarrow \exists\, s \in \Sigma^{|x|}$ [M accepts $[x,s]^T$]
  2. $x \notin L \rightarrow \forall s \in \Sigma^{|x|}$ [M rejects $[x,s]^T$]

Determinism vs. Nondeterminism

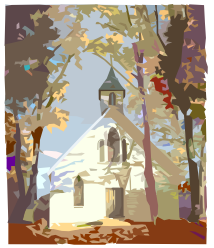
- By definitions, 1RFA $\subseteq$ MA(1rfa).

- However, it is not clear if “determinism” and “nondeterminism” coincide for reversible finite automata.
- In other words,

  1RFA = MA(1rfa) ?
- This is because a standard proof for “1DFA = 1NFA” does not seem to work for reversible finite automata.



Randomized Algorithms



- A one-way probabilistic finite automata (or 1pfa) is a randomized version of a 1dfa.
- Recall the bounded-error requirement:

  There exists a constant ε with $0 \le \varepsilon < 1/2$ s.t.
  - $x \in L \rightarrow$ M accepts x with probability $\ge 1-\varepsilon$;
  - $x \notin L \rightarrow$ M rejects x with probability $\ge 1-\varepsilon$.

- $L \in$ MA(1pfa) $\Leftrightarrow \exists M$:1pfa $\exists \varepsilon \in [0,1/2)$ s.t., for any x,
  - $x \in L \rightarrow \exists \phi$: proof [M accepts x with prob. $\ge 1-\varepsilon$];
  - $x \notin L \rightarrow \forall \phi$: proof [M rejects x with prob. $\ge 1-\varepsilon$].

Properties of MA(1pfa)

- MA(1pfa) corresponds to the notion of finite automata with nondeterminsitic and probabilistic moves.

- (Claim) MA(1pfa) = REG. [Condon et al. (1997)]

- Next, we will expand this MA(1pfa) to a new class MA(1qfa) defined with 1-way quantum finite automata.

Proof-Verification for Quantum Finite Automata

- Let us consider proof-verification process for 1qfa's.
- We define MA(1qfa) as follows.

- A language L is in MA(1qfa) $\Leftrightarrow$

  $\exists$M:1qfa $\exists\varepsilon\in[0,1/2)$ s.t., for any x,

  - $x\in L \rightarrow \exists|\phi\rangle$: proof [M accepts x with prob. $\geq 1-\varepsilon$];
  - $x\notin L \rightarrow \forall|\phi\rangle$: proof [M rejects x with prob. $\geq 1-\varepsilon$],

  where $|\phi\rangle$ is a quantum state over basis states $\Sigma^{|x|}$.

MA(1rfa) and MA(1qfa)

- Lemma: [Villagra-Yamakami (2015)]

  MA(1rfa) $\neq$ 1QFA.

- In other words, “nondeterminism” is not powerful enough for 1rfa’s to simulate 1qfa’s.

- Recall from Week 3 that 1QFA/n $\neq$ REG/n.
- We can obtain the following separation result.
- Lemma: [Villagra-Yamakami (2015)]

  MA(1qfa)/n $\neq$ REG/n.

1QFA vs. MA(1qfa)

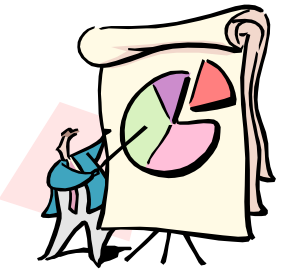
- It is clear that 1QFA $\subseteq$ MA(1qfa).

- Lemma: [Villagra-Yamakami (2015)]
  MA(1qfa) $\subseteq$ REG.

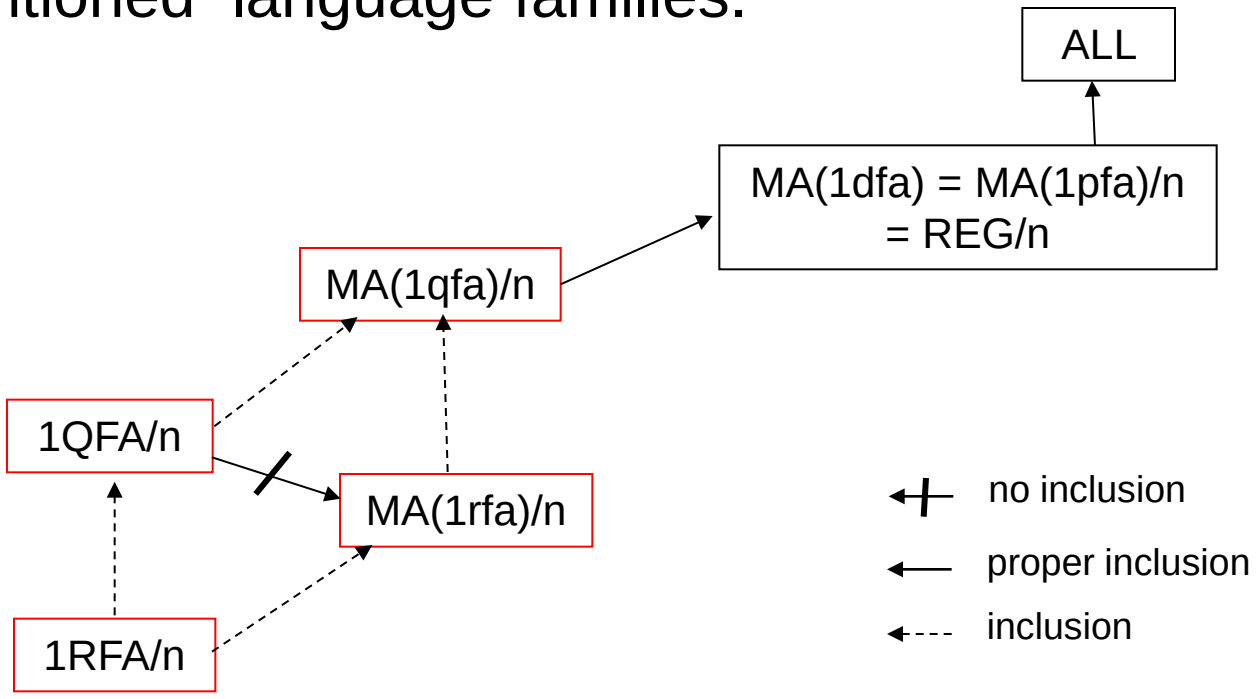
❑ Proof Sketch:

- By applying the result QIP(1qfa) = REG [Nishimura-Yamakami (2009)], we obtain MA(1qfa) $\subseteq$ REG.

Class Separations



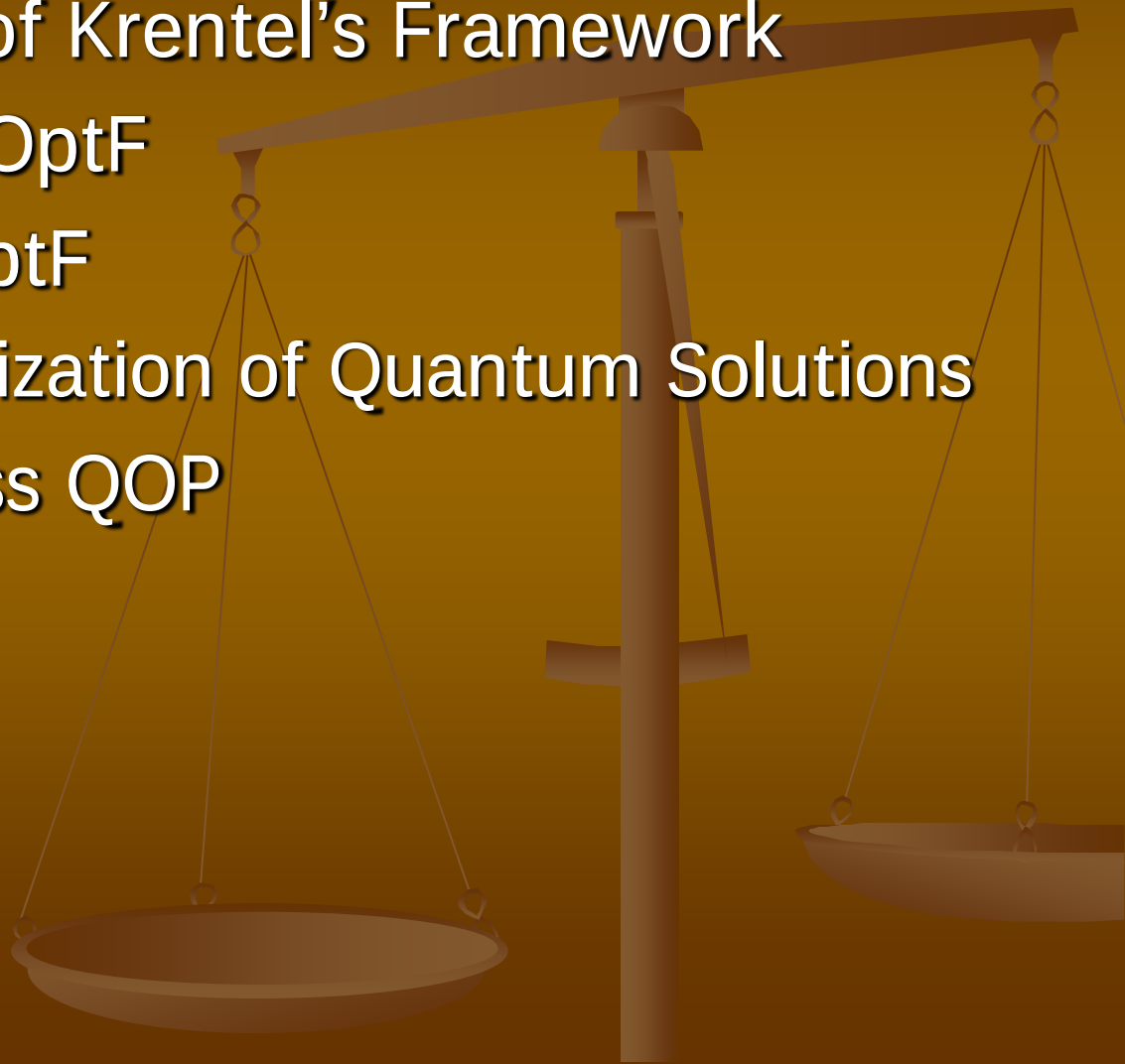
- Villagra and Yamakami (2015) showed the following class separations among the aforementioned language families.



- (Open Problem) Determine whether each inclusion is proper.

VI. Quantum Optimization Problems

1. Generalization of Krentel's Framework
2. Function Class OptF
3. Properties of OptF
4. Quantum Optimization of Quantum Solutions
5. Complexity Class QOP
6. Relationships



Generalization of Krentel's Framework

- Optimization problems have been important problems in theory and in practice.
- Krentel (1988) defined OptP, which is an optimization class consisting of optimal cost functions for NP optimization problems.
- Given a function class F, we want to introduce a general notion of OptF.

Function Class OptF I

- Here, we consider only partial functions from $\Sigma^*\times\Sigma^*$to R.
- Let F be any set of such partial functions.
- Let F be any set of such partial functions.

- A partial function f : $\Sigma^*\times\Sigma^* \to$ R is in OptF $\Leftrightarrow$ there are a polynomial p and a maximization problem $\Pi=(\Sigma^*,sol_\Pi,g)$ with Σ^* as the set of instances and g as a partial cost function chosen from F satisfying that
  1. $sol_\Pi(x) \subseteq \Sigma^{p(|x|)}$ for all x, and
  2. f is the maximal cost function for Π, namely,

     $f(x) = \max\{ g(x,s) \mid (x,s) \in dom(g), s \in \Sigma^{p(|x|)} \}$

     if g(x,s) exists.

Function Class OptF II

- By taking

  FP, FPSPACE, #P, #QP, and FBQP as F,

  we obtain

  OptFP, OptFPSPACE, Opt#P, Opt#QP, and OptFBQP.

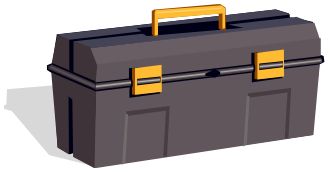
- Note that OptFP coincides with Krentel's OptP.

- (Claim) OptFPSPACE = FPSPACE

- (Open Problem) Is OptFP = FP?

Examples: MAXQTM

- We see a simple example of elements of Opt#QP.
- Fix a universal QTM, say, U.
- Maximum QTM Problem (MAXQTM)
  - instance: $\langle M,x,1^t,1^m\rangle$ with a QTM M, $t\in N$, and $m\in N^+$
  - solution: maximal acceptance probability over all strings $s\in\Sigma^{|x|}$ of U on input $\langle M,xs,1^t,1^m\rangle$
- (Claim) MAXQTM $\in$ Opt#QP



Properties of OptF I

- Yamakami (2002) showed the following properties of OptF.

- Lemma: $\mathrm{OptP} \cup \#\mathrm{P} \subseteq \mathrm{Opt}\#\mathrm{P} \subseteq \mathrm{FPSPACE}$
- Lemma: $FBQP \subseteq OptFBQP \subseteq FP^{\Sigma_2^p(BQP)}$
- Proposition: $\#QP \subseteq Opt\,\#QP \subseteq \#QP^{NP^{PP}}$
- Theorem:

  $\mathrm{NP}^{\mathrm{PP}} = \mathrm{EQP} \Rightarrow \mathrm{Opt}\#\mathrm{QP} = \#\mathrm{QP} \Rightarrow \mathrm{NP}^{\mathrm{PP}} = \mathrm{WQP}.$

Properties of OptF II

- Lemma: [Yamakami (2002)]
  OptFBQP = FBQP $\Leftrightarrow$ $NP^{BQP} = BQP$.

❑ Proof Sketch:

- If $NP^{BQP} = BQP$, then $\Sigma_2^{p(BQP)} = BQP$.
- Thus, OptFBQP $\subseteq$ $FP^{\Sigma p2(BQP)} \subseteq FP^{BQP} = FBQP$.

QED

Quantum Optimization of Quantum Solutions

- Let F be a set of partial functions from $\Sigma^* \times \Phi_\infty$ to R.
- A partial function f : $\Sigma^* \to$ R is in QoptF $\Leftrightarrow$ there are a polynomial p and a maximization problem Π with Σ^* and g as a partial cost function in F s.t., for any instance $x \in \Sigma^*$,

$$f(x) = \sup\{ g(x,|\phi\rangle) \mid (x,|\phi\rangle) \in \mathrm{dom}(g),\ |\phi\rangle \in \Phi_{p(|x|)} \}$$

  if such an $g(x,|\phi\rangle)$ exists. Otherwise, f(x) is undefined.

- A set of quantum solutions of x for Π is

$$\mathrm{sol}_g(x) = \{|\phi\rangle \in \Phi_{p(|x|)} \mid (x,|\phi\rangle) \in \mathrm{dom}(g) \}.$$

Complexity Class QOP

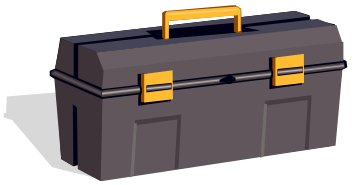
- A language L is in QOP (quantum optimization polynomial time) $\Leftrightarrow$ there exist two functions $f,g \in \#QP$ and a function $h \in FP$ (called a selection function) s.t., for any x,

$$x \in L \leftrightarrow \lfloor 2^{|h(x)|} f(x) \rfloor > \lfloor 2^{|h(x)|} g(x) \rfloor.$$

- Moreover, we define

$$QOP\hat{} = \{ L \in QOP \mid \forall x \ [\lfloor 2^{|h(x)|} f(x) \rfloor > \lfloor 2^{|h(x)|} g(x) \rfloor] \}.$$

- (Claim) $QOP\hat{} \subseteq QOP$



Relationships

- We use the notion $\ll^e$ defined by:

  $F \ll^e G \Leftrightarrow \forall f \in F\ \forall p$:polynomial $\exists g \in F$ s.t.

  $\forall x\ |f(x) - g(x)| \leq 2^{-p(|x|)}$.

- Yamakami (2002) showed the following relationships.
- Theorem:
  1. $EQP = QOP \rightarrow Opt\#QP \ll^e \#QP$.
  2. $Qopt\#QP \ll^e \#QP \rightarrow QOP^{\wedge} = PP$.
  3. $Opt\#QP \subseteq Qopt\#QP \ll^e \#QP QOP \ll^e FPSPACE$.

Thank you for listening

Q & A

I’m happy to take your question!

END