Analysis of Quantum Entanglement in Quantum Programs using Stabilizer Formalism

Kentaro Honda
Department of Computer Science, the University of Tokyo, Japan
honda@is.s.u-tokyo.ac.jp

Quantum entanglement plays an important role in quantum computation and communication. It is necessary for many protocols and computations, but causes unexpected disturbance of computational states. Hence, static analysis of quantum entanglement in quantum programs is necessary. Several papers studied the problem. They decided qubits were entangled if multiple qubits unitary gates are applied to them, and some refined this reasoning using information about the state of each separated qubit. However, they do not care about the fact that unitary gate undoes entanglement and that measurement may separate multiple qubits. In this paper, we extend prior work using stabilizer formalism. It refines reasoning about separability of quantum variables in quantum programs.

1 Introduction

Quantum entanglement plays an important role in quantum computation and communication. It allows us to teleport quantum states [3] and reduce necessary numbers of qubits for communication [4]. Moreover, it is the essential resource in a one-way quantum computation model [14] and indispensable for outperforming classical computers. Quantum entanglement also introduces some difficulty in compiling quantum programs. For example, when a system uses an ancilla, the ancilla is possibly entangled with the computation system and removal of it will disturb the computational state of the system. Compilers of quantum programs should care about existence of quantum entanglement. Hence, static analysis of quantum entanglement is necessary. Several papers studied the problem using types [10], abstract interpretation [12], and Hoare-like logic [13]. The first paper reasoned that two qubits are entangled whenever a two qubits gate is applied to these qubits. The other papers improved the reasoning by restricting two qubit gates to the controlled-not gate CX and by memorising information about the basis of separated qubits. Since CX does not create entanglement if the control qubit is in Z-basis or the target qubit is in X-basis, we can reason that two qubits are not entangled even after applying CX to the qubits. However, these papers do not care about the fact that unitary gate undoes entanglement. Our motivating example is as follows.

\[
\text{GHZ} \equiv \text{INIT}; \text{H}(q_0); \text{CX}(q_0, q_1); \text{CX}(q_1, q_2)
\]

where \text{INIT} changes states of all qubits \(q_0, q_1, q_2\) into \(|0\rangle\). \text{GHZ} creates \text{GHZ} state \(|\text{GHZ}\rangle \equiv \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)\), where all qubits are entangled. \text{SEP}_0 destroys the entanglement without measurement. Indeed, \((\text{CX} \otimes I)(I \otimes \text{CX})|\text{GHZ}\rangle = |+00\rangle\) and all qubits are separated. The prior work reasons correctly that entanglement exists after \text{GHZ} but incorrectly that entanglement still exists after \text{SEP}_0. Another example is

\[
\text{SEP}_1 \equiv \text{GHZ}; \text{meas}(q_0)
\]

\[
\text{NSEP} \equiv \text{GHZ}; \text{H}(q_0); \text{meas}(q_0).
\]
After executions, SEP$_1$ produces all separated qubits but NSEP does one separated and two entangled qubits regardless of the measurement results. In this paper, we borrow the framework of Perdrix’s work [12] and extend it using stabilizer formalism [11] [7] [9], which gives a segment of quantum computation that can be classically simulated. It refines reasoning about separability of quantum variables in quantum programs.

2 Preliminaries

2.1 Stabilizer Formalism

Stabilizer formalism allows us to express a certain class of states in a compact way.

Let $G_n$ be the Pauli group on $n$ qubits. The stabilizer $S$ of a non-trivial subspace $V_S$ of the $2^n$-dimensional complex Hilbert space $\mathcal{H}_{2^n}$ is \{ $P \in G_n \mid \forall |\psi\rangle \in V_S . \; P |\psi\rangle = |\psi\rangle$ \}. Any stabilizer $S$ is abelian and $-I^{\otimes n} \notin S$. A subgroup $S$ of $G_n$ is a stabilizer (on $n$ qubits) if it is the stabilizer of some non-trivial subspace of $\mathcal{H}_{2^n}$. If \{ $M_0, \ldots, M_{k-1}$ \} is a set of independent generators of $S$, we use \{ $M_0, \ldots, M_{k-1}$ \} to denote $S$. If $S = \langle M_0, \ldots, M_{k-1} \rangle$, the dimension of $V_S$ is $2^{n-k}$. In particular, if $k = n$, there exists a unique state $|\psi_S\rangle$ stabilized by $S$. We call a state $|\psi\rangle$ is a stabilizer state if $|\psi\rangle = |\psi_S\rangle$ for some stabilizer $S$. $P^{\pm}_{M_i} = \frac{1}{2} (I^{\otimes n} \pm M_i)$ is the projection onto eigenspaces corresponding to eigenvalues $\pm 1$.

Stabilizers have matrix expressions. Let $S = \langle M_0, \ldots, M_{k-1} \rangle$. Each generator $M_i$ has a form of either $\sigma_{i,0} \otimes \sigma_{i,1} \otimes \cdots \otimes \sigma_{i,n-1}$ or $-\sigma_{i,0} \otimes \sigma_{i,1} \otimes \cdots \otimes \sigma_{i,n-1}$ where $\sigma_{i,m}$ is a Pauli matrix, i.e. $\sigma_{i,m} \in \{ I, X, Y, Z \}$. A stabilizer array \[2\] is a $k \times (n+1)$ matrix whose $(i,j)$th entry is $\sigma_{i,j}$ for $j < n$ or the sign of $M_i$ for $j = n$, and it denotes $S$. For example, $\langle -ZZ, XX \rangle = \{ I, XX, YY, -ZZ \}$ stabilizes $\frac{1}{\sqrt{2}} (|01\rangle + |10\rangle)$.

\[
\begin{bmatrix}
  Z & Z & - \\
  X & X & +
\end{bmatrix}
\]
is a stabilizer array of the stabilizer. We identify the $i$th row of a stabilizer array and the generator $M_i$. Obviously, both permutation of rows and multiplication of the $i$th row and the $j$th row do not change the stabilizer provided $i \neq j$ where “multiplication of the $i$th row and the $j$th row” is replacement of the $i$th row with the product of the $i$th row and the $j$th row. Stabilizer arrays are compact but have sufficient information to their stabilizers. We use stabilizer arrays to operate stabilizers.

Let $S = \langle M_0, \ldots, M_{k-1} \rangle$ and $T = \langle N_0, \ldots, N_{l-1} \rangle$ be stabilizers on $k$ and $l$ qubits. Their tensor product $S \otimes T$ is the stabilizer $\langle M_0 \otimes I^{l'}, \ldots, M_{k-1} \otimes I^{l'}, I^{d'} \otimes N_0, \ldots, I^{d'} \otimes N_{l-1} \rangle$ on $k + l$ qubits. In stabilizer array expression, the tensor product is the direct sum of two matrices.

When $S = \langle M_0, \ldots, M_{n-1} \rangle$ is the stabilizer of $V_S$, $U M_i U^\dagger$ “stabilizes” $U V_S$ for any unitary gate $U$. However, some $U M_i U^\dagger$ may exceed $G_n$ and hence may not be a stabilizer. A Clifford gate is a unitary gate that sends any stabilizer to a stabilizer. Any Clifford gate can be composed of the controlled-X gate $\text{CX}$, the Hadamard gate $H$, and the phase gate $S$. A well-known non-Clifford gate is the $\frac{\pi}{8}$-gate $T$. Indeed, $\text{TXT}^\dagger = \frac{1}{\sqrt{2}} (X + Y)$ and $T |+\rangle = \frac{1}{\sqrt{2}} (|0\rangle + e^{\frac{\pi}{4}} |1\rangle)$ is not a stabilizer state.

Let $\langle M_0, \ldots, M_{n-1} \rangle$ be a stabilizer on $n$ qubits. If any $M_i$ commutes with $Z(j) \equiv I^{\otimes j} \otimes Z \otimes I^{\otimes n-j-1}$, i.e. the $j$th column of a stabilizer array consists of I or Z, then the measurement result of the $j$th qubit is deterministic and does not change the state. If not, the measurement result is probabilistic. Through multiplication of rows, we can take a unique generator $M_i$ that does not commute with $Z(j)$. The stabilizer of the post-measurement state is $\langle M_0, \ldots, M_{i-1}, \pm Z(j), M_{i+1}, \ldots, M_{n-1} \rangle$ if the measurement result is $\pm 1$, respectively.

K. Honda 263

\[2\]
2.2 Quantum Imperative Language

Following prior work \cite{12}, we use Quantum Imperative Language (QIL) as a target language. Fix the set $Q$ of quantum variables \{${q_0, \ldots, q_{N-1}}$\}. We assume $Q$ is finite and often identify a quantum variable and its index. The syntax of QIL \cite{11} is the following.

$$C, C' ::= \text{skip} | C ; C' | X(i) | Y(i) | Z(i) | H(i) | S(i) | T(i) | CX(i, j)$$

$$\text{if } i \text{ then } C \text{ else } C' \text{ fi} \mid \text{while } i \text{ do } C \text{ od}$$

where $i \neq j$. QIL is the set of QIL programs. The concrete denotational semantics of QIL is a superoperator $\llbracket \cdot \rrbracket_{QIL} : QIL \rightarrow D_{2N} \rightarrow D_{2N}$ where $D_n$ is the set of $n$-dimensional partial density matrices, which is a CPO \cite{15}.

$$\llbracket \text{skip} \rrbracket(\rho) = \rho$$

$$\llbracket C ; C' \rrbracket(\rho) = \llbracket C' \rrbracket(\llbracket C \rrbracket(\rho))$$

$$\llbracket U(i) \rrbracket(\rho) = U(i)\rho U(i)^\dagger$$

$$\llbracket CX(i, j) \rrbracket(\rho) = CX(i, j)\rho CX(i, j)^\dagger$$

$$\llbracket \text{if } i \text{ then } C \text{ else } C' \text{ fi} \rrbracket(\rho) = \llbracket C \rrbracket(\llbracket 0 \rrbracket(0)(\rho)\rho(0)\rho(0)0) + \llbracket C' \rrbracket(\llbracket 1 \rrbracket(1)(\rho)\rho(1)\rho(1)1)$$

$$\llbracket \text{while } i \text{ do } C \text{ od} \rrbracket(\rho) = \sum_{n \in \mathbb{N}} \llbracket 1 \rrbracket(1)\rho(\mu(\rho))\rho(1)\rho(1)1$$

where $U \in \{X, Y, Z, H, S, T\}$, $f(\rho) = \llbracket C \rrbracket(\llbracket 0 \rrbracket(0)(\rho)\rho(0)\rho(0)0)\rho(1)\rho(1)1$.

QIL has a control structure and hence we can change any state of a quantum variable into a constant.

$$\text{INIT}_i \equiv \text{if } i \text{ then skip else } X(i) \text{ fi}$$

$$\text{INIT} \equiv \text{INIT}_0; \text{INIT}_1; \cdots; \text{INIT}_{N-1}$$

Indeed, $\llbracket \text{GHZ} \rrbracket(\rho) = |\text{GHZ}\rangle\langle \text{GHZ}|$ and $\llbracket \text{SEP}_0 \rrbracket(\rho) = |+00\rangle\langle +00|$. 

In the work \cite{12}, an abstract domain $A^Q$ to analyse entanglement was introduced. An element of the domain is a pair $(b, \pi)$ of a partition $\pi$ of $Q$ and a function $b : Q \rightarrow \{1, X, Z, T\}$. $\pi$ denotes that the quantum state $\rho$ is $\pi$-separable:

$$\rho = \sum_k p_k \bigotimes_{A_j \in \pi} \rho_{k,j}$$

where $\rho_{k,j}$ is a quantum state of $A_j$. Moreover, if the $i$th qubit is separated from the others, $b(i)$ shows which basis it is. For example, if $b(i) = Z$, the quantum state $\rho$ is:

$$\rho = p_0 |0\rangle\langle 0| \otimes \rho_0 + p_1 |1\rangle\langle 1| \otimes \rho_1$$

for some $p_0, p_1, \rho_0, \rho_1$. It implies that the $i$th qubit will be still separated even after executing $CX(i, j)$.

3 Abstract domain on stabilizers

Although $A^Q$ gives us some information about separability of a quantum state, it contains no information about entanglement except that qubits are entangled. In order to analyse more, we will refine the abstract domain $A^Q$ using stabilizer formalism. We follow the idea of $A^Q$, where $Z$ and $X$ denote that a state can be
generators and contain none of $X$.

The relation $\leq$.

Let $\rho$. Let $\alpha$.

With some probability $p$, $\alpha$.

Hence, when we can divide a stabilizer into the tensor product of multiple stabilizers, it is good to separate them. In particular, if a stabilizer on multiple qubits contains $X_i$, $Y_i$, or $Z_i$, then the $i$th qubit can be separated from the others.

Naive algorithms on a stabilizer array allow us to compute whether $X_i$ belongs to a given stabilizer in $O(N)$ time and to divide a stabilizer into two stabilizers in $O(N^2)$ time.

Definition 3.1. Let $\mathcal{S}_k$ be the set of stabilizers on $k \geq 2$ qubits that are generated by $k$ independent generators and contain none of $X_i$, $Y_i$, and $Z_i$. $\mathcal{S}_1 = \{1, (X, (Y, (Z))\}$.

We add the non-stabilizer $\blacksquare$ to all $\mathcal{S}_k$. Define $\mathcal{S} = \bigcup_{k \leq N} \mathcal{S}_k$. We call $\alpha \in 2^Q \times \mathcal{S}$ a (stabilizer) assignment if $pr_0 \alpha$ is a partition of $Q$ and for any $(A, S) \in \alpha, S \in \mathcal{S}_{|A|}$. Here, $pr_i$ is the $i$th projection. The set of stabilizer assignments is $C^Q$.

Notation 3.2. Let $\alpha$ be an assignment. We sometimes regard $\alpha$ as a function from $Q$ to $2^Q \times \mathcal{S}$ such that $\alpha(i) = (A, S)$ where $i \in A$. We define $\alpha_0 = pr_0 \circ \alpha, \alpha_1 = pr_1 \circ \alpha$. Hence, $\alpha_0(i) \in 2^Q$ and $\alpha_1(i) \in \mathcal{S}$.

We also regard a partition of $Q$ as a function from $Q$ to $2^Q$. $\alpha[(A, S) / i]$ is a new assignment $(\alpha \setminus \alpha(i)) \cup \{(A, S)\}$. We extend the notation into $\alpha[\{A_0, S_0\}, \ldots, \{A_{k-1}, S_{k-1}\}] / i$ in a natural manner. $\alpha[S / i]$ means $(\alpha \setminus \alpha(i)) \cup \{(\alpha(i), S)\}$. $\alpha[S / i, j] = (\alpha \setminus (\alpha(i) \cup \alpha(j))) \cup \{((\alpha_0(i) \cup \alpha_0(j), S))\}$.

Definition 3.3. Let $\rho$ be a quantum state and $\alpha$ be an assignment. We write $\alpha \models \rho$ if

$$\rho = \sum_k p_k \bigotimes_{(A, S) \in \alpha} \rho^{k, (A, S)}$$

with some probability $p_k$ and some state $\rho^{k, (A, S)}$ on $A$ qubits where $\rho^{k, (A, S)}$ has a form of $1/2$ if $S = 1$ and $|\psi_S \rangle \langle \psi_S|$ if $S$ is another stabilizer.

Although an assignment tells how to separate a quantum state, it is just an overapproximation. Even if a stabilizer is assigned to two qubits, it does not mean the qubits are entangled. Indeed, although $1/2(1 \otimes 1)$ is a separable state, $\{(0, 1), (XX, ZZ)\} \equiv 1/2(1 \otimes 1)$.

Each assignment contains information about entanglement of a quantum state. Intuitively, an assignment $\alpha$ has more information than another assignment $\beta$ if $\beta \models \rho$ whenever $\alpha \models \rho$. It gives $C^Q$ a lattice structure: For $S, S' \in \mathcal{S}$, we write $S \leq S'$ if $S = I, S' = \blacksquare$, or $S = S'$. Obviously, $\leq$ is an order. Let $\leq \pi$ be an order of partitions: $\pi \leq \pi'$ if for any $A' \in \pi'$, there exist $A_0, \ldots, A_{k-1} \in \pi$ such that $A' = \bigcup_{\pi \in \{0, \ldots, k-1\}} A_i$. Moreover, we write $\alpha \leq \beta$ if $\alpha_0 \leq \pi \beta_0$ and for each $i \in Q$, $\bigwedge_{j \in \beta_0(i)} \alpha(j) \leq \beta_1(i)$ where

$$\bigwedge_{j \in J} (A_j, S_j) = \begin{cases} S_j & \text{(all } A_j \text{ are the same)} \\ 1 & \text{(all } S_j \text{ are } 1) \\ \blacksquare & \text{(otherwise)} \end{cases}.$$

The relation $\leq$ makes $C^Q$ a CPO.
Proposition 3.4. \(C^Q\) is a finite lattice and hence a CPO.

Proof. It is easy to see \(\leq\) is an order. The maximum assignment is \(\{(Q),\top\}\) and the minimum is \(\{\{|i\},1\} \mid i \in Q\). Let \(\alpha, \beta\) be assignments. Take the join \(\pi\) of \(\alpha_0\) and \(\beta_0\) with respect to \(\leq\). Let \(A \in \pi\). Define \(S\) as the join of \(\bigotimes_{j \in A} \alpha_1(j)\) and \(\bigotimes_{j \in A} \beta_1(j)\). The set of these pairs \((A,S)\) is the join of \(\alpha\) and \(\beta\). The meet of \(\alpha\) and \(\beta\) can be constructed similarly. \(\square\)

We define an abstract semantics \([\_]_{C}: \text{QIL} \to C^Q \to C^Q\) inductively. For simplicity, we define \(U\top = \top\) for any unitary \(U\) and \(\top \otimes S = \top = S \otimes \top\) for any \(S \in \mathcal{S}\), and assume that conditions are exclusive and an upper condition has priority.

\[
\begin{align*}
\text{[skip]}_{C}(\alpha) &= \alpha \\
[C;C^\prime]_{C}(\alpha) &= [C']_{C}[C]_{C}(\alpha) \\
[U(i)]_{C}(\alpha) &= \alpha[U(i)\alpha(i)U(i)^\dagger/i] \\
[T(i)]_{C}(\alpha) &= \begin{cases} 
\alpha & (\alpha_0(i) \text{ and } Z(i) \text{ commute}) \\
\alpha[\top/i] & (\text{otherwise}) 
\end{cases} \\
[CX(i,j)]_{C}(\alpha) &= \begin{cases} 
\alpha[\text{update}\{i,j\}, \alpha_0(i), CX(i,j)\alpha(i)CX(i,j)^\dagger/i] & (\alpha_0(i) = \alpha_0(j)) \\
\alpha[\langle Z\rangle/i] & (\alpha(i) = \langle Z\rangle, \alpha(j) = \langle X\rangle, \text{ or } \alpha(i) = \alpha(j) = 1) \\
\alpha[\langle X\rangle/j] & (\alpha(i) = 1) \\
\alpha[\langle Z\rangle/i, CX(i,j)\alpha(i)\alpha(j)CX(i,j)^\dagger/i] & (\alpha(j) = 1) \\
\alpha & (\text{otherwise}) 
\end{cases}
\end{align*}
\]

\[
\begin{align*}
\text{if } i \text{ then } C \text{ else } C^\prime \text{ fi }_{C}(\alpha) &= [C]_{C}(\text{meas}(i, \alpha)) \lor [C^\prime]_{C}(\text{meas}(i, \alpha)) \\
\text{while } i \text{ do } C \text{ od }_{C}(\alpha) &= \bigvee_{n \in \mathbb{N}} \text{meas}_s([C]_{C} \circ \text{meas}_s)^n(\alpha)
\end{align*}
\]

where \(U \in \{X, Y, Z, H, S\}\). \text{update} makes a “pseudo-” assignment to satisfy the condition that each stabilizer contains none of \(X(i), Y(i), \text{ and } Z(i)\). The first argument is possibly-unentangled qubits.

\[
\begin{align*}
\text{update}(J, A, \top) &= \{(A, \top)\} \\
\text{update}(0, A, S) &= \{(A, S)\} \\
\text{update}(\{i\} \cup J, A, S) &= \begin{cases} 
\{(i, S')\} \cup \text{update}(J, A \setminus \{i\}, S') & (S = S' \otimes S'' \text{ such that } S' \in \mathcal{S}_1 \text{ and } S' \text{ has non-identity entry only in the } i\text{th column}) \\
\text{update}(J, A, S) & (\text{otherwise}) 
\end{cases}
\end{align*}
\]

\(\text{meas}\) means measurement and \(\text{meas}_s(\alpha) = \text{meas}(i, \alpha)\). After measurement, the measured qubit is always separated.

\[
\begin{align*}
\text{meas}(i, \alpha) &= \begin{cases} 
\alpha[\langle Z\rangle/i] & (|\alpha_0(i)| = 1) \\
\alpha[\{\{i\}, \langle Z\rangle, \alpha_0(i) \setminus \{i\}, \top\}]/i & (\alpha(i) = \top) \\
\text{update}(\alpha_0(i), \alpha_0(i), \text{meas}_s(i, \alpha_1(i)))/i & (\text{otherwise}) 
\end{cases}
\end{align*}
\]

where \(\text{meas}_s\) is the measurement process of the \(i\)th qubit in stabilizer formalism.
In the previous section, we use stabilizers and the symbol \( \mathbf{■} \) that represents a non-stabilizer. The symbol \( \mathbf{■} \) contains no information. It just states that the state of the associated qubits is unknown. The abstract semantics \( \mathbb{C} \) introduces the symbol when it faces the non-Clifford gate \( T \) because the post-execution state is a non-stabilizer state. Can we really extract meaningful information from the post-execution state? Let us take the following QIL program:

\[ \text{exm}_1 \equiv \text{GHZ}; \text{T}(1); \text{meas}(0) \]
The abstract semantics

\[
\begin{bmatrix}
X & X & X \\
Z & Z & I \\
Z & I & Z \\
\end{bmatrix}
\xrightarrow{T(1)}
\begin{bmatrix}
\bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet \\
\end{bmatrix}
\xrightarrow{\text{meas}(0)}
\begin{bmatrix}
Z \\
Z \\
Z \\
\end{bmatrix}
\]

tells us that the first qubit is separated but the second and the third qubits may be entangled last. Now, let us try not to fill the matrix with \(\bullet\) when \(T\) appears but to memorise the applied gates. Recall that \(USU^\dagger\) “stabilizes” \(UV_S\) if \(S\) is the stabilizer of \(V_S\).

\[
\begin{bmatrix}
X & X & X \\
Z & Z & I \\
Z & I & Z \\
\end{bmatrix}
\xrightarrow{T(1)}
\begin{bmatrix}
X & TXT^\dagger & X \\
Z & Z & Z \\
Z & I & I \\
\end{bmatrix}
\xrightarrow{\text{meas}(0)}
\begin{bmatrix}
Z \\
Z \\
Z \\
\end{bmatrix}
\]

It means all qubits are separated. Indeed, \([\text{exm}_1](\rho) = \frac{1}{2} (|000\rangle\langle000| + |111\rangle\langle111|)\). The example shows that the effect of \(T\) may be bounded locally and will be removed later. We introduce a new symbol \(\heartsuit\), which means a unitary matrix that may not be a Pauli matrix or their tensor product. Note that \(\heartsuit\) means not only a single qubit unitary matrix but also an \(n\) qubit unitary matrix. Using the symbol \(\heartsuit\), we will extend our abstract domain \(C^Q\) to a new domain \(E^Q\). Before doing it, we extend stabilizers so that they allow us to put \(\heartsuit\) on them.

**Definition 4.1.** Let \(k\) be a positive integer and \(A\) be a \(k \times k\) matrix whose entries are in \{I, X, Y, Z, \heartsuit\}. We now identify two matrices \(A\) and \(B\) if one can be converted into the other via permutations and multiplications of rows. Here, \(\heartsuit\) behaves as an absorbing element. We call a row containing the symbol \(\heartsuit\) and a row containing no \(\heartsuit\) a \(\heartsuit\)-row and an L-row, respectively. We always require any L-rows commute and these rows are independent. Moreover, we require that for any \(\heartsuit\)-row \(R_i\) and row \(M_j\), by substituting I, X, Y, or Z for each \(\heartsuit\) in \(R_i\) and \(M_j\), the result rows can commute. For example, the matrix consisting of two rows \(\heartsuit\)X and IZ is excluded, but the matrix consisting of \(\heartsuit\)X and XZ is right because substitution of Z for \(\heartsuit\) makes these rows commute. Finally, we exclude some matrices. Let \(k \geq 2\). If a matrix has a row II \(-\cdots\) I, \(X(j), Y(j), Z(j), \) or \(\heartsuit(j)\), then it is excluded. We name the set of those matrices \(\mathcal{E}_k\). Then, \(\bullet\) is added into all \(\mathcal{E}_k\). \(\mathcal{E}\) is the union of these \(\mathcal{E}_k\)s.

**Example 4.2.**

\[
\begin{bmatrix}
I \\
\heartsuit \\
X \heartsuit X \\
\end{bmatrix},
\begin{bmatrix}
X \heartsuit Y \\
Z \heartsuit Z \\
Y \heartsuit Y \\
\end{bmatrix} \in \mathcal{E},
\begin{bmatrix}
\heartsuit \Z \Y \\
I \X \Y \\
Z \heartsuit \heartsuit \\
\end{bmatrix},
\begin{bmatrix}
\heartsuit \Y \\
I \X \\
\heartsuit \heartsuit \\
\end{bmatrix} \notin \mathcal{E}
\]

The third matrix is an abstraction of matrices such as

\[
\begin{bmatrix}
X & I \\
I & X \\
Z & Z \\
\end{bmatrix},
\begin{bmatrix}
X & Z \\
I & X \\
Z & X \\
\end{bmatrix},
\begin{bmatrix}
X & HTXT^\dagger H^\dagger \\
I & X \\
Z & Z \\
\end{bmatrix}.
\]

Recall that \(\mathcal{S}\) has the order \(\leq_s\). Regardless of the addition of \(\heartsuit\), the same definition seems to give an order of \(\mathcal{E}\). \(E \in \mathcal{E}\) is lower than or equal to \(E' \in \mathcal{E}\) if \(E = I, E' = \bullet, \) or \(E = E'\). However, it does not answer our purpose. Recall the join operator corresponds with the summation of density matrices. For example, \([X \heartsuit \heartsuit ]\) may represent \([X I] \) or \([X I] \). However, the summation of stabilized states by them does not always have the form of \([X \heartsuit \heartsuit ]\). The example shows the join of \([X \heartsuit \heartsuit ]\) and \([X \heartsuit \heartsuit ]\) should not be itself, so the “order” is not reflexive.
In order to obtain a join operator, we remove rows that contain \(\bigodot\). We give up keeping information about unitary matrices when we take a join.

**Definition 4.3.** Take \(A \in \mathcal{E}_k\). Remove all \(\bigodot\)-rows. If all rows are \(\bigodot\)-rows, we obtain \(\blacksquare\). We call this procedure *normalisation* and these matrices *normal forms*. The set of normal forms is \(\mathcal{F}_k\) and the union of them is \(\mathcal{F}\). We redefine \(\mathcal{E}_k\) so that it includes any element of \(\mathcal{E}_k\) even if some rows are removed. \(\mathcal{E}\) is the union of them. Note \(\mathcal{F}_k \subset \mathcal{E}_k\) and thus \(\mathcal{F} \subset \mathcal{E}\).

**Notation 4.4.** For each \(E \in \mathcal{E}\), \(E_{nl}\) is the normal form of \(E\).

**Example 4.5.**

\[
\begin{bmatrix}
1 \\
\blacksquare \\
X & Y & Z
\end{bmatrix} \in \mathcal{F}
\]

\(\mathcal{F}\) has an order \(\leq_f\): \(F \leq_f F'\) if \(F = I, F' = \blacksquare\), or \(F = F'\). Obviously, \(\mathcal{F}\) has the maximum, the minimum, and the join and the meet of any two elements. We can take an approximation of a join operator of \(\mathcal{E}\) via the subset \(\mathcal{F}\).

Now, we define our second abstract domain \(E^Q\).

**Definition 4.6.** We call \(\gamma \subset 2^Q \times \mathcal{E}\) an extended (stabilizer) assignment if \(pr_0 \gamma\) is a partition of \(Q\) and for any \((A, E) \in \gamma, E \in \mathcal{E}_|A|\). The set of extended assignments is \(E^Q\). For each extended assignment \(\gamma\), an extend assignment \(\{ (A, E_{nl}) \mid (A, E) \in \gamma \}\) is the normal form of \(\gamma\). \(F^Q\) is the set of normal forms of extended assignments.

**Notation 4.7.** For extended assignments, we use the same notation as for assignments.

**Definition 4.8.** Let \(\rho\) be a quantum state and \(\gamma\) be an extended assignment. We write \(\gamma \models \rho\) if \(\rho\) is \(pr_0 \gamma\)-separable, for any L-row \(L_k\) of any \(\gamma(i)\) that is not I or \(\blacksquare\), \(P_L^+ P_L^- = 0\), and for any \(i\) such that \(\gamma(i) = I\), \(\rho = \frac{1}{2} I(i) \otimes \rho'\) with some state \(\rho'\) of the \(Q \setminus i\) qubits. Recall \(P_k^\pm = \frac{1}{2} (I^\otimes n \pm L_k)\).

The same construction as \(C^Q\) makes \(F^Q\) a CPO. Although \(E^Q\) does not have joins, we can define an approximate join operator \(\uplus\) on \(E^Q\) through \(F^Q\): for each \(\gamma, \delta \in E^Q\), \(\gamma \uplus \delta\) is the join of the normal forms of \(\gamma\) and \(\delta\). Note that the approximate join \(\uplus\) of two elements can be computed efficiently. Now, we define our second abstract semantics \(\lfloor \cdot \rfloor_E: \text{QIL} \to E^Q \to E^Q\). Since \(\bigodot\) loses some information, we have to avoid introducing \(\bigodot\) if possible. For simplicity, we define \(U \blacksquare U^\dagger = \blacksquare\) for any \(U\), \(\blacksquare \otimes E = \blacksquare = E \otimes \blacksquare\) for any \(E \in \mathcal{E}\), \(U \bigodot U^\dagger = \bigodot\) for any 1 qubit unitary \(U\), and \(\text{CX}(\bigodot U)\text{CX}^\dagger = \text{CX}(U \bigodot)\text{CX}^\dagger = \bigodot\) for any \(U\). Moreover, we assume that conditions are exclusive and an upper condition has priority.

\[
\begin{align*}
\lfloor \text{skip} \rfloor_E(\gamma) &= \gamma \\
\lfloor C; C' \rfloor_E(\gamma) &= \lfloor C' \rfloor_E(\lfloor C \rfloor_E(\gamma)) \\
\lfloor U(i) \rfloor_E(\gamma) &= \gamma(U(i) \gamma(i) U^\dagger(i)/i) \\
\lfloor T(i) \rfloor_E(\gamma) &= \begin{cases} 
\gamma & (\gamma(i) \text{ and } Z(i) \text{ commute}) \\
\gamma(\text{addC}(i, \gamma(i))/i) & (\text{otherwise}) 
\end{cases} \\
\lfloor \text{CX}(i, j) \rfloor_E(\gamma) &= \begin{cases} 
\gamma(\text{update}_E(\{i, j\}, \gamma_0(i), \text{CX}_{(i, j)} \gamma(i) \text{CX}_{(i, j)}^\dagger)/i) & (\gamma(i) = \gamma(j)) \\
\gamma & (\gamma(i) = (Z), \gamma(j) = (X), \text{ or } \\
\gamma(i) = (\gamma(j) = I) \\
\gamma(Z)/i \\
\gamma(X)/j \\
\gamma(\text{CX}_{(i, j)}(\gamma(i) \otimes \gamma(j) \text{CX}_{(i, j)}^\dagger)/i, j) & (\text{otherwise}) 
\end{cases}
\end{align*}
\]
The result \( \text{update}_E(J,A,E) \) is computed as follows. (1) If \( E \) belongs to \( \mathcal{S} \), then \( \text{update}(J,A,E) \) is the result. (2) If not, take all \( j_0, \ldots, j_{k-1} \in J \) such that \( E \) has rows \( \hat{\diamond}_{j_i} \) where \( \hat{\diamond}_{j_i} \in \{X,Y,Z\} \). When there is no such \( j_i \), \( \{(A,E)\} \) is the result. Otherwise, define \( K = A \setminus \{j_0, \ldots, j_{k-1}\} \). Then, the result is \( \{(K,\bullet)\} \cup \{(\{j_i\},\hat{\diamond}_{j_i}) \mid l = 0, \ldots, k-1\} \).

The result of \( \text{meas}_E(i, \gamma) \) varies with \( \gamma \). (1) If \( |\gamma_0(i)\rangle = 1 \), then \( \gamma'_i(Z)/i \) is the result. (2) If \( \gamma_i(i) \) belongs to \( \mathcal{S} \setminus \mathcal{S}_i \), then \( \gamma'_\text{update}(\gamma_0(i), \gamma_i(i), \text{meas}_E(i, \gamma_i(i))/i) \), which is the same as \( \text{meas} \). (3) If exactly one row of \( \gamma_i(i) \) has X or Y in the \( i \)th column and the others have I or Z, \( \text{meas}_E(i, \gamma) \) is computed as follows. First, the row and the \( i \)th column are removed from \( \gamma_i(i) \). Let us call the matrix \( E' \). Then \( \gamma'_i((\{i\},Z)) \cup \text{update}_E((\gamma_0(i) \setminus \{i\}), (\gamma_0(i) \setminus \{i\}, E'))/i \) is the result. (4) Otherwise, we cannot obtain information about the post-measurement state. The result is \( \gamma'_i((\{i\},Z), (\gamma_0(i) \setminus \{i\}, \bullet))/i \).

The function \( \text{add}_\bigtriangledown \) changes X and Y in the \( i \)th column into \( \bigtriangledown \). By the definition of equality in \( \mathcal{S} \), we can assume that exactly one of the following holds: (1) the \( i \)th column does not contain X or Y, (2) exactly one L-row has X or Y in the \( i \)th column, and (3) only \( \bigtriangledown \)-rows have X or Y in the \( i \)th column. In the first case, \( \text{add}_\bigtriangledown \) does nothing and returns the second argument. In the second and third cases, \( \text{add}_\bigtriangledown \) changes all X and Y in the \( i \)th column into \( \bigtriangledown \) and returns the matrix. Hence, \( \text{add}_\bigtriangledown \) changes at most one L-row into a \( \bigtriangledown \)-row.

**Example 4.9.** Now, we compute \( [\text{exm_1}]_E(\gamma) \).

\[
\begin{bmatrix}
X & X & Z \\
Z & Z & I \\
Z & I & Z \\
\end{bmatrix}
\xrightarrow{T(1)}
\begin{bmatrix}
X & \bigtriangledown & X \\
Z & Z & I \\
Z & I & Z \\
\end{bmatrix}
\xrightarrow{\text{meas}(0)}
\begin{bmatrix}
Z & Z \\
Z \\
\end{bmatrix}
\]

Thus, we conclude that all qubits are separated.

Finally, we show \( [\cdot]_E \) is sound.

**Theorem 4.10.** For any state \( \rho \), extended assignment \( \gamma \), and program \( C \), \( \gamma \vdash \rho \) implies \( \llbracket C \rrbracket_E(\gamma) \models \llbracket C \rrbracket(\rho) \).

**Proof.** By induction on the structure of \( C \). For \( \text{skip} \), \( C;C', U(i), \) and \( T(i) \), it is easy. For \( CX(i,j) \), since the number of \( \bigtriangledown \)-rows does not increase, the statement holds. Extended stabilisers also satisfy \( \gamma \uplus \delta \vdash \rho + \sigma \) whenever \( \gamma \vdash \rho \) and \( \delta \vdash \sigma \). For \( \text{if } i \text{ then } C \text{ else } C' \text{ fi} \), we have to check \( \text{meas}_E \). However, since it also just decrease the number of \( \bigtriangledown \)-rows, \( \text{meas}_E(\gamma) \models |0\rangle|0\rangle(p)0|0\rangle|0\rangle \). Hence, the statement holds for \( \text{if } i \text{ then } C \text{ else } C' \text{ fi} \). Finally, we show for \( \text{while } i \text{ do } C \text{ od} \). Since \( C^Q \) is finite, \( \llbracket \text{while } i \text{ do } C \text{ od} \rrbracket_E(\gamma) \models \sum_{n \leq M} |1\rangle|1\rangle|f^n(\rho)|1\rangle|1\rangle \) for sufficiently large \( M \). Therefore, the statement holds by continuity of projection.

## 5 Conclusion

We used stabilizer formalism to improve entanglement analysis in quantum programs. First, we introduced an abstract domain \( C^Q \) and an abstract semantics. It assigns stabilizers or non-stabilizers to each
segment of a quantum state, where non-stabilizers are assigned when non-Clifford gates are applied to the segment. The method enables us to analyse separability of qubits in quantum programs more precisely. Specifically, we could deduce that all qubits are separated after executing SEP$_0$ or SEP$_1$. Moreover, we defined an abstract domain $E^Q$, as $C^Q$ introduces too many non-stabilizers. Even when non-Clifford gates appear, the domain does not discard stabilizers but keeps Pauli matrices that are not disturbed by the gates. Hence, it suppresses effects of non-Clifford gates that will be removed later. We showed soundness of both semantics.

In a field of model checking, stabilizer formalism was used to verify quantum programs and analyse entanglement of those programs [5, 6]. However, in these studies, quantum gates in a target language were restricted to only Clifford gates. It is worth noting that our target language QIL has a non-Clifford gate. This is a big advantage of our work and actually one of the challenges of our work was how to manage the non-Clifford gate. We restricted the effect by overapproximation. Although we refined the approximation from $C^Q$ to $E^Q$, further refinement is still needed such as finding a better approximate join operator in $E^Q$.

Acknowledgements

We thank Yoshihiko Kakutani for helpful comments. This work was supported by JSPS Grant-in-Aid for JSPS Fellows Grant No. 26 - 9148.

References

[1] Scott Aaronson & Daniel Gottesman (2004): Improved simulation of stabilizer circuits. Physical Review A 70, p. 052328, doi:10.1103/PhysRevA.70.052328
[2] Koenraad M R Audenaert & Martin B Plenio (2005): Entanglement on mixed stabilizer states: normal forms and reduction procedures. New Journal of Physics 7(1), p. 170, doi:10.1088/1367-2630/7/1/170
[3] Charles H. Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres & William K. Wootters (1993): Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. Physical Review Letters 70(13), pp. 1895–1899, doi:10.1103/PhysRevLett.70.1895
[4] Charles H. Bennett & Stephen J. Wiesner (1992): Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states. Physical Review Letters 69, pp. 2881–2884, doi:10.1103/PhysRevLett.69.2881
[5] Simon J. Gay, Rajagopal Nagarajan & Nikolaos Papanikolaou (2008): QMC: A Model Checker for Quantum Systems. In Aarti Gupta & Sharad Malik, editors: Computer Aided Verification, Lecture Notes in Computer Science 5123, Springer Berlin Heidelberg, pp. 543–547, doi:10.1007/978-3-540-70545-1_51
[6] Simon J. Gay, Rajagopal Nagarajan & Nikolaos Papanikolaou (2009): Specification and Verification of Quantum Protocols. In Simon Gay & Ian Mackie, editors: Semantic Techniques in Quantum Computation, Cambridge University Press, pp. 414–472, doi:10.1017/CBO9781139193313.012 Cambridge Books Online.
[7] Daniel Gottesman (1996): Class of quantum error-correcting codes saturating the quantum Hamming bound. Physical Review A 54, pp. 1862–1868, doi:10.1103/PhysRevA.54.1862
[8] Philippe Jorrand & Simon Perdrix (2009): Abstract Interpretation Techniques for Quantum Computation. In Simon Gay & Ian Mackie, editors: Semantic Techniques in Quantum Computation, Cambridge University Press, pp. 206–234, doi:10.1017/CBO9781139193313.007
[9] Michael A. Nielsen & Isaac L. Chuang (2000): Quantum Computation and Quantum Information. Cambridge University Press.
Analysis of Quantum Entanglement in Quantum Programs using Stabilizer Formalism

[10] Simon Perdrix (2007): Quantum Patterns and Types for Entanglement and Separability. Electronic Notes in Theoretical Computer Science 170(0), pp. 125–138, doi:10.1016/j.entcs.2006.12.015

[11] Simon Perdrix (2008): A Hierarchy of Quantum Semantics. Electronic Notes in Theoretical Computer Science 192(3), pp. 71–83, doi:10.1016/j.entcs.2008.10.028

[12] Simon Perdrix (2008): Quantum Entanglement Analysis Based on Abstract Interpretation. In María Alpuente & Germán Vidal, editors: Static Analysis, Lecture Notes in Computer Science 5079, Springer Berlin Heidelberg, pp. 270–282, doi:10.1007/978-3-540-69166-2_18

[13] Frédéric Prost & Chaouki Zerrari (2009): Reasoning about Entanglement and Separability in Quantum Higher-Order Functions. In CristianS. Calude, JoséFélix Costa, Nachum Dershowitz, Elisabete Freire & Grzegorz Rozenberg, editors: Unconventional Computation, Lecture Notes in Computer Science 5715, Springer Berlin Heidelberg, pp. 219–235, doi:10.1007/978-3-642-03745-0_25

[14] Robert Raussendorf & Hans J. Briegel (2001): A One-Way Quantum Computer. Physical Review Letters 86, pp. 5188–5191, doi:10.1103/PhysRevLett.86.5188

[15] Peter Selinger (2004): Towards a quantum programming language. Mathematical Structures in Computer Science 14, pp. 527–586, doi:10.1017/S0960129504004256