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 reduces 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.

$$\begin{aligned}\mathtt{GHZ} &\equiv \mathtt{INIT;H(}q_0\mathtt{);CX(}q_0\mathtt{,}q_1\mathtt{);CX(}q_1\mathtt{,}q_2\mathtt{)} \\ \mathtt{SEP}_0 &\equiv \mathtt{GHZ;CX(}q_0\mathtt{,}q_1\mathtt{);CX(}q_0\mathtt{,}q_2\mathtt{)}\end{aligned}$$

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

$$\begin{aligned}\mathtt{SEP}_1 &\equiv \mathtt{GHZ;meas(}q_0\mathtt{)} \\ \mathtt{NSEP} &\equiv \mathtt{GHZ;H(}q_0\mathtt{);meas(}q_0\mathtt{)}.\end{aligned}$$

To appear in EPTCS.

After executions, $\mathtt{SEP}_1$ 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 [1, 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 nontrivial subspace V_S of the 2^n-dimensional complex Hilbert space $\mathscr{H}_{2^n}$ is $\{P \in G_n \mid \forall |\psi\rangle \in V_S.\ P|\psi\rangle = |\psi\rangle\}$. Any stabilizer S is abelian and $-\mathrm{I}^{\otimes n} \notin S$. A subgroup S of G_n is a stabilizer (on n qubits) if it is the stabilizer of some nontrivial subspace of $\mathscr{H}_{2^n}$. If $\{M_0,\dots,M_{k-1}\}$ is a set of independent generators of S, we use $\langle M_0,\dots,M_{k-1}\rangle$ to denote S. If $S = \langle M_0,\dots,M_{k-1}\rangle$, 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. $P^{\pm}_{M_i} = \frac{1}{2}(\mathrm{I}^{\otimes n} \pm M_i)$ is the projection onto eigenspaces corresponding to eigenvalues ± 1.

Stabilizers have matrix expressions. Let $S = \langle M_0,\dots,M_{k-1}\rangle$. Each generator M_l has a form of either $\sigma_{l,0}\otimes\sigma_{l,1}\otimes\cdots\otimes\sigma_{l,n-1}$ or $-\sigma_{l,0}\otimes\sigma_{l,1}\otimes\cdots\otimes\sigma_{l,n-1}$ where $\sigma_{l,m}$ is a Pauli matrix, i.e. $\sigma_{l,m} \in \{\mathrm{I},\mathrm{X},\mathrm{Y},\mathrm{Z}\}$. 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, $\langle -\mathrm{ZZ},\mathrm{XX}\rangle = \{\mathrm{I},\mathrm{XX},\mathrm{YY},-\mathrm{ZZ}\}$ stabilizes $\frac{1}{\sqrt{2}}(|01\rangle+|10\rangle)$. $\begin{bmatrix} \mathrm{Z} & \mathrm{Z} & - \\ \mathrm{X} & \mathrm{X} & + \end{bmatrix}$ is a stabilizer array of the stabilizer. We identify the ith row of a stabilizer array and the generator M_i. Obviously, both permutation of rows and multiplication of the ith row and the jth row do not change the stabilizer provided $i \neq j$ where "multiplication of the ith row and the jth row" is replacement of the ith row with the product of the ith row and the jth row. Stabilizer arrays are compact but have sufficient information to their stabilizers. We use stabilizer arrays to operate stabilizers.

Let $S = \langle M_0,\dots,M_{k-1}\rangle$ and $T = \langle N_0,\dots,N_{l-1}\rangle$ be stabilizers on k and l qubits. Their tensor product $S\otimes T$ is the stabilizer $\langle M_0\otimes \mathrm{I}^{\otimes l},\dots,M_{k-1}\otimes \mathrm{I}^{\otimes l},\mathrm{I}^{\otimes k}\otimes N_0,\dots,\mathrm{I}^{\otimes k}\otimes N_{l-1}\rangle$ on $k+l$ qubits. In stabilizer array expression, the tensor product is the direct sum of two matrices.

When $S = \langle M_0,\dots,M_{n-1}\rangle$ is the stabilizer of V_S, $USU^\dagger = \langle UM_0U^\dagger,\dots,UM_{n-1}U^\dagger\rangle$ "stabilizes" UV_S for any unitary gate U. However, some $UM_iU^\dagger$ 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 from the controlled-X gate CX, the Hadamard gate H, and the phase gate S. A well-known non-Clifford gate is the $\frac{\pi}{8}$-gate T. Indeed, $\mathrm{TXT}^\dagger = \frac{1}{\sqrt{2}}(\mathrm{X}+\mathrm{Y})$ and $\mathrm{T}|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{\pi}{4}}|1\rangle)$ is not a stabilizer state.

Let $\langle M_0,\dots,M_{n-1}\rangle$ be a stabilizer on n qubits. If any M_i commutes with $\mathrm{Z}_{(j)} \equiv \mathrm{I}^{\otimes j}\otimes\mathrm{Z}\otimes\mathrm{I}^{\otimes n-j-1}$, i.e. the jth column of a stabilizer array consists of I or Z, then the measurement result of the jth 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 $\mathrm{Z}_{(j)}$. The stabilizer of the post-measurement state is $\langle M_0,\dots,M_{i-1},\pm\mathrm{Z}_{(j)},M_{i+1},\dots,M_{n-1}\rangle$ if the measurement result is ± 1, respectively.

2.2 Quantum Imperative Language

Following prior work [12], we use Quantum Imperative Language (QIL) as a target language. Fix the set $\mathbf{Q}$ of quantum variables $\{\mathtt{q}_0, \ldots, \mathtt{q}_{N-1}\}$. We assume $\mathbf{Q}$ is finite and thus often identify a quantum variable and its index. The syntax of QIL [11] is the following.

$$\begin{aligned} C, C' ::= {} & \mathtt{skip} \mid C\,\mathtt{;}\,C' \mid \mathtt{X(}i\mathtt{)} \mid \mathtt{Y(}i\mathtt{)} \mid \mathtt{Z(}i\mathtt{)} \mid \mathtt{H(}i\mathtt{)} \mid \mathtt{S(}i\mathtt{)} \mid \mathtt{T(}i\mathtt{)} \mid \mathtt{CX(}i\mathtt{,}\, j\mathtt{)} \\ & \mid \mathtt{if}\ i\ \mathtt{then}\ C\ \mathtt{else}\ C'\ \mathtt{fi} \mid \mathtt{while}\ i\ \mathtt{do}\ C\ \mathtt{od} \end{aligned}$$

where $i \neq j$. **QIL** is the set of QIL programs. The concrete denotational semantics of QIL is a superoperator $[\![\cdot]\!] \colon \mathbf{QIL} \to D_{2^N} \to D_{2^N}$ where D_n is the set of n-dimensional partial density matrices, which is a CPO [15].

$$\begin{aligned} [\![\mathtt{skip}]\!](\rho) &= \rho \\ [\![C\,\mathtt{;}\,C']\!](\rho) &= [\![C']\!]([\![C]\!](\rho)) \\ [\![U(i)]\!](\rho) &= U_{(i)} \rho U_{(i)}^{\dagger} \\ [\![\mathtt{CX(}i\mathtt{,}\,j\mathtt{)}]\!](\rho) &= \mathrm{CX}_{(i,j)} \rho \mathrm{CX}_{(i,j)}^{\dagger} \\ [\![\mathtt{if}\ i\ \mathtt{then}\ C\ \mathtt{else}\ C'\ \mathtt{fi}]\!](\rho) &= [\![C]\!](|0\rangle\langle 0|_{(i)} \rho |0\rangle\langle 0|_{(i)}) + [\![C']\!](|1\rangle\langle 1|_{(i)} \rho |1\rangle\langle 1|_{(i)}) \\ [\![\mathtt{while}\ i\ \mathtt{do}\ C\ \mathtt{od}]\!](\rho) &= \sum_{n \in \mathbb{N}} |1\rangle\langle 1|_{(i)} f^n(\rho) |1\rangle\langle 1|_{(i)} \end{aligned}$$

where $U \in \{\mathtt{X}, \mathtt{Y}, \mathtt{Z}, \mathtt{H}, \mathtt{S}, \mathtt{T}\}$, $f(\rho) = [\![C]\!](|0\rangle\langle 0|_{(i)} \rho |0\rangle\langle 0|_{(i)})$.

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

$$\begin{aligned} \mathtt{INIT}_i &\equiv \mathtt{if}\ i\ \mathtt{then}\ \mathtt{skip}\ \mathtt{else}\ \mathtt{X(}i\mathtt{)}\ \mathtt{fi} \\ \mathtt{INIT} &\equiv \mathtt{INIT}_0\mathtt{;}\mathtt{INIT}_1\mathtt{;}\cdots\mathtt{;}\mathtt{INIT}_{N-1} \end{aligned}$$

Indeed, $[\![\mathtt{GHZ}]\!](\rho) = |\mathrm{GHZ}\rangle\langle\mathrm{GHZ}|$ and $[\![\mathtt{SEP}_0]\!](\rho) = |{+}00\rangle\langle{+}00|$.

In the work [12], an abstract domain $A^{\mathbf{Q}}$ to analyse entanglement was introduced. An element of the domain is a pair (b, π) of a partition π of $\mathbf{Q}$ and a function $b \colon \mathbf{Q} \to \{\mathrm{I}, \mathrm{X}, \mathrm{Z}, \top\}$. π denotes that the quantum state ρ is π-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 ith qubit is separated from the others, $b(i)$ shows which basis it is. For example, if $b(i) = \mathrm{Z}$, the quantum state ρ is:

$$\rho = p_0 |0\rangle\langle 0|_{(i)} \otimes \rho_0 + p_1 |1\rangle\langle 1|_{(i)} \otimes \rho_1$$

for some p_0, p_1, ρ_0, ρ_1. It implies that the ith qubit will be still separated even after executing $\mathtt{CX(}i\mathtt{,}\,j\mathtt{)}$.

3 Abstract domain on stabilizers

Although $A^{\mathbf{Q}}$ 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 $A^{\mathbf{Q}}$ using the stabilizer formalism. We follow the idea of $A^{\mathbf{Q}}$, where Z and X denote that a state can be separated through $|0\rangle, |1\rangle$ and $|+\rangle, |-\rangle$ respectively. We suppose that a stabilizer $S = \langle M_0, \ldots, M_{n-1} \rangle$ on N qubits represents not only the stabilized state $|\psi_S\rangle$ but also the eigenstates of it, i.e. $\{|\psi\rangle \mid \forall M_i\ M_i|\psi\rangle = |\psi\rangle \text{ or } M_i|\psi\rangle = -|\psi\rangle\}$. We reuse $|\psi_S\rangle$ to denote an eigenstate. The sign of each generator has no longer any meaning. From now on, we assume any generator has the plus sign and we ignore the last column of any stabilizer array.

Our idea of using stabilizers, of course, has a problem about non-Clifford gates. Since QIL has the $\frac{\pi}{8}$-gate `T`, even if we start an execution of a QIL program from a stabilizer state, we may not get a stabilizer state. We prepare a symbol $\blacksquare$ that denotes a non-stabilizer.

Now, we introduce our abstract domain $C^{\mathbf{Q}}$, which is composed of assignments of stabilizers to each segment of partitions of $\mathbf{Q}$. When `T(`i`)` appears, we forget about a stabilizer that expresses the current state of the segment containing the ith qubit, and keep just the symbol $\blacksquare$. Hence, when we can divide a stabilizer into the tensor product of multiple stabilizers, it is good to separated them. In particular, if a stabilizer on multiple qubits contains either $\mathrm{X}_{(i)}$, $\mathrm{Y}_{(i)}$, or $\mathrm{Z}_{(i)}$, then the ith qubit can be separated from the others. Naive algorithms on a stabilizer array allow us to compute whether $\mathrm{X}_{(i)}$ 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 $\mathscr{S}_k$ be the set of stabilizers on $k \geq 2$ qubits that are generated by k independent generators and contain neither X_i, Y_i, nor Z_i. $\mathscr{S}_1 = \{\mathrm{I}, \langle \mathrm{X} \rangle, \langle \mathrm{Y} \rangle, \langle \mathrm{Z} \rangle\}$. We add the non-stabilizer $\blacksquare$ to all $\mathscr{S}_k$. Define $\mathscr{S} = \bigcup_{k \leq N} \mathscr{S}_k$. We call $\alpha \subset 2^{\mathbf{Q}} \times \mathscr{S}$ a *(stabilizer) assignment* if $\mathrm{pr}_0 \alpha$ is a partition of $\mathbf{Q}$ and for any $(A,S) \in \alpha$, $S \in \mathscr{S}_{|A|}$. Here, pr_i is the ith projection. The set of stabilizer assignments is $C^{\mathbf{Q}}$.

Notation 3.2. Let α be an assignment. We sometimes regard α as a function from $\mathbf{Q}$ to $2^{\mathbf{Q}} \times \mathscr{S}$ such that $\alpha(i) = (A,S)$ where $i \in A$. We define $\alpha_0 = \mathrm{pr}_0 \circ \alpha, \alpha_1 = \mathrm{pr}_1 \circ \alpha$. Hence, $\alpha_0(i) \in 2^{\mathbf{Q}}$ and $\alpha_1(i) \in \mathscr{S}$. We also regard a partition of $\mathbf{Q}$ as a function from $\mathbf{Q}$ to $2^{\mathbf{Q}}$. $\alpha[(A,S)/i]$ is a new assignment $(\alpha \backslash \alpha(i)) \cup \{(A,S)\}$. 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 $(\alpha \backslash \alpha(i)) \cup \{(\alpha_0(i), S)\}$. $\alpha[S/i,j] = (\alpha \backslash (\alpha(i) \cup \alpha(j))) \cup \{(\alpha_0(i) \cup \alpha_1(i), S)\}$.

Definition 3.3. Let ρ be a quantum state and α be an assignment. We write $\alpha \vDash \rho$ 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 $\frac{1}{2}\mathrm{I}$ if $S = \mathrm{I}$ 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 $\frac{1}{4}(\mathrm{I} \otimes \mathrm{I})$ is a separable state, $\{(\{0,1\}, \langle \mathrm{XX}, \mathrm{ZZ} \rangle)\} \vDash \frac{1}{4}(\mathrm{I} \otimes \mathrm{I})$.

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

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

The relation $\leq_c$ makes $C^{\mathbf{Q}}$ a CPO.

Proposition 3.4. *$C^{\mathbf{Q}}$ is a finite lattice and hence a CPO.*

Proof. It is easy to see $\leq_c$ is order. The maximum assignment is $\{(\mathbf{Q},\blacksquare)\}$ and the minimum is $\{(\{i\},\mathrm{I}) \mid i\in\mathbf{Q}\}$. Let α,β be assignments. Take the join π of α_0 and β_0 with respect to $\leq_\pi$. Let $A\in\pi$. Define S as the join of $\bigodot_{j\in A}\alpha_1(j)$ and $\bigodot_{j\in A}\beta_1(j)$. The set of these pairs (A,S) is the join of α and β. The meet of α and β can be constructed similarly. □

We define an abstract semantics $[\![\cdot]\!]_{\mathrm{C}}\colon \mathbf{QIL}\to C^{\mathbf{Q}}\to C^{\mathbf{Q}}$ inductively. For simplicity, we define $U\blacksquare U^\dagger=\blacksquare$ for any unitary U and assume that conditions are exclusive and an upper condition has priority.

$$
\begin{aligned}
[\![\mathtt{skip}]\!]_{\mathrm{C}}(\alpha) &= \alpha\\
[\![C;C']\!]_{\mathrm{C}}(\alpha) &= [\![C']\!]_{\mathrm{C}}([\![C]\!]_{\mathrm{C}}(\alpha))\\
[\![U(i)]\!]_{\mathrm{C}}(\alpha) &= \alpha[U_{(i)}\alpha_1(i)U_{(i)}^\dagger/i]\\
[\![\mathtt{T}(i)]\!]_{\mathrm{C}}(\alpha) &= \begin{cases}\alpha & (\alpha_1(i)\text{ and }\mathrm{Z}_{(i)}\text{ commute})\\ \alpha[\blacksquare/i] & (\text{otherwise})\end{cases}\\
[\![\mathtt{CX}(i,j)]\!]_{\mathrm{C}}(\alpha) &= \begin{cases}\alpha & (\alpha_0(i)=\alpha_0(j)\text{ and }\alpha_1(i)=\blacksquare)\\
\mathit{update}(\{i,j\},\alpha[\mathrm{CX}_{(i,j)}\alpha_1(i)\mathrm{CX}_{(i,j)}^\dagger/i]) & (\alpha_0(i)=\alpha_0(j))\\
\alpha & (\alpha_1(i)=\langle\mathrm{Z}\rangle,\alpha_1(j)=\langle\mathrm{X}\rangle,\\
 & \ \alpha_1(i)=\alpha_1(j)=\mathrm{I})\\
\alpha[\blacksquare/i,j] & (\alpha_1(i)=\blacksquare,\alpha_1(j)=\blacksquare)\\
\alpha[\langle Z\rangle/i] & (\alpha_1(i)=\mathrm{I})\\
\alpha[\langle X\rangle/j] & (\alpha_1(j)=\mathrm{I})\\
\alpha[\mathrm{CX}_{(i,j)}(\alpha_1(i)\otimes\alpha_1(j))\mathrm{CX}_{(i,j)}^\dagger/i,j] & (\text{otherwise})\end{cases}
\end{aligned}
$$

$$
\left[\!\!\left[\begin{array}{l}\mathtt{if}\ \ i\\ \quad\mathtt{then}\ \ C\\ \quad\mathtt{else}\ \ C'\\ \mathtt{fi}\end{array}\right]\!\!\right]_{\mathrm{C}}(\alpha) = [\![C]\!]_{\mathrm{C}}(\mathit{meas}(\alpha))\vee[\![C']\!]_{\mathrm{C}}(\mathit{meas}(\alpha))
$$

$$
[\![\mathtt{while}\ i\ \mathtt{do}\ C\ \mathtt{od}]\!]_{\mathrm{C}}(\alpha) = \bigvee_{n\in\mathbb{N}}\mathit{meas}(([\![C]\!]_{\mathrm{C}}\circ\mathit{meas})^n(\alpha))
$$

where $U\in\{\mathtt{X},\mathtt{Y},\mathtt{Z},\mathtt{H},\mathtt{S}\}$. *update* makes a "pseudo-"assignment to satisfy the condition that each stabilizer contains neither X_i, Y_i, nor Z_i. The first argument are possibly-unentangled qubits.

$$
\begin{aligned}
\mathit{update}(J,\xi) &= \{(A,S)\in\xi\mid A\cap J=\emptyset\}\cup\{\mathit{divide}(J,A,S)\mid J\subset A\}\\
\mathit{divide}(\emptyset,A,S) &= \{(A,S)\}\\
\mathit{divide}(\{i\}\cup J,A,S) &= \begin{cases}\{(\{i\},S')\}\cup\mathit{divide}(J,A\backslash\{i\},S'') & (S=S'\otimes S''\text{ such that }S'\in\mathscr{S}_1\text{ and }S'\text{ has}\\ & \ \text{non-identity entry only in the }i\text{th column})\\ \mathit{divide}(J,A,S) & (\text{otherwise})\end{cases}
\end{aligned}
$$

Note that, in the definition of $[\![\cdot]\!]_{\mathrm{C}}$, the arguments of update satisfy that there exists unique $(A,S)\in\xi$ such that $A\cap J\neq\emptyset$. *meas* means measurement. After measurement, the measured qubit is always separated.

$$
\mathit{meas}(\alpha) = \begin{cases}\alpha[\langle\mathrm{Z}\rangle/i] & (|\alpha_0(i)|=1)\\ \alpha[\{(\{i\},\langle\mathrm{Z}\rangle),(\alpha_0(i)\backslash\{i\},\blacksquare)\}/i] & (\alpha_1(i)=\blacksquare)\\ \mathit{update}(\alpha_0(i),\alpha[\mathit{meas}_{st}(i,\alpha_1(i))/i]) & (\text{otherwise})\end{cases}
$$

where $meas_{st}$ is the measurement process of the ith qubit in stabilizer formalism.

Example 3.5. $[\![\texttt{GHZ}]\!]_{\mathrm{C}}(\alpha) = \{(\{0,1,2\},\langle \mathrm{XXX},\mathrm{ZZI},\mathrm{IZZ}\rangle)\}$, $[\![\texttt{SEP}_0]\!]_{\mathrm{C}}(\alpha) = \{(\{0\},\langle \mathrm{X}\rangle),(\{1\},\langle \mathrm{Z}\rangle),(\{2\},\langle \mathrm{Z}\rangle)\}$, $[\![\texttt{SEP}_1]\!]_{\mathrm{C}}(\alpha) = \{(\{0\},\langle \mathrm{Z}\rangle),(\{1\},\langle \mathrm{Z}\rangle),(\{2\},\langle \mathrm{Z}\rangle)\}$, and $[\![\texttt{NSEP}]\!]_{\mathrm{C}}(\alpha) = \{(\{0\},\langle \mathrm{Z}\rangle),(\{1,2\},\langle \mathrm{XX},\mathrm{ZZ}\rangle)\}$ where $\texttt{meas}(i) \equiv \texttt{if } i \texttt{ then skip else skip fi}$.

Example 3.6. Take a QIL program $\texttt{exm}_0 = \texttt{T(0);if 1 then skip else CX(2,3) fi}$. Let $|B_{00}\rangle$ be a Bell state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ and $\rho_{\texttt{exm}_0} = |B_{00}\rangle\langle B_{00}| \otimes |B_{00}\rangle\langle B_{00}|$. Since $|B_{00}\rangle$ is stabilized by $\langle \mathrm{ZZ},\mathrm{XX}\rangle$ and

$$\left[\begin{array}{cc|cc} \mathrm{Z} & \mathrm{Z} & & \\ \mathrm{X} & \mathrm{X} & & \\ \hline & & \mathrm{X} & \mathrm{X} \\ & & \mathrm{Z} & \mathrm{Z} \end{array}\right] \xrightarrow{\texttt{T(0)}} \left[\begin{array}{cc|cc} \blacksquare & \blacksquare & & \\ \blacksquare & \blacksquare & & \\ \hline & & \mathrm{X} & \mathrm{X} \\ & & \mathrm{Z} & \mathrm{Z} \end{array}\right] \xrightarrow{meas} \left[\begin{array}{c|c|cc} \blacksquare & & & \\ \hline & \mathrm{Z} & & \\ \hline & & \mathrm{X} & \mathrm{X} \\ & & \mathrm{Z} & \mathrm{Z} \end{array}\right] \xrightarrow{\texttt{CX(2,3)}} \left[\begin{array}{c|c|c|c} \blacksquare & & & \\ \hline & \mathrm{Z} & & \\ \hline & & \mathrm{X} & \\ \hline & & & \mathrm{Z} \end{array}\right],$$

$[\![\texttt{exm}_0]\!]_{\mathrm{C}}(\rho_{\texttt{exm}_0})$ is $\{(\{0\},\blacksquare),(\{1\},\mathrm{Z}),(\{2,3\},\blacksquare)\}$. Note $\{(\{2,3\},\langle \mathrm{XX},\mathrm{ZZ}\rangle)\} \vee \{(\{2\},\langle \mathrm{X}\rangle),(\{3\},\langle \mathrm{Z}\rangle)\}$ is $\{(\{2,3\},\blacksquare)\}$.

In the above example, we can see that CX undoes quantum entanglement between the second and third qubits. It enables us to analyse entanglement in a QIL program more deeply than the prior work. Of course, in order to use $[\![\cdot]\!]_{\mathrm{C}}$ for analysis, the abstract semantics should be sound for the concrete semantics. Indeed, $[\![\cdot]\!]_{\mathrm{C}}$ is monotone and sound as the abstract semantics in the paper [12] is.

Proposition 3.7. *For any assignment α,β, and QIL program C, $\alpha \leq_c \beta$ implies $[\![C]\!]_{\mathrm{C}}(\alpha) \leq_c [\![C]\!]_{\mathrm{C}}(\beta)$.*

Proof. By induction on the structure of C. □

Theorem 3.8. *For any state ρ, assignment α, and QIL program C, $\alpha \vDash \rho$ implies $[\![C]\!]_{\mathrm{C}}(\alpha) \vDash [\![C]\!](\rho)$.*

Proof. By induction on the structure of C. For skip, $C;C'$, $U(i)$, and $\texttt{T}(i)$, it is easy. For $\texttt{CX}(i,j)$, there are several cases. But, in any case, it is straightforward that the statement holds by the definition of $\alpha \vDash \rho$ and computation in stabilizer formalism. Note that $\alpha \vee \beta \vDash \rho + \sigma$ whenever $\alpha \vDash \rho$ and $\beta \vDash \sigma$. The statement holds for $\texttt{if } i \texttt{ then } C \texttt{ else } C' \texttt{ fi}$ because of the above fact, $meas(\alpha) \vDash |0\rangle\langle 0|\rho|0\rangle\langle 0|$, and $meas(\alpha) \vDash |1\rangle\langle 1|\rho|1\rangle\langle 1|$. Finally, we show for $\texttt{while } i \texttt{ do } C \texttt{ od}$. Because of $meas(\alpha) \vDash |0\rangle\langle 0|\rho|0\rangle\langle 0|$ and the induction hypothesis, $\bigvee_{n\leq M} meas(([\![C]\!]_{\mathrm{C}} \circ meas)^n(\alpha)) \vDash \sum_{n\leq M} |1\rangle\langle 1|_{(i)} f^n(\rho)|1\rangle\langle 1|_{(i)}$. Since $C^{\mathbf{Q}}$ is finite, $[\![\texttt{while } i \texttt{ do } C \texttt{ od}]\!]_{\mathrm{C}}(\alpha) \vDash \sum_{n\leq M} |1\rangle\langle 1|_{(i)} f^n(\rho)|1\rangle\langle 1|_{(i)}$ for sufficiently large M. Thus, the statement holds. □

4 Abstract domain on extended stabilizers

In the previous section, we use stabilizers and the symbol ■ that represents a non-stabilizer. The symbol ■ contains no information. It just states that the state of the associated qubits is unknown. The abstract semantics $[\![\cdot]\!]_{\mathrm{C}}$ introduces the symbol when it faces the non-Clifford gate T because the post-execution state is a non-stabilizer state. Can not we really extract meaningful information from the post-execution state? Let us take the following QIL program.

$$\texttt{exm}_1 \equiv \texttt{GHZ;T(1);meas(0)}$$

The abstract semantics

$$\begin{bmatrix} \mathrm{X} & \mathrm{X} & \mathrm{X} \\ \mathrm{Z} & \mathrm{Z} & \mathrm{I} \\ \mathrm{Z} & \mathrm{I} & \mathrm{Z} \end{bmatrix} \xrightarrow{\texttt{T(1)}} \begin{bmatrix} \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare \end{bmatrix} \xrightarrow{\texttt{meas(0)}} \left[\begin{array}{c|cc} \mathrm{Z} & & \\ \hline & \blacksquare & \blacksquare \\ & \blacksquare & \blacksquare \end{array}\right]$$

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 ■ when T appears, and to memorise the applied gates. Recall that $USU^\dagger$ "stabilizes" UV_S if S is the stabilizer of V_S.

$$\begin{bmatrix} \mathrm{X} & \mathrm{X} & \mathrm{X} \\ \mathrm{Z} & \mathrm{Z} & \mathrm{I} \\ \mathrm{Z} & \mathrm{I} & \mathrm{Z} \end{bmatrix} \xrightarrow{\mathtt{T(1)}} \begin{bmatrix} \mathrm{X} & \mathrm{TXT}^\dagger & \mathrm{X} \\ \mathrm{Z} & \mathrm{Z} & \mathrm{I} \\ \mathrm{Z} & \mathrm{I} & \mathrm{Z} \end{bmatrix} \xrightarrow{\mathtt{meas(0)}} \left[\begin{array}{c|c|c} \mathrm{Z} & & \\ \hline & \mathrm{Z} & \\ \hline & & \mathrm{Z} \end{array}\right]$$

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

Definition 4.1. Let k be a natural number and A be a $k \times k$ matrix whose entries are in $\{\mathrm{I}, \mathrm{X}, \mathrm{Y}, \mathrm{Z}, \heartsuit\}$. We now identify two matrices A and B if A can be converted into B via permutation of rows. We name a row containing the symbol ♡ and a row containing no ♡ by a ♡-row and an L-row respectively. We always require any L-rows commute. Moreover, we require that for any ♡-row R_i and row M_j, by substituting I, X, Y, or Z for each ♡ in R_i and M_j, the result rows can commute. For example, the matrix consisting of two rows ♡X and IZ is excluded, but the matrix consisting of ♡X and XZ is right because substitution of Z for ♡ makes these rows commute. We further identify two matrices A and B if for any L-row L_i of A, there exists L-rows $L_{j_0}, \ldots, L_{j_{m_i}}$ of B such that $L_i = L_{j_0} \cdots L_{j_{m_i}}$ and for any ♡-row R_i of A, there exists rows $R_{j_0}, \ldots, R_{j_{m_i}}, L_{j_0}, \ldots, L_{j_{l_i}}$ of B such that $R_i = R_{j_0} \cdots R_{j_{m_i}} L_{j_0} \cdots L_{j_{l_i}}$, and vice versa. Here, ♡ behaves as an absorbing element. Finally, we excludes some matrices. Let $k \geq 2$. If a matrix has a row $\mathrm{II}\cdots\mathrm{I}$, $\mathrm{X}_{(j)}$, $\mathrm{Y}_{(j)}$, $\mathrm{Z}_{(j)}$, or $\heartsuit_{(j)}$, then it is excluded. If a matrix has a column that contains only I and one of $\{\mathrm{X}, \mathrm{Y}, \mathrm{Z}\}$, then the matrix is also eliminated. Finally, if a matrix has a column such that exact one entry is ♡ and the others are I, it is excluded. We name the set of those matrices $\mathscr{E}_k$. Then, ■ is added into all $\mathscr{E}_k$. $\mathscr{E}$ is the union of these $\mathscr{E}_k$s.

Example 4.2.

$$\begin{bmatrix} \mathrm{I} \end{bmatrix}, \begin{bmatrix} \heartsuit \end{bmatrix}, \begin{bmatrix} \mathrm{X} & \heartsuit \\ \heartsuit & \mathrm{X} \end{bmatrix}, \begin{bmatrix} \heartsuit & \mathrm{X} & \mathrm{Y} \\ \mathrm{Z} & \heartsuit & \heartsuit \\ \mathrm{X} & \mathrm{Y} & \mathrm{Z} \end{bmatrix} \in \mathscr{E}, \quad \begin{bmatrix} \heartsuit & \heartsuit & \mathrm{Y} \\ \mathrm{I} & \mathrm{X} & \mathrm{I} \\ \mathrm{Z} & \mathrm{I} & \mathrm{X} \end{bmatrix}, \begin{bmatrix} \mathrm{X} & \mathrm{Z} \\ \mathrm{X} & \heartsuit \end{bmatrix}, \begin{bmatrix} \heartsuit & \mathrm{Y} \\ \mathrm{I} & \mathrm{X} \end{bmatrix} \notin \mathscr{E}$$

The third matrix is an abstraction of matrices such as

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

Recall that $\mathscr{S}$ has the order $\leq_s$. Regardless of the addition of ♡, the same definition seems to give an order of $\mathscr{E}$: $E \in \mathscr{E}$ is lower than or equal to $E' \in \mathscr{E}$ if $E = \mathrm{I}$, $E' = $ ■, or $E = E'$. However, it does not answer our purpose. Recall the join operator corresponds with the summation of density matrices. For example, $\begin{bmatrix} \mathrm{X} & \heartsuit \\ \heartsuit & \mathrm{X} \end{bmatrix}$ may represent $\begin{bmatrix} \mathrm{X} & \mathrm{I} \\ \mathrm{I} & \mathrm{X} \end{bmatrix}$ or $\begin{bmatrix} \mathrm{X} & \mathrm{Z} \\ \mathrm{Z} & \mathrm{X} \end{bmatrix}$. But, the summation of stabilized states by them does not always have the form of $\begin{bmatrix} \mathrm{X} & \heartsuit \\ \heartsuit & \mathrm{X} \end{bmatrix}$. The example shows the join of $\begin{bmatrix} \mathrm{X} & \heartsuit \\ \heartsuit & \mathrm{X} \end{bmatrix}$ and $\begin{bmatrix} \mathrm{X} & \heartsuit \\ \heartsuit & \mathrm{X} \end{bmatrix}$ should not be itself, so the "order" is not reflexive.

In order to obtain a join operator, we remove rows that contain ♡. We give up keeping information about unitary matrices when we take a join.

Definition 4.3. Take $A \in \mathscr{E}_k$. Remove all ♡-rows. If all rows are ♡-rows, we obtain ■. We call this procedure *normalisation* and these matrices *normal forms*. The set of normal forms is $\mathscr{F}_k$ and the union of them is $\mathscr{F}$. We redefine $\mathscr{E}_k$ and $\mathscr{E}$ as $\mathscr{E}_k \cup \mathscr{F}_k$ and $\mathscr{E} \cup \mathscr{F}$ respectively.

Notation 4.4. For each $E \in \mathscr{E}$, E_{nl} is the normal form of E.

Example 4.5.

$$\left[\ \mathrm{I}\ \right], \left[\ \blacksquare\ \right], \left[\ \mathrm{X} \quad \mathrm{Y} \quad \mathrm{Z}\ \right] \in \mathscr{F}$$

$\mathscr{F}$ has an order $\leq_f$: $F \leq_f F'$ if $F = \mathrm{I}$, $F' = \blacksquare$, or $F = F'$. Obviously, $\mathscr{F}$ 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 $\mathscr{E}$ via the subset $\mathscr{F}$.

Now, we define our second abstract domain $E^{\mathbf{Q}}$.

Definition 4.6. We call $\gamma \subset 2^{\mathbf{Q}} \times \mathscr{E}$ an *extended (stabilizer) assignment* if $\mathrm{pr}_0\gamma$ is a partition of $\mathbf{Q}$ and for any $(A,E) \in \gamma$, $E \in \mathscr{E}_{|A|}$. The set of extended assignments is $E^{\mathbf{Q}}$. For each extended assignment γ, an extend assignment $\{(A, E_{nl}) \mid (A,E) \in \gamma\}$ is the *normal form* of γ. $F^{\mathbf{Q}}$ 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 ρ be a quantum state and γ be an extended assignment. We write $\gamma \vDash \rho$ if ρ is $\mathrm{pr}_0\gamma$-separable and for any L-row L_i of any $E \in \gamma$, $P^+_{L_i} \rho P^-_{L_i} = 0$. Recall $P^{\pm}_{L_i} = \frac{1}{2}(\mathrm{I}^{\otimes n} \pm L_i)$.

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

$$[\![\mathtt{skip}]\!]_{\mathrm{E}}(\gamma) = \gamma$$

$$[\![C;C']\!]_{\mathrm{E}}(\gamma) = [\![C']\!]_{\mathrm{E}}([\![C]\!]_{\mathrm{E}}(\gamma))$$

$$[\![U(i)]\!]_{\mathrm{E}}(\gamma) = \gamma[U_{(i)}\gamma_1(i)U^\dagger_{(i)}/i]$$

$$[\![\mathtt{T}(i)]\!]_{\mathrm{E}}(\gamma) = \begin{cases} \gamma & (\gamma_1(i) \text{ and } \mathrm{Z}_{(i)} \text{ commute}) \\ \gamma[add_\heartsuit(i, \gamma_1(i))/i] & (\text{otherwise}) \end{cases}$$

$$[\![\mathtt{CX}(i,j)]\!]_{\mathrm{E}}(\gamma) = \begin{cases} \gamma & (\gamma_0(i) = \gamma_0(j) \text{ and } \gamma_1(i) = \blacksquare) \\ update_E(\{i,j\}, \gamma[\mathrm{CX}_{(i,j)}\gamma_1(i)\mathrm{CX}^\dagger_{(i,j)}/i]) & (\gamma_0(i) = \gamma_0(j)) \\ \gamma & (\gamma_1(i) = \langle \mathrm{Z} \rangle, \gamma_1(j) = \langle \mathrm{X} \rangle, \\ & \ \gamma_1(i) = \gamma_1(j) = \mathrm{I}) \\ \gamma[\blacksquare/i,j] & (\gamma_1(i) = \blacksquare, \gamma_1(j) = \blacksquare) \\ \gamma[\langle Z \rangle/i] & (\gamma_1(i) = \mathrm{I}) \\ \gamma[\langle X \rangle/j] & (\gamma_1(j) = \mathrm{I}) \\ \gamma[\mathrm{CX}_{(i,j)}(\gamma_1(i) \otimes \gamma_1(j))\mathrm{CX}^\dagger_{(i,j)}/i,j] & (\text{otherwise}) \end{cases}$$

$$\left[\!\!\left[\begin{array}{l} \texttt{if}\quad i \\ \qquad \texttt{then}\quad C \\ \qquad \texttt{else}\quad C' \\ \texttt{fi} \end{array}\right]\!\!\right]_{\mathrm{E}}(\gamma) = [\![C]\!]_{\mathrm{E}}(meas_E(\gamma)) \uplus [\![C']\!]_{\mathrm{E}}(meas_E(\gamma))$$

$$[\![\texttt{while}\ i\ \texttt{do}\ C\ \texttt{od}]\!]_{\mathrm{E}}(\gamma) = \biguplus_{n\in\mathbb{N}} meas_E(([\![C]\!]_{\mathrm{E}} \circ meas_E)^n(\gamma))$$

where $U \in \{\mathtt{X}, \mathtt{Y}, \mathtt{Z}, \mathtt{H}, \mathtt{S}\}$.

The result $update_E(J', \gamma)$ is computed as follows. Take $(J, E) \in \gamma$ such that $J' \subset J$. (1) If E does not contain $\heartsuit$, then $update(J', \gamma)$ is the result. (2) If E contains $\heartsuit$ but belongs to $\mathscr{E}$, then γ is the result. Note that in this case, the definition of equality in $\mathscr{E}$ ensures that there is not a row such as $\mathrm{X}_{(i)}$. (3) If not, take all $j_0, \ldots, j_{k-1} \subset J'$ such that each j_l column is composed of I and one of X, Y, Z. It means that the j_l qubit is separated. Let $\diamondsuit_{j_l}$ is X, Y, or Z that the j_lth column contains. Define $J'' = J \backslash \{j_0, \ldots, j_{k-1}\}$. Then, the result is $\gamma[\{(J'', \blacksquare)\} \cup \{(\{j_l\}, \diamondsuit_{j_l}) \mid l = 0, \ldots, k-1\}/J]$.

The result of $meas_E(\gamma)$ varies with γ. (1) If $|\gamma_0(i)| = 1$, then $\gamma[\langle \mathrm{Z} \rangle / i]$ is the result. (2) If $\gamma_1(i)$ is a square matrix and does not contain $\heartsuit$ but is not $\blacksquare$, then $update(\gamma_0(i), \gamma[meas_{st}(i, \gamma_1(i))/i])$, which is the same as $meas$. (3) If exactly one row of $\gamma_1(i)$ has X or Y in the ith column, $meas_E(\gamma)$ is computed as follows. First, the row and the ith column are removed from $\gamma_1(i)$. Let us call the matrix E'. Then $update_E(\gamma_0(i) \backslash \{i\}, \gamma[\{(\{i\}, \langle \mathrm{Z} \rangle), (\gamma_0(i) \backslash \{i\}, E')\}/i])$ is the result. (4) Otherwise, we cannot obtain information about the post-measurement state. The result is $\gamma[\{(\{i\}, \langle \mathrm{Z} \rangle), (\gamma_0(i) \backslash \{i\}, \blacksquare)\}/i]$.

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

Example 4.9. Now, we compute $[\![\mathtt{exm_1}]\!]_{\mathrm{E}}(\gamma)$.

$$\begin{bmatrix} \mathrm{X} & \mathrm{X} & \mathrm{Z} \\ \mathrm{Z} & \mathrm{Z} & \mathrm{I} \\ \mathrm{Z} & \mathrm{I} & \mathrm{Z} \end{bmatrix} \xrightarrow{\mathtt{T(1)}} \begin{bmatrix} \mathrm{X} & \heartsuit & \mathrm{X} \\ \mathrm{Z} & \mathrm{Z} & \mathrm{I} \\ \mathrm{Z} & \mathrm{I} & \mathrm{Z} \end{bmatrix} \xrightarrow{\mathtt{meas(0)}} \left[\begin{array}{c|c|c} \mathrm{Z} & & \\ \hline & \mathrm{Z} & \\ \hline & & \mathrm{Z} \end{array}\right]$$

Thus, we conclude that all qubits are separated.

Finally, we show $[\![\cdot]\!]_{\mathrm{E}}$ is sound.

Theorem 4.10. *For any state ρ, extended assignment γ, and program C, $\gamma \vDash \rho$ implies $[\![C]\!]_{\mathrm{E}}(\gamma) \vDash [\![C]\!](\rho)$.*

Proof. By induction on the structure of C. For `skip`, $C; C'$, $U(i)$, and $\mathtt{T}(i)$, it is easy. For $\mathtt{CX}(i, j)$, since the number of $\heartsuit$-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 $\mathtt{if}\ i\ \mathtt{then}\ C\ \mathtt{else}\ C'\ \mathtt{fi}$, we have to check $meas_E$. However, since it also just decrease the number of $\heartsuit$-rows, $meas_E(\gamma) \vDash |0\rangle\langle 0|_{(i)} \rho |0\rangle\langle 0|_{(i)}$. Hence, the statement holds for $\mathtt{if}\ i\ \mathtt{then}\ C\ \mathtt{else}\ C'\ \mathtt{fi}$. Finally, we show for $\mathtt{while}\ i\ \mathtt{do}\ C\ \mathtt{od}$. Since $C^{\mathbf{Q}}$ is finite, $[\![\mathtt{while}\ i\ \mathtt{do}\ C\ \mathtt{od}]\!]_{\mathrm{E}}(\gamma) \vDash \sum_{n \le M} |1\rangle\langle 1|_{(i)} f^n(\rho) |1\rangle\langle 1|_{(i)}$ 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 $C^{\mathbf{Q}}$ 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 $\mathtt{SEP}_0$ or $\mathtt{SEP}_1$. Moreover, we defined an abstract domain $E^{\mathbf{Q}}$, as $C^{\mathbf{Q}}$ 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, the stabilizer formalism was used to verify quantum programs and analyse entanglement of those programs [5, 6]. However, in the study, 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 $C^{\mathbf{Q}}$ to $E^{\mathbf{Q}}$, further refinement is still needed such as finding a better approximate join operator in $E^{\mathbf{Q}}$.

Acknowledgements

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

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$_5$1.

[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.

[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$_1$8.

[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$_2$5.

[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.