MATH 3QC3 - Assignment 4 Solutions

Matthew Yu — 400322243 — Yum77

April 15, 2025

This set of problems will guide you through a simplified proof that quantum circuits using only certain gates ("Clifford" gates) can be simulated efficiently on a classical computer. No group theory background is required; all necessary definitions are introduced.

Overview

A *Clifford circuit* is built from:

- Initial qubits in the state $0^{\otimes n}$,
- A sequence of gates drawn from the set $\{\text{CNOT}, H, S\}$, where

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}, \quad S = \begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix},$$

- Measurements in the computational (Z) basis (i.e., measuring σ_z on each qubit).

We will show that such circuits can be classically simulated in time polynomial in n (number of qubits) and T (number of gates).

Preliminaries: Pauli Operators and Their Binary Representation

We denote the single-qubit Pauli matrices by $\sigma_x, \sigma_y, \sigma_z$ (sometimes denoted X, Y, Z):

$$\sigma_x = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}, \quad \sigma_y = \begin{pmatrix}0 & -i\\ i & 0\end{pmatrix}, \quad \sigma_z = \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}.$$

Problem 1: Representing Single-Qubit Pauli Operators by Two Bits.

(a) We will use the notation (x, z) to represent a single-qubit Pauli operator. Define a rule that matches:

$$(0,0) \mapsto I, \quad (1,0) \mapsto \sigma_x, \quad (0,1) \mapsto \sigma_z, \quad (1,1) \mapsto \sigma_y.$$

Each bit in the pair (x, z) represents the presence (1) or absence (0) of a σ_x or σ_z factor. Why does $(1,1)$ correspond to σ_y (up to a global phase)?

(b) Briefly explain why ignoring the global phase does not affect measurement outcomes in quantum mechanics. (You do not need a rigorous proof, just the physical reasoning.)

QUESTION 1

Problem 2: Extending to n Qubits.

(a) An n-qubit Pauli operator is a tensor product of single-qubit Pauli operators, e.g., $\sigma_x \otimes \sigma_z \otimes \cdots \otimes I$. Explain how we can store an n-qubit Pauli operator by a $2n$-bit string:

$$(x_1, z_1) \parallel (x_2, z_2) \parallel \cdots \parallel (x_n, z_n).$$

(b) Give two examples of 3-qubit Pauli operators (e.g., $\sigma_y \otimes I \otimes \sigma_z$) and write them in the above bit-string notation.

Actions of the Clifford Gates on Pauli Operators

The key fact is that each Clifford gate G *conjugates* Pauli operators to Pauli operators (up to a global phase). Explicitly, $G\sigma G^\dagger$ is again some Pauli operator. In the bit form, this conjugation corresponds to simple *bitwise updates*.

QUESTION 2

Problem 3: Single-Qubit Gate Conjugation.
Consider a single qubit on which you apply H or S. Suppose this qubit is acted on by a Pauli operator with bits (x, z).

(a) **Hadamard gate H.** Show that

$$H\sigma_x H^\dagger = \sigma_z, \quad H\sigma_z H^\dagger = \sigma_x.$$

Conclude that, on the bit-level, (x, z) is mapped to (z, x). (Ignore global phases for now.)

(b) **Phase gate S.** Recall that $S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$. Show that

$$S\sigma_z S^\dagger = \sigma_z, \quad S\sigma_x S^\dagger = \sigma_y \quad \text{(up to a global phase } i\text{)}.$$

Conclude a rule in the bit form, for example $(x, z) \mapsto (x \oplus z, z)$. Verify this rule with a short calculation.

QUESTION 3

Problem 4: CNOT on Two Qubits.
For a two-qubit system, let qubit c be the control and qubit t be the target. Denote the bit-string for the operator on qubit c by (x_c, z_c) and on qubit t by (x_t, z_t). We apply $\text{CNOT}_{c \to t}$.

(a) Show explicitly how CNOT conjugates σ_x and σ_z on the control and target. (Hint: $\text{CNOT}(\sigma_x \otimes I)\text{CNOT}^\dagger = ?$, etc.)

(b) Based on part (a), deduce the update rule for $(x_c, z_c | x_t, z_t)$ in the bit-string representation. (You may give the final answer in terms of XOR operations.)

Stabilizer States and the Simulation Algorithm

Background (No Formal Group Theory Required). An n-qubit *stabilizer state* can be described as the unique $+1$ eigenstate of n independent, mutually commuting Pauli operators. In the *binary matrix* approach, we represent those n operators by n rows in a $2n$-bit matrix (plus some extra bits for phases, if needed).

QUESTION 4

Problem 5: Updating Stabilizer Generators Under Clifford Gates.
You now represent the quantum state using a list of n commuting n-qubit Pauli operators $P_1, \ldots, P_n$, which stabilize the state:

$$P_i \psi = \psi \quad \text{for all } i = 1, \ldots, n.$$

These P_i are stored in bitwise form as n rows of a $2n$-bit matrix (each row corresponds to one operator).

(a) Suppose you apply a Clifford gate G (either H, S, or CNOT) to one or two qubits. Show that the new state $G\psi$ is stabilized by the operators $GP_iG^\dagger$. Give a short explanation of why this holds.

(b) For each generator P_i, apply the appropriate bitwise update rule (from Problems 3 and 4) to compute $GP_iG^\dagger$ in bit-string form. Explain why the result is again a valid Pauli operator.

(c) Conclude that, if each gate update requires only bitwise operations on each of n rows of $2n$ bits, then the total simulation time for a circuit with T gates is polynomial in n and T.

QUESTION 5

Problem 6: Measurement in the Z Basis.
To *measure* qubit k in the σ_z basis, we need to determine the outcome probabilities (0 vs. 1) and update the post-measurement state.

(a) If there is a stabilizer generator with a σ_z on qubit k (i.e., the bit pattern $(0, 1)$ at position k), then ψ is already an eigenstate of σ_z on that qubit. Show why the measurement outcome is *deterministic* in that case.

(b) If all stabilizer generators act on qubit k by σ_x or I (i.e., $(1, 0)$ or $(0, 0)$), then the measurement outcome is random (50-50). Briefly explain how to *update* the stabilizer matrix after obtaining a measurement result. (Hint: You may need to replace one generator with σ_z on that qubit, ensuring it still commutes with the others.)

Final Conclusion

QUESTION 6

Problem 7: Putting It All Together.
Using all the above steps, write a short argument (one paragraph) explaining why:

Any circuit on n qubits using only $\{CNOT, H, S\}$ gates, starting in the $0^{\otimes n}$ state and ending in Z-basis measurements, can be simulated by a classical computer in time polynomial in n and in the number of gates.

You do not need to provide the constant factors; just argue that each gate and measurement can be handled by updating n bit-strings (the stabilizer generators), each of length $2n$, in polynomial time.