Quantum Merlin-Arthur (QMA) Verification

Naomi Jiang

1 Introduction

Interactive proof systems and Arthur-Merlin games model the notion of computationally efficient verification. These systems involve a polynomial-time verifier interacting with a computationally unbounded prover to validate claims about input strings being part of a specific language.

1.1 Interactive Proof Systems

In an interactive proof system, a verifier with access to private random bits interacts with a prover to verify claims. The interaction typically involves a series of messages exchanged between the verifier and the prover, aiming to ascertain the membership of an input string in a predetermined language.

1.2 Arthur-Merlin Games

The Arthur-Merlin game is a variant of the interactive proof system. It restricts the verifier (Arthur) to using a public source of randomness that is also visible to the prover (Merlin). Unlike traditional interactive proofs, where the verifier's random bits are private, in Arthur-Merlin games the random source is public, which affects the interaction dynamics but not the computational performance.

2 Complexity Classes

Arthur-Merlin games and interactive proof systems, though structurally different, are computationally equivalent in terms of the complexity classes they generate. This equivalence and the definitions of several important complexity classes are discussed below.

2.1 AM and MA

- **AM (Arthur-Merlin)**: This class is defined by games in which exactly two messages are exchanged—the first from Arthur to Merlin, followed by a response from Merlin back to Arthur. It is equivalent to any language that has an interactive proof system with a constant number of message exchanges.

- **MA (Merlin-Arthur)**: A subset of AM, where the game involves only one effective message being sent from the prover (Merlin) to the verifier (Arthur). Typically, this class is seen as a probabilistic variant of NP, where the verifier uses randomness in the decision process. MA can be described using two-message Arthur-Merlin games where the second message is not counted as it does not convey new information but merely uses the random source.

3 QMA Verification Procedure

Definition 1 (QMA Verification Procedure)**.** *A QMA verification procedure A is defined as a family of quantum circuits $\{A_x : x \in \Sigma^*\}$ that is generated in polynomial time. Associated with this is a function $m \in poly$, which specifies the length of Merlin's message to Arthur. It is assumed that each circuit A_x acts on $m(|x|) + k(|x|)$ qubits where k is a function indicating the number of work qubits used by the circuit.*

3.1 Simplification of Notation

In practice, when the input x is fixed or implicit in discussion:

- write m to mean $m(|x|)$,
- write k to mean $k(|x|)$.

We refer to A as an m-qubit QMA verification procedure when focusing on the length of Merlin's message.

3.2 Complexity Classes $QMA(a, b)$ and $QMA_m(a, b)$

Definition 2 (Class $QMA(a, b)$)**.** *The class $QMA(a, b)$ consists of all languages $L \subseteq \Sigma^*$ for which there exists a QMA verification procedure A for which the following holds:*

1. *For all $x \in L$ there exists an m-qubit quantum state $|\psi\rangle$ such that $\Pr[A_x \text{ accepts } |\psi\rangle] \geq a$.*
2. *For all $x \notin L$ and all m-qubit quantum states $|\psi\rangle$, $\Pr[A_x \text{ accepts } |\psi\rangle] \leq b$.*

Definition 3 (Class $QMA_m(a, b)$)**.** *For any $m \in poly$, the class $QMA_m(a, b)$ consists of all languages $L \subseteq \Sigma^*$ for which there exists an m-qubit QMA verification procedure that satisfies the above properties. One may consider the cases where a and b are constants or functions of the input length $n = |x|$. If a and b are functions of the input length, it is assumed that $a(n)$ and $b(n)$ can be computed deterministically in time polynomial in n. When no reference is made to the probabilities a and b, it is assumed $a = \frac{2}{3}$ and $b = \frac{1}{3}$.*

3.3 Strong Error Reduction in QMA

3.3.1 Kitaev's Error Reduction

Let $a, b : \mathbb{N} \rightarrow [0, 1]$ and $q \in$ poly satisfy the condition that $a(n) - b(n) \geq \frac{1}{q(n)}$ for all $n \in \mathbb{N}$. Then, $QMA(a, b) \subseteq QMA(1 - 2^{-r}, 2^{-r})$ for every $r \in$ poly.

The proof's central idea is as follows:

Consider a quantum verification procedure A with two key probabilities:

- Completeness probability (a): Probability that A accepts if the input is in the language.
- Soundness probability (b): Probability that A accepts if the input is not in the language.

Procedure Construction:

1. *Multiple Independent Runs:* Run A several times independently using copies of the potential solution (quantum state).
2. *Acceptance Counting:* Count how many times A accepts the solution across these runs.
3. *Threshold Comparison:* Compare this count to a threshold set at $\frac{a+b}{2}$. If the count exceeds this threshold, the new procedure accepts the solution.

Key Challenges and Insights:

- The new verification may involve complex quantum states (possibly entangled). However, entanglement does not increase the cheating potential significantly.
- If the input is not in the language ($x \notin L$), the probability of a single run accepting the solution is at most b, regardless of entanglement.
- This strategy improves reliability by using the law of large numbers, essentially averaging out the fluctuations in individual test outcomes.

This construction requires an increase in the length of Merlin's message to Arthur in order to reduce error

3.3.2 Error Reduction in QMA with Fixed Message Length

The main result of this section is the following theorem, which states that one may decrease error without any increase in the length of Merlin's message.

Statement: Let $a, b : \mathbb{N} \rightarrow [0, 1]$ and $q \in$ poly satisfy

$$a(n) - b(n) \geq \frac{1}{q(n)}$$

for all $n \in \mathbb{N}$. Then $QMA_m(a, b) \subseteq QMA_m(1 - 2^{-r}, 2^{-r})$ for every $m, r \in$ poly.

Proof: Assume $L \in QMA_m(a, b)$, and let A be an m-qubit QMA verification procedure that witnesses this membership. We construct a new m-qubit QMA verification procedure B that aims to drastically reduce the completeness and soundness error for the language L.

Procedure Construction: The key to B's construction is to simulate the original procedure A repeatedly while managing a carefully tuned threshold for acceptance based on the aggregated outcomes. Here are the detailed steps and logic:

1. The input x is fixed (for simplification), and we consider A and B as A_x and B_x, respectively.
2. We define a quantum register R that combines m message qubits and k workspace qubits from A, making an $m + k$ qubit register.
3. We define projection operators $\Pi_1 = |1\rangle\langle 1| \otimes I_{m+k-1}$, $\Delta_1 = I_m \otimes |0^k\rangle\langle 0^k|$, and similar for Π_0 and Δ_0. These are used to measure whether the first qubit is in the state $|1\rangle$ or $|0\rangle$ and whether the last k qubits are all zeros, respectively.

Operational Logic of B:

1. **Initialization:** Assume the initial state where the first m qubits contain Merlin's message $|\psi\rangle$ and the remaining k qubits are set to $|0^k\rangle$.
2. **Iteration Setup:** Set $y_0 \leftarrow 1$ and $i \leftarrow 1$.
3. **Repeat Process:**
   a. *Apply A and Measure:* Apply the verification procedure A to the register R, and then measure R using the projections $\{\Pi_0, \Pi_1\}$. Let y_i denote the outcome of this measurement, and then increment i by 1.
   b. *Apply $A^\dagger$ and Measure:* Apply $A^\dagger$ (the inverse of A) to R and measure R using the projections $\{\Delta_0, \Delta_1\}$. Update y_i with the outcome and increment i by 1.
4. **Termination Condition:** Continue the above steps until $i \geq N$, where $N = 8q^2 r$.
5. **Decision Rule:**
   a. For each $i = 1, \dots, N$, set z_i as follows:
$$z_i = \begin{cases} 1 & \text{if } y_i = y_{i-1} \\ 0 & \text{if } y_i \neq y_{i-1} \end{cases}$$
   b. *Final Acceptance:* The procedure B accepts if the sum of z_i from 1 to N is at least $N \cdot \frac{a+b}{2}$, otherwise, it rejects.

3.4 Applications of strong error reduction

3.4.1 QMA $\subseteq$ PP

Proof: Let $L \subseteq \Sigma^*$ be a language in QMA. According to the strong error reduction, there exists a polynomial function m such that

$$L \in QMA_m\left(1 - 2^{-(m+2)}, 2^{-(m+2)}\right).$$

Let A be a verification procedure that witnesses this fact. Specifically, each circuit A_x acts on $k+m$ qubits, for some k in polynomial time, and satisfies the following:

- If $x \in L$, then there exists an m-qubit state $|\psi\rangle$ such that
$$\Pr[A_x \text{ accepts } |\psi\rangle] \geq 1 - 2^{-m-2},$$
- If $x \notin L$, then
$$\Pr[A_x \text{ accepts } |\psi\rangle] \leq 2^{-m-2}$$
for every m-qubit state $|\psi\rangle$.

For each $x \in \Sigma^*$, define a $2^m \times 2^m$ matrix Q_x as

$$Q_x = \left(I_m \otimes \langle 0^k|\right) A_x^\dagger \Pi_1 A_x \left(I_m \otimes |0^k\rangle\right).$$

Each Q_x is positive semidefinite and $\langle\psi|Q_x|\psi\rangle = \Pr[A_x \text{ accepts } |\psi\rangle]$ for any unit vector $|\psi\rangle$ on m qubits. The maximum probability with which A_x can be made to accept is the largest eigenvalue of Q_x. Since the trace of a matrix is equal to the sum of its eigenvalues and all eigenvalues of Q_x are nonnegative, it follows that:

- If $x \in L$, then $\mathrm{tr}(Q_x) \geq 1 - 2^{-m-2} \geq \frac{3}{4}$,
- If $x \notin L$, then $\mathrm{tr}(Q_x) \leq 2^m 2^{-m-2} \leq \frac{1}{4}$.

Using a modification of the method of [FR99], we define polynomially-bounded functions g and GapP functions f_1 and f_2 such that:

$$\Re(Q_x[i,j]) = \frac{f_1(x,i,j)}{2^g(x)}, \quad \Im(Q_x[i,j]) = \frac{f_2(x,i,j)}{2^g(x)},$$

for $0 \leq i, j < 2^m$. Define

$$h(x) = \sum_{i=0}^{2^m-1} f_1(x,i,i).$$

Since GapP functions are closed under exponential sums, $h \in$ GapP and

$$h(x) = \frac{2}{2^g(x)}\,\mathrm{tr}(Q_x),$$

therefore,

$$x \in L \Rightarrow h(x) \geq \frac{3}{4} 2^{g(x)}, \quad x \notin L \Rightarrow h(x) \leq \frac{1}{4} 2^{g(x)}.$$

Because $2^{g(x)}$ is an FP function, it follows that $2h(x) - 2^{g(x)}$ is a GapP function that is positive if $x \in L$ and negative if $x \notin L$. Thus, $L \in PP$ as required.

3.4.2 Remark 3.5: Extension to A0PP

A simple modification of the proof for Theorem 3.4 shows that QMA is also contained within A0PP. Specifically, the GapP function $2h(x)$ and the FP function $2^{g(x)}$ meet the necessary criteria for A0PP:

- If $x \in L$, then $2h(x) \geq 2^{g(x)}$,
- If $x \notin L$, then $2h(x) \leq \frac{1}{2} 2^{g(x)}$.

3.4.3 $QMA_{\log} = BQP$

Proof: The inclusion $BQP \subseteq QMA_{\log}$ is straightforward, hence we focus on proving $QMA_{\log} \subseteq BQP$. Assume a language L belongs to QMA_m for m logarithmic in size, and let A be the QMA verification procedure with completeness and soundness errors less than $2^{-(m+2)}$. Define the matrix Q_x as:

$$Q_x = \left(I_m \otimes \langle 0^k|\right) A_x^\dagger \Pi_1 A_x \left(I_m \otimes |0^k\rangle\right).$$

From the prior proofs, we know:

$$x \in L \Rightarrow \operatorname{tr}(Q_x) \geq \frac{3}{4},$$
$$x \notin L \Rightarrow \operatorname{tr}(Q_x) \leq \frac{1}{4}.$$

We introduce a polynomial-time quantum algorithm B that decides L with bounded error. This algorithm constructs a totally mixed state over m qubits and utilizes the verification procedure A on this state instead of a specific message from Merlin. The use of the totally mixed state simulates running A on uniformly chosen standard basis states, achievable via Hadamard transforms and reversible computation.

The density matrix for the totally mixed state on m qubits is $\frac{1}{2^m} I_m$, leading to:

$$\Pr[B \text{ accepts } x] = \operatorname{tr}\left(\frac{Q_x}{2^m} I_m\right) = \frac{1}{2^m} \operatorname{tr}(Q_x).$$

Given the logarithmic nature of m relative to the size of x, the probabilities for B accepting $x \in L$ and $x \notin L$ are sufficiently separated to allow error amplification by standard techniques, concluding that $L \in BQP$.

4 Quantum Arthur-Merlin (QAM) Complexity Class

The Quantum Arthur-Merlin (QAM) model extends the Quantum Merlin-Arthur (QMA) verification framework by integrating a sequence of interactions involving random bits, providing a more dynamic approach to quantum verification.

4.0.1 QAM Verification Procedure

A QAM verification procedure comprises:

- A polynomial-time generated family of quantum circuits $\{A_{x,y} : x \in \Sigma^*, y \in \Sigma^{s(|x|)}\}$, where:
  - x represents the input string.
  - y represents a sequence of coin-flips generated by Arthur.
  - $s \in \text{poly}$ dictates the length of y.
- Each circuit $A_{x,y}$ operates on two sets of qubits:
  - $m(|x|)$ qubits received from Merlin.
  - $k(|x|)$ work qubits used by Arthur.
- The acceptance of a message $|\psi_y\rangle$ by circuit $A_{x,y}$ mirrors the QMA acceptance condition.

4.0.2 QAM Complexity Class Definition

Definition 4.1: The class QAM(a, b) consists of all languages $L \subseteq \Sigma^*$ satisfying:

1. **Completeness:** For all $x \in L$, there exists a set of states $\{|\psi_y\rangle\}$ on m qubits such that the average acceptance probability across all y is at least a:
$$\frac{1}{2^s} \sum_{y \in \Sigma^s} \Pr[A_{x,y} \text{ accepts } |\psi_y\rangle] \geq a.$$
2. **Soundness:** For all $x \notin L$ and any set of states $\{|\psi_y\rangle\}$, the average acceptance probability is at most b:
$$\frac{1}{2^s} \sum_{y \in \Sigma^s} \Pr[A_{x,y} \text{ accepts } |\psi_y\rangle] \leq b.$$

Parameters:

- a and b can be constants or functions of the input length $n = |x|$, with polynomial-time computability.
- The default values are typically $a = 2/3$ and $b = 1/3$ for standard configurations.

5 Quantum Merlin-Arthur-Merlin (QMAM)

The Quantum Merlin-Arthur-Merlin (QMAM) model extends the QMA framework by incorporating an additional message from Merlin to Arthur, which allows for more dynamic and potentially more powerful quantum verification processes.

5.0.1 QMAM Verification Procedure

The QMAM verification involves:

- A polynomial-time generated family of quantum circuits $\{A_{x,y} : x \in \Sigma^*, y \in \Sigma^{s(|x|)}\}$, reflecting interactions based on Arthur's random bits (y).
- Each circuit $A_{x,y}$ interacts with qubits divided as follows:
  - $m_1(|x|)$ qubits for Merlin's first message.
  - $m_2(|x|)$ qubits for Merlin's second message after processing the coin-flips.
  - $k(|x|)$ work qubits in Arthur's workspace.
- Merlin's strategy can involve complex actions such as transforming parts of the quantum state based on the received coin-flips to generate the second message.

5.0.2 Definition of the QMAM Class

Definition 5.1: A language $L \subseteq \Sigma^*$ belongs to QMAM(a, b) if:

1. **Completeness:** There exists a quantum state $|\psi\rangle$ on $m_1 + m_2 + l$ qubits, and a set of unitary operations $\{U_y\}$ such that:

$$\frac{1}{2^s} \sum_{y \in \Sigma^s} \Pr[A_{x,y} \text{ accepts } (I_{m_1} \otimes U_y)|\psi\rangle] \geq a.$$

2. **Soundness:** For all states $|\psi\rangle$ and unitary operations $\{U_y\}$, the acceptance probability is bounded by b:

$$\frac{1}{2^s} \sum_{y \in \Sigma^s} \Pr[A_{x,y} \text{ accepts } (I_{m_1} \otimes U_y)|\psi\rangle] \leq b.$$